A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries

Bo Li[a,*], Yijuan Lu[a], Chunyuan Li[b], Afzal Godil[b], Tobias Schreck[c], Masaki Aono[d], Martin Burtscher[a], Qiang Chen[e], Nihad Karim Chowdhury[d], Bin Fang[e], Hongbo Fu[f], Takahiko Furuya[g], Haisheng Li[h], Jianzhuang Liu[i], Henry Johan[j], Ryuichi Kosaka[d], Hitoshi Koyanagi[d], Ryutarou Ohbuchi[g], Atsushi Tatsuma[d], Yajuan Wan[h], Chaoli Zhang[h], Changqing Zou[k]

[a] *Department of Computer Science, Texas State University, San Marcos, USA*
[b] *Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, USA*
[c] *Computer and Information Science, University of Konstanz, Konstanz, Germany*
[d] *Department of Computer Science and Engineering, Toyohashi University of Technology, Japan*
[e] *College of Computer Science, Chongqing University, China*
[f] *School of Creative Media, City University of Hong Kong, Hong Kong, China*
[g] *Department of Computer Science and Engineering, University of Yamanashi, Yamanashi, Japan*
[h] *School of Computer and Information Engineering, Beijing Technology and Business University, Beijing, China*
[i] *Media Laboratory, Huawei Technologies Co. Ltd., Shenzhen, China*
[j] *Visual Computing, Fraunhofer IDM@NTU, Singapore*
[k] *Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China*

Abstract

Large-scale 3D shape retrieval has become an important research direction in content-based 3D shape retrieval. To promote this research area, two Shape Retrieval Contest (SHREC) tracks on large scale comprehensive and sketch-based 3D model retrieval have been organized by us in 2014. Both tracks were based on a unified large-scale benchmark that supports multimodal queries (3D models and sketches). This benchmark contains 13,680 sketches and 8,987 3D models, divided into 171 distinct classes. It was compiled to be a superset of existing benchmarks and presents a new challenge to retrieval methods as it comprises generic models as well as domain-specific model types. Twelve and six distinct 3D shape retrieval methods have competed with each other in these two contests, respectively. To measure and compare the performance of the participating and other promising Query-by-Model or Query-by-Sketch 3D shape retrieval methods and to solicit state-of-the-art approaches, we perform a more comprehensive comparison of twenty-six (eighteen originally participating algorithms and eight additional state-of-the-art or new) retrieval methods by evaluating them on the common benchmark. The benchmark, results, and evaluation tools are publicly available at our websites [1, 2].

Keywords:
3D shape retrieval, Large-scale benchmark, Multimodal queries, Unified, Performance evaluation, Query-by-Model, Query-by-Sketch, SHREC

1. Introduction

With the increasing number of 3D models created every day and stored in databases, the development of effective and scalable 3D search algorithms has become an important research area. Generally speaking, their objective is to retrieve 3D models similar to a 2D/3D sketch/image or a complete 3D model query from a large collection of 3D shapes. In this paper, we present a new large-scale benchmark that includes a large number of diverse types of sketches and models. Owing to the integration of the most important existing benchmarks to date, the newly created benchmark is the most extensive to date in terms of the number of semantic query categories covered as well as the variations of model types. In particular, it combines generic and domain-dependent model types and therefore rates the retrieval performance with respect to cross-domain retrieval tasks. The benchmark supports both sketch and 3D model queries, thus providing a unified platform to test diverse 3D model retrieval algorithms belonging to either Query-by-Model or Query-by-Sketch 3D retrieval techniques.

Query-by-Model 3D retrieval is one of the most commonly seen and most widely studied 3D model retrieval techniques. Many dedicated algorithms and several benchmarks have been developed for this type of 3D retrieval. However, it requires users to provide a 3D model as a query.

Query-by-Sketch (sketch-based) 3D retrieval is to retrieve a list of 3D models that closely match a provided input sketch. Compared to Query-by-Model, it is more intuitive and easier

*Corresponding author at: 601 University Drive, Department of Computer Science, Texas State University, San Marcos, Texas 78666; E-mail: b_l58@txstate.edu, li.bo.ntu0@gmail.com; Tel: +001 512 245 6580; Fax: +001 512 245 8750.

to use because users do not need to provide 3D models. However, it is also more challenging because of the semantic and representational gap between the 2D query sketches and the 3D models, and because user sketches may vary widely in sketching style and level of detail, as well. It has many applications, including sketch-based modeling and recognition, and sketch-based 3D animation [3].

Two previous Shape Retrieval Contest (SHREC) tracks, SHREC'12 [4] and SHREC'13 [5], have been successfully organized on the topic of sketch-based 3D model retrieval. They invigorated this research area by providing a small-scale and a large-scale sketch-based retrieval benchmark, respectively, and attracted state-of-the-art algorithms to compete with each other. Yet, even the large-scale SHREC'13 Sketch Track Benchmark (**SHREC13STB**) [5] based on Eitz et al. [6] and the Princeton Shape Benchmark (PSB) [7] contains only 90 classes of 7,200 sketches and 1,258 models. Compared with the complete dataset of 250 user sketch classes compiled by Eitz et al. [6], there is still substantial room to make the benchmark more comprehensive in terms of completeness of object classes existing in the real world. Thus, we felt it is necessary to build an even larger sketch-based 3D retrieval benchmark with more sketches and more models to help better evaluate the scalability of existing and newly developed sketch-based 3D model retrieval algorithms. Considering this, we created a new large-scale benchmark (**LSB**) comprising 13,680 sketches and 8,987 available 3D models from 171 classes that can be and also have been used to evaluate both Query-by-Sketch and Query-by-Model 3D retrieval algorithms. Figure 1 shows several example sketches and their relevant 3D models.

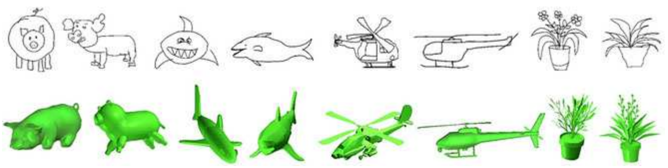


Figure 1: Example 2D sketches and their relevant 3D models in the large scale benchmark (**LSB**).

Based on this new benchmark, we organized a SHREC 2014 track [8] on large scale sketch-based 3D model retrieval to further foster this challenging research area by soliciting retrieval results from current state-of-the-art retrieval methods for comparison, especially in terms of scalability to a large-scale scenario. Moreover, by utilizing only the 3D target dataset of the benchmark, we organized another SHREC'14 track [9] on the topic of large scale comprehensive 3D shape retrieval to perform a comparison, especially for practical retrieval performance, of top 3D model retrieval methods. Thus, the two contest tracks have demonstrated the unification and large-scale properties of our benchmark in evaluating both Query-by-Model and Query-by-Sketch 3D retrieval techniques.

In the rest of the paper, we first review the related work (w.r.t. techniques and benchmarks) in Section 2. In Section 3, we introduce the motivation, building process, contents, and evaluation metrics (containing both general and weighted variations) of the benchmark. Section 4 gives a brief introduction of the contributors of the paper. A short and concise description of each contributed method is presented in Section 5. Section 6 describes the evaluation results of the 22 Query-by-Model and 6 Query-by-Sketch 3D retrieval algorithms on the unified benchmark. Section 7 concludes the paper and lists several future research directions.

2. Related work

In this section, we mainly concentrate on related work published within the last three years. The latest review of sketch-based 3D model retrieval techniques and benchmarks is presented in [10]. Thus, we will primarily review the recent progress in the Query-by-Model techniques, especially in generic, non-rigid, and semantics-based 3D model retrieval. For partial 3D retrieval techniques, please refer to [11] and [12] for the latest reviews.

2.1. Generic 3D model retrieval techniques

Three important surveys have been written by Iyer et al. [13], Bustos et al. [14], and Tangelder et al. [15], who reviewed typical generic 3D model retrieval techniques before 2008. Based on the types of features employed, existing generic 3D model retrieval techniques can be classified into four categories: geometry-based, graph-based, view-based, and hybrid techniques.

2.1.1. Geometry-based techniques

Geometry-based techniques characterize the geometric information of a 3D model based on the distribution of geometric elements. Research on the feature extraction of generic 3D models is usually designed with the following two goals: (1) strong discriminative ability w.r.t various 3D models; and (2) adequate generality w.r.t the robustness to different geometric representations, including surfaces (i.e., meshes and parametric/subdivison/implicit surfaces), solids (i.e., volume data), and raw data (i.e., point clouds, range images, or polygon soups). These 3D features can be either global, such as Shape Distribution [16] and Shape Histogram [17]; or local, such as the 3D shape context [18, 19, 20], Extended Gaussian Images (EGI) [21], conformal factor [22], spherical harmonics [23], and Poisson histogram descriptor [24].

Recently, Sipiran et al. [25] enhanced the traditional Bag-of-Feature framework for generic shapes with their data-aware partition approach. Zou et al. [26] proposed a combined shape distribution descriptor based on principal plane analysis and group integration.

Two of the methods evaluated in this paper belong to this category: Zhang's Modified Shape Distribution (MSD) and Shell-Distance-Sum (SDS) (Section 5.1.6).

2.1.2. Graph-based techniques

Graph-based methods perform matching among models by using their skeletal or topological graph structures. Skeleton graph-based approaches abstract a 3D model as a low-dimensional graph, which visually preserves the global shape configuration and whose nodes and edges correspond to the geometric attributes of the shape components. A typical example is proposed in [27]. Recently, a geodesic skeleton path-based approach has been proposed in [28], where the geometry of a 3D mesh is coded as a sequence of radii of the maximal balls at the skeleton points.

Topology-based methods compare 3D models based on the difference in their global topological structures. Among the various topology representations, Reeb graphs, which are rooted in the Morse theory, are considered one of the most popular. One typical example based on Reeb graph is presented in [29]. Recently, Barra et al. [30] compared 3D models based on the kernel functions defined on extended Reeb graphs. Another direction relies on the theory of Topological Persistence. It was first formalized by Edelsbrunner et al. [31] as the concept of persistence diagram or barcode and builds on previous related work on size functions [32]. The method provides a principled way to qualitatively visualize and measure the topological structures via the feature functions defined on the shape surface. Topological Persistence recently became of interest for shape retrieval tasks [33, 34] partially due to the popularity of topological data analysis [35].

2.1.3. View-based techniques

View-based techniques use a set of rendered views to represent a 3D model. The visual similarity between the views of two models is regarded as the model difference. A special survey has been published in [36]. Efforts along this line are mostly devoted to two stages: descriptive feature extraction from certain view images and appropriate comparison between sets of visual features. For the former, typical approaches include Light Field descriptors [37], the Multi-view Depth Line Approach (MDLA) [38], salient local visual features [39], Compact Multi-View Descriptor (CMVD) [40], and View Context shape descriptor [41]. For the latter, basic work includes the Bag-of-Features based approach [42] and its variants such as Bag-of-Region-Words [43] as well as more accurate 3D model alignment-based methods [44].

Recently, Ding et al. [45] defined a view-based shape descriptor named Sphere Image that integrates the spatial information of a collection of viewpoints and their corresponding view features that are matched based on a probabilistic graphical model. Similar to the Sphere Image, Bonaventura et al. [46] proposed a 3D shape descriptor of the Information Sphere and utilized mutual information-based measures for the matching, whereas Liang et al. [47] designed a feature named Spherical-SIFT to represent the salient local features on spherical images. As for applications, Sfikas et al. [48] retrieved complete 3D pottery models based on the panoramic feature views of a partial range image query. These view-based methods have a unique advantage for generic 3D model retrieval tasks in that they focus on the visual features of view images and thus can work on arbitrarily structured 3D models.

The following evaluated methods in this paper belong to this category: Aono's KAZE local feature [49] with the VLAD encoding scheme [50] (KVLAD) (Section 5.1.1), Furuya's Bag-of-Features of Dense SIFT (BF-DSIFT), per-View Matching of One SIFT (VM-1SIFT), Manifold Ranking of BF-DSIFT (MR-BF-DSIFT), Manifold Ranking of D1SIFT (MR-D1SIFT) and Manifold Ranking of 1SIFT (MR-VM-1SIFT) (Section 5.1.3); Tatsuma's Depth Buffered Super-Vector Coding (DBSVC) and Locally Constrained Diffusion Ranking of DBSVC (LCDR-DBSVC) (Section 5.1.5).

2.1.4. Hybrid techniques

Hybrid approaches explicitly employ at least two of the above features to characterize a 3D model. Many hybrid shape descriptors have been proposed in the literature. We list a few recent works, such as DESIRE [51], and DSH [52], which combines **D**epth buffer-based 2D features and **S**pherical **H**armonics-based 3D features. PANORAMA [53] represents a 3D model based on a set of panoramic views and achieves state-of-the-art performance on several generic 3D model databases.

Recently, a hybrid descriptor named ZFDR comprising both geometric and view information has been proposed in [54]. Li et al. [55] combined the topological feature multiresolutional Reeb graph (MRG) based features and modified BOF-based view features. Liu et al. [56] adopted several representative geometric features such as shape diameter function, average geodesic distance, and heat kernel signature, to characterize low-level semantic patches. Tabia et al. [57] proposed to first sample a set of points on the surface of a 3D model, then use the covariance matrices of multiple local features as shape descriptors for 3D face matching, and further apply an extended Bag-of-Words framework on the covariance matrix-based local shape descriptors for 3D model retrieval. Hybrid descriptors are interesting because the integration of different features may better accommodate a diversity of 3D shapes.

Among the evaluated methods, Aono's Center-Symmetric Local Binary Pattern (CSLBP), and Hybrid shape descriptor comprising several features including Surface-Roughness and DEpth-buffer (HSR-DE) (Section 5.1.1), Chen's hybrid shape descriptor DBNAA_DERE, which combines Shape Distribution (D2) [58], Bounding Box, Normal Angle Area, DEpth buffer, and Ray Extend based features [59] (Section 5.1.2), Li's ZFDR hybrid shape descriptor, which integrates Zernike moments, Fourier descriptors, Depth information [59], and Ray-based features [59] (Section 5.1.4), Zhang's Multi-Feature Fusion Based on Entropy Weights (MFF-EW) (Section 5.1.6) and Papadakis' PANORAMA, which stands for PANoramic Object Representation for Accurate Model Attributing [53], fall into this group.

2.2. Non-rigid 3D model retrieval techniques

Unlike generic 3D model retrieval for rigid models, non-rigid 3D model retrieval techniques are dedicated to retrieving the specific and ubiquitous non-rigid 3D models with diverse poses or articulations. Due to the non-rigid properties of the models, it is more challenging to perform the retrieval. For a review of non-rigid 3D retrieval techniques based on geodesic distance and spectrum analysis approaches, as well as different canonical form transforms for non-rigid models based on multidimensional scaling, please refer to [12]. Another recent survey of non-rigid shape retrieval is presented in [60], where a performance comparison of several descriptors derived from spectral geometry is given.

Stability and repeatability are two important properties for local descriptors and interest point detectors, and, hence, are important building blocks for non-rigid shape retrieval methods. Stability and repeatability properties have been studied for a number of object transformations, including non-rigid transformations [61].

Recently, significant efforts have been invested in exploring the invariance properties of shapes to non-rigid deformations. In particular, the emerging field of spectral geometry provides an elegant framework for the geometric analysis of non-rigid shapes, which relies on the Eigensystem (eigenvalues and/or eigenfunctions) of the Laplace-Beltrami operator [62, 63]. Prominent work in this direction includes Shape DNA [64], heat kernel signature (HKS) [65, 66], and wave kernel signature (WKS) [67]. From the perspective of spectral graph wavelets, a general form of spectral descriptors was presented in [68], which includes HKS and WKS as special cases. A classic work in shape retrieval applications is the Shape Google algorithm [69], which aggregates spectral descriptors based on the Bag-of-Features framework. Later, as the spatial partition version, an intrinsic spatial pyramid matching algorithm was developed in [70]. Despite the elegance and popularity of these spectral methods, they require the input 3D models to have a manifold data structure, which is unrealistic for most models collected from the web. Therefore, extra preprocessing is generally needed to remesh the surfaces before feeding them into the framework.

2.3. Semantics-based 3D model retrieval techniques

Semantics-based 3D model retrieval techniques incorporate high-level semantic information of the query and/or 3D models into the retrieval process to bridge the semantic gap existing in traditional content-based 3D model retrieval techniques. A survey of three typical semantics processing techniques (relevance feedback, machine learning, and ontology) is presented in [71]. Typical semantics-based 3D retrieval approaches include relevance feedback [72], semantic labeling [73], neural networks [74], supervised [75, 76, 77, 78] or semi-supervised [79, 80, 81] learning, boosting [82], prototypes [83], autotagging [84], spectral clustering [85], manifold ranking [86], semantic tree [87], feature dimension reduction [88], semantic subspaces [89], class distances [54], semantics annotation of 3D models [90], semantic correspondences [91], and sparse structure regularized ranking [92].

Recently, the attribute-based semantic approach has become popular and has demonstrated promising performance, such as multiple shape indexes (attributes) [93] and attribute-augmented semantic hierarchy [94]. Gong et al. [95] proposed to use attribute signature (AS) and reference set signature (RSS) to perform semantic 3D model retrieval. They selected 11 attributes including symmetry, flexibility, rectilinearity, circularity, dominant-plane, long, thin, swim, fly, stand with leg(s), and natural. They found that their high-level semantic approaches (AS and RSS) can complement low-level features, and they non-trivially improve the retrieval performance when used in combination. They also mentioned that one advantage of their semantic features is the compactness (making them efficient for large-scale retrieval scenarios).

The following evaluated algorithms belong to this type: Aono's machine learning-based method CSLBP* (Section 5.1.1); the manifold ranking-based approaches, including Furuya's MR-D1SIFT and MR-VM-1SIFT (Section 5.1.3) and Tatsuma's LCDR-DBSVC (Section 5.1.5) Query-by-Model algorithms; and Furuya's CDMR (Section 5.2.1) and Tatsuma's SCMR-OPHOG (Section 5.2.3) Query-by-Sketch algorithms.

2.4. 3D model retrieval benchmarks

A recent overview of existing sketch-based 3D model retrieval benchmarks is available in [10]. Hence, we mainly concentrate on the review of currently available generic or specialized 3D model retrieval benchmarks for the Query-by-Model retrieval.

2.4.1. Generic 3D model retrieval benchmarks

To evaluate the performance of a generic 3D model retrieval algorithm, researchers have built generic 3D model retrieval benchmarks including: the Princeton Shape Benchmark (**PSB**) [7], the SHREC'12 Generic Track Benchmark (**SHREC12GTB**) [96], the Toyohashi Shape Benchmark (**TSB**) [97], and the Konstanz 3D Model Benchmark (**CCCC**) [59].

2.4.2. Specialized 3D model retrieval benchmarks

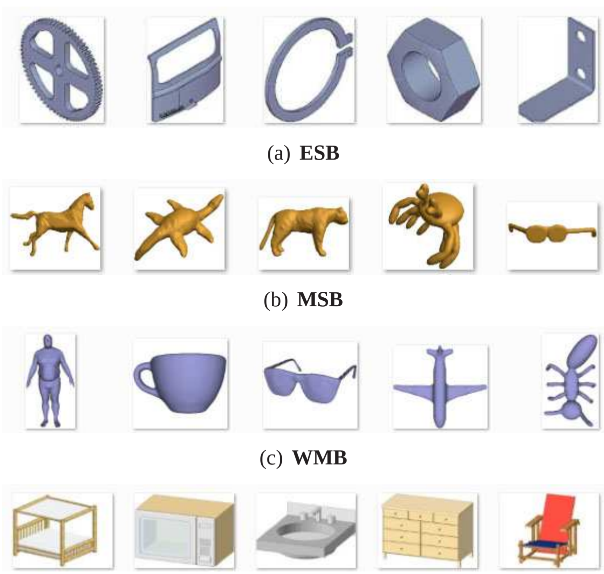
Specialized 3D model retrieval benchmarks are dedicated to testing the performance of a 3D model retrieval algorithm on a particular type of 3D models, such as non-rigid, watertight, or professional. For example, the following specialized 3D benchmarks exist: the Watertight Model Benchmark (**WMB**) [98], the McGill 3D Shape Benchmark (**MSB**) [99], Bonn's Architecture Benchmark (**BAB**) [100], and the Engineering Shape Benchmark (**ESB**) [101].

Table 1 lists the basic classification information of the above eight benchmarks whereas Fig. 2 shows some example models for the four specialized benchmarks. We selected these eight benchmarks to create the 3D target dataset of our benchmark.

Aside from the above mentioned benchmarks, there are several other benchmarks or 3D model resources that may have overlap with the eight benchmarks we selected. They include: (1) generic 3D model datasets like the National Taiwan University 3D model database (**NTU**) [37], the **NIST** dataset [102], the AIM@SHAPE Shape Repository [103], and the SHREC contests datasets (generic retrieval tracks, 2006~2014) [104]; (2) specialized 3D model retrieval benchmarks like the **TOSCA** [105] and SHREC contests datasets (non-rigid, watertight, textured 3D, CAD, protein, face, human, range scan or parts-based partial retrieval tracks, 2006~2014) [104].

Table 1: Classification information of the eight generic or specialized 3D model retrieval benchmarks.

Benchmarks	Types	Number of models	Number of classes	Average number of models per class
PSB	Generic	907 (train) 907 (test)	90 (train) 92 (test)	10 (train) 10 (test)
SHREC12GTB	Generic	1,200	60	20
TSB	Generic	10,000	352	28
CCCC	Generic	473	55	9
WMB	Watertight (articulated)	400	20	20
MSB	Articulated	457	19	24
BAB	Architecture	2,257	183 (function-based) 180 (form-based)	12 (function-based) 13 (form-based)
ESB	CAD	867	45	19



(a) **ESB**

(b) **MSB**

(c) **WMB**

(d) **BAB**

Figure 2: Example 3D models in **ESB**, **MSB**, **WMB** and **BAB** datasets.

3. Benchmark

3.1. Motivation and considerations

The benchmark was motivated by the latest large collection of human-drawn sketches built by Eitz et al. [6]. To explore human sketch recognition and how humans draw sketches, they collected 20,000 human-drawn sketches, categorized into 250 classes, each with 80 sketches. This sketch dataset is exhaustive in terms of the number of object categories. Thus, we believe that a 3D model retrieval benchmark based on their object categorizations will be more comprehensive and appropriate than other currently available 3D retrieval benchmarks to more objectively and accurately evaluate the real-world performance of a 3D model retrieval algorithm. In addition, the sketch dataset avoids the bias issue since it contains the same number of sketches for every class, and the number of sketches for one class is also adequate for a large-scale retrieval benchmark. Moreover, the sketch variation within one class is also sufficient.

SHREC13STB [5] has found 1,258 relevant models for 90 of the 250 classes from the **PSB** benchmark. However, it is neither complete nor large enough. 160 classes, i.e., the majority, have not been included. Thus, we felt a new 3D model retrieval benchmark based on Eitz et al.'s sketch dataset and **SHREC13STB**, but extended by finding more models from other 3D data sources, was needed. It is useful for the proper evaluation of sketch-based or model query-based 3D model retrieval algorithms, especially their scalability, which is very important in practice.

To this end, we built a unified large-scale benchmark supporting both sketch and model queries by extending **SHREC13STB** by means of identifying and consolidating relevant models for the 250 classes of sketches from the major prior 3D shape retrieval benchmarks. When creating the benchmark, our target was to find models for as many of the 250 classes as possible, and, for each class, to find as many models as possible. These previous benchmarks have been compiled with different goals in mind and, to date, have not been considered in combination. Our work is the first to integrate them to form a new, larger benchmark corpus for both Query-by-Model and Query-by-Sketch retrieval.

3.2. Building process

Based on the above considerations, to build up a better and more comprehensive large-scale 3D retrieval benchmark, we extend the search to eight available benchmarks. To avoid adding replicate models, aside from the **PSB** used in **SHREC13STB**, the other seven available 3D model benchmark sources we considered include the **SHREC12GTB**, **TSB**, **CCCC**, **WMB**, **MSB**, **BAB**, and **ESB**, as listed in Table 1.

We (one undergraduate student, one master student, one researcher with a master degree and one with a Ph.D. degree) adopted a voting scheme to classify models. For the classification of each model, we obtained at least two votes. If these two votes agree with each other, we confirm that the classification is correct; otherwise, we performed a third vote to finalize the classification. During the building process, we only kept one model for the models that have duplicate copies spanning different source datasets.

In the end, we found 13,680 sketches and 8,987 models, classified into 171 classes (for the remaining 79 classes we did not find relevant models in the selected benchmarks), which substantially increase the scale of the benchmark and form the currently largest unified retrieval benchmark. The average number of models in each class is 53, which is also much more than any of the benchmarks in Table 1. This benchmark provides an important resource for the community of 3D model retrieval and will likely foster the development of practical Query-by-Model and Query-by-Sketch 3D retrieval applications.

3.3. Unified large scale benchmark: ***LSB***

Our extended large-scale 3D model retrieval benchmark (**LSB**) [1] is motivated by the latest large collection of human-drawn sketches built by Eitz et al. [6] and the SHREC'13 Sketch Track Benchmark (**SHREC13STB**) [5]. The details of the benchmark are as follows.

3.3.1. 2D sketch dataset

The 2D sketch query set contains 13,680 sketches (171 classes, each with 80 sketches) from Eitz et al.'s [6] human sketch recognition dataset, each of which has relevant models in the selected 3D benchmarks. This sketch dataset was used as the 2D query sketch dataset in evaluating large scale sketch-based 3D shape retrieval algorithms in the SHREC'14 track on large scale sketch-based 3D shape retrieval [2].

3.3.2. 3D model dataset

In total, the 3D model dataset of the **LSB** benchmark contains 8,987 models classified into 171 classes. Each model is saved in the ".OFF" format as a text file. This 3D dataset was used in evaluating Query-by-Model 3D shape retrieval algorithms in the SHREC'14 track on comprehensive 3D shape retrieval [1]. It was also used as the target 3D model dataset in evaluating sketch-based 3D shape retrieval algorithms in the SHREC'14 track on extended large scale sketch-based 3D shape retrieval [2].

3.3.3. Ground truth

All the sketches and models are categorized according to the classifications in Eitz et al. [6] and the selected source benchmarks, respectively. In our classification and evaluation, we adopt the class names from Eitz et al. [6].

3.3.4. Training and testing subsets

To evaluate and compare the performance of both learning-based and non-learning based Query-by-Sketch 3D model retrieval algorithms, we randomly selected 50 sketches from each class for training and used the remaining 30 sketches per class for testing, while the 3D model dataset as a whole was used for both training and testing.

[1] The large-scale 3D model retrieval benchmark (**LSB**) is available at `http://www.itl.nist.gov/iad/vug/sharp/contest/2014/SBR/`.

3.4. Properties of the ***LSB*** benchmark

Table 2 lists the correspondences between the target 3D model dataset of **LSB** and its source benchmarks. The indexing and mapping relationship between our models and their original names in the source benchmarks, as well as and the name list of the 171 classes are available on the websites [1, 2]. The average number of vertices per model is 5,233. Though, on average, the number of models per class is 53, it ranges from only 1 (i.e., for the basket, cake, fire hydrant, giraffe, lion, owl, parking meter, parrot, penguin, tennis racket, and van classes) to more than 600 (i.e., the chair and table classes have 632 and 601 models, respectively). The 79 classes that we did not find relevant models for are listed in Table 3. As can be seen, quite a few of them are either only parts (i.e., arm, eye, mouth, foot, and feather), or less representative or common to see (i.e., angel, boomerang, crane, mermaid, and pretzel), or relatively professional (i.e. harp, saxophone, and trombone). Therefore, the 171 classes for which we have found relevant models in the eight major 3D benchmarks are more representative and, as a whole, cover the majority of normal objects that appear in our lives.

Note that in the area of image retrieval, benchmarks with millions of image objects [106] are considered large-scale by current standards. Often, these image benchmarks are obtained by crawling the web. In the 3D object case, compiling publicly available object repositories of large size is still a challenge. While a lot of 3D content is available in private and commercial repositories, the number of unique 3D objects freely available on the web is limited. Hence, million-sized 3D object benchmarks are not yet realistic. We therefore consider our **LSB** benchmark large in the sense that it is based on freely available and carefully compiled content. Eventually, this situation may change due to wider availability and easy-to-use 3D acquisition technology (see also Section 7).

3.5. Evaluation metrics

3.5.1. General evaluation metrics

To perform a comprehensive evaluation of a retrieval algorithm based on either a sketch or model query, we employed seven commonly used performance metrics [7, 1, 2] in Information Retrieval Evaluation that are also widely used in the 3D model retrieval field. They are Precision-Recall (PR) diagram, Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), E-Measures (E), Discounted Cumulated Gain (DCG) [7], and Average Precision (AP) [54]. We have developed code [1, 2] to compute all of these metrics. Their meaning and definitions are listed below.

- **Precision-Recall plot (PR):** Assume there are n models in the dataset, precision P is to measure the accuracy of the relevant models among the top K ($1 \leq K \leq n$) ranking results, while recall R is the percentage of the relevant class that has been retrieved in the top K results.

- **Nearest Neighbor (NN):** NN is the precision of the top most model.

Table 2: Composition of the 8,987 target 3D models in terms of the eight generic or specialized 3D model retrieval benchmarks: the number of used models and its percentages.

Benchmarks	Generic				Non-rigid		Professional	
	PSB	**SHREC12GTB**	**TSB**	**CCCC**	**WMB**	**MSB**	**BAB**	**ESB**
#Used models	1,371	940	4,617	382	44	367	1,239	27
Used percentage	75.6%	78.3%	46.2%	80.8%	11.0%	80.3%	54.9%	3.1%
LSB percentage	15.3%	10.5%	51.4%	4.3%	0.5%	4.1%	13.8%	0.3%
Domain percentage	81.3%				4.6%		14.1%	

Table 3: Seventy-nine remaining classes without relevant models in the selected benchmarks.

angel	arm	backpack	bell	binoculars	boomerang	bottle opener	bulldozer	cactus	calculator
canoe	carrot	cat	cloud	comb	computer mouse	crane machine	crown	donut	envelope
eye	feather	flashlight	foot	frying pan	grenade	hamburger	harp	head phones	hedgehog
hot dog	ipod	lobster	loudspeaker	megaphone	mermaid	moon	mosquito	mouse (animal)	mouth
nose	panda	paper clip	parachute	pigeon	pineapple	pizza	power outlet	present	pretzel
purse	radio	rainbow	revolver	rollerblades	rooster	Santa Claus	saxophone	snail	snowboard
socks	speed boat	sponge bob	squirrel	strawberry	streetlight	sun	swan	T-shirt	tiger
tomato	toothbrush	tractor	trombone	trousers	trumpet	walkie-talkie	wheelbarrow	zebra	

- **First Tier (FT):** Assume there are C relevant models in the database, FT is the recall of the top C-1 (for Query-by-Model retrieval, excluding the query model itself) or the top C (for Query-by-Sketch retrieval) retrieved models.

- **Second Tier (ST):** Similarly, ST is the recall of the top 2(C-1) (for Query-by-Model retrieval) or the top 2C (for Query-by-Sketch retrieval) retrieved models.

- **E-Measure (E):** Since generally people are more interested in the retrieval results on the first page, E-Measure is defined [7] to measure the composite retrieval performance of both precision and recall of the top 32 retrieved models (that is, the exact results that usually can be shown within one page),

$$E = \frac{2}{\frac{1}{P} + \frac{1}{R}}. \qquad (1)$$

- **Discounted Cumulated Gain (DCG):** The positions where the relevant models appear in the retrieval list are important since people are more interested in the models in the front part of the list. DCG is therefore defined as the normalized summed weighted value about the positions of the relevant models. To compute DCG, the retrieval list R is first transformed into a vector G, where G_i=1 if R_i is a relevant model, otherwise G_i=0. Then, DCG is computed according to the following equation:

$$DCG_i = \begin{cases} G_1 & i = 1, \\ DCG_{i-1} + \frac{G_i}{\lg_2 i} & \text{otherwise.} \end{cases}$$

Finally, it is normalized by its optimum:

$$DCG = \frac{DCG_n}{1 + \sum_{j=2}^{C} \frac{1}{\lg_2 j}}. \qquad (2)$$

- **Average Precision (AP):** AP is used to measure the overall performance. It is computed as the total area under the Precision-Recall curve. Therefore, it combines both precision and recall.

We need to mention that, for the seven metrics above, a higher value indicates better performance.

3.5.2. *Weighted evaluation metrics*

Besides the common definitions of the evaluation metrics, we also have developed two weighted versions for the benchmark by incorporating the model variations in each class. Basically, we use the number of available models to define the model variation. We assume there is a linear correlation between the number of available models in one class and the degree of variation of the class. Therefore, we adopt a weight based on the number of models or its reciprocal to define each weighted performance metric.

The proportionally m_p and reciprocally m_r weighted metrics (m=NN/FT/ST/E/DCG/AP) are defined as follows.

$$m_p = \frac{\sum_{i=1}^{M} n_i \cdot m_i}{\sum_{i=1}^{M} n_i}, \qquad (3)$$

$$m_r = \frac{\sum_{i=1}^{M} \frac{1}{n_i} \cdot m_i}{\sum_{i=1}^{M} \frac{1}{n_i}}, \qquad (4)$$

where M is the total number of model/sketch queries, n_i is the size of the class to which the i^{th} query belongs, and m_i is the non-weighted NN/FT/ST/E/DCG/AP metric value for the i^{th} query. m_p assigns bigger weights to the classes with more variations. In contrast, m_r highlights the overall performance in retrieving diverse classes by assigning bigger weights to the classes with few models/variations. It is also intended to avoid the bias on the performance evaluation because of the different number of models in different classes.

4. Contributors

The first five authors of this paper built the above benchmark and organized the SHREC'14 tracks on the topics of large scale comprehensive and sketch-based 3D model retrieval as well as this follow-up study. Information about the other contributors of the two tracks is listed next.

4.1. Query-by-Model retrieval

There are five groups who have successfully participated in the SHREC'14 Comprehensive 3D Shape Retrieval track. In total, they have submitted fourteen dissimilarity matrices. In addition, a new group (Zhang et al.) has contributed seven new methods and the organizers also ran the PANORAMA [53] method on our benchmark based on the publically available executable [107]. Below are details about the contributors and their twenty-two runs.

- *CSLBP-Run-1*, *CSLBP-Run-2*, *CSLBP-Run-3*, *HSR-DE* and *KVLAD* submitted by Masaki Aono, Nihad Karim Chowdhury, Hitoshi Koyanagi, and Ryuichi Kosaka from Toyohashi University of Technology, Japan (Section 5.1.1)

- *DBNAA_DERE* submitted by Qiang Chen and Bin Fang from Chongqing University, China (Section 5.1.2)

- *BF-DSIFT*, *VM-1SIFT*, *MR-BF-DSIFT*, *MR-D1SIFT* and *MR-VM-1SIFT* submitted by Takahiko Furuya and Ryutarou Ohbuchi from the University of Yamanashi, Japan (Section 5.1.3)

- *ZFDR* submitted by Bo Li and Yijuan Lu from Texas State University, USA; and Henry Johan from Fraunhofer IDM@NTU, Singapore (Section 5.1.4)

- *DBSVC* and *LCDR-DBSVC* submitted by Atsushi Tatsuma and Masaki Aono from Toyohashi University of Technology, Japan (Section 5.1.5)

- *MSD, SDS, MFF-EW, SHELL, SECTOR, SECSHELL, and D2* submitted by Chaoli Zhang, Haisheng Li, and Yajuan Wan from the Beijing Technology and Business University, China (Section 5.1.6)

- *PANORAMA [53]* submitted by the organizers based on the results from the publicly available executable [107]

4.2. Query-by-Sketch retrieval

Four groups have participated in the SHREC'14 track on Extended Large Scale Sketch-Based 3D Shape Retrieval. Twelve rank list results (runs) for six different methods developed by four groups have been submitted. The participants and their runs are listed next.

- *BF-fGALIF*, *CDMR (σ_{SM}=0.1, α=0.6)*, *CDMR (σ_{SM}=0.1, α=0.3)*, *CDMR (σ_{SM}=0.05, α=0.6)*, and *CDMR (σ_{SM}=0.05, α=0.3)* submitted by Takahiko Furuya and Ryutarou Ohbuchi from the University of Yamanashi, Japan (Section 5.2.1)

- *SBR-VC (α=1)* and *SBR-VC ($\alpha = \frac{1}{2}$)* submitted by Bo Li and Yijuan Lu from Texas State University, USA; Henry Johan from Fraunhofer IDM@NTU, Singapore; and Martin Burtscher from Texas State University, USA (Section 5.2.2)

- *OPHOG* and *SCMR-OPHOG* submitted by Atsushi Tatsuma and Masaki Aono from Toyohashi University of Technology, Japan (Section 5.2.3)

- *BOF-JESC (Words800_VQ)*, *BOF-JESC (Words1000_VQ)*, and *BOF-JESC (FV_PCA32_Words128)* submitted by Changqing Zou from the Chinese Academy of Sciences, China; Hongbo Fu from the City University of Hong Kong, China; and Jianzhuang Liu from Huawei Technologies Co. Ltd., China (Section 5.2.4)

To provide an even better overview of the twenty-six evaluated 3D model retrieval algorithms, we classify them in Table 4 based on the following taxonomy: type of feature (e.g., view-based, geometric, or hybrid), feature coding/matching methods (e.g., direct feature matching (DFM), Bag-of-Words (BoW) or Bag-of-Features (BoF) framework, super-vector coding (SVC), or sparse coding (SC)), learning scheme (e.g., manifold learning (MR), supervised learning (SL), unsupervised learning (USL), or deep learning (DL)), and semantic information (e.g., usage of classification or label information). However, since 3D model retrieval methods have become more and more complex due to involvement of different local/global/hybrid features, diverse feature coding methods and various machine learning strategies or semantic information are being used, making it difficult to provide both a descriptive and a compact taxonomy to classify and differentiate 3D model retrieval algorithms.

We also need to mention that each method has some parameter settings, which can be found in the following section on method description.

5. Methods

5.1. Query-by-Model retrieval methods

5.1.1. Hybrid shape descriptors CSLBP*, HSR-DE, and KVLAD, by M. Aono, N.K., Chowdhury, H. Koyanagi, and R. Kosaka

We have investigated accurate 3D shape descriptors over the years for massive 3D shape datasets. In the Large Scale Comprehensive 3D Shape Retrieval track, we have attempted to apply three different methods with five runs. Note that all the five runs, we apply pose normalization [85] as preprocessing.

For the first three runs, we applied CSLBP*, a hybrid shape descriptor, composed of **C**enter-**S**ymmetric **L**ocal **B**inary **P**attern (CSLBP) feature [108], **E**ntropy descriptor [109], and optional Chain Code (CC). The difference between the three runs comes from the number of view projections and the existence of the optional CC: 16 views for CSLBP in Run-1, 24 views for CSLBP in Run-2 and Run-3, while no CC for Run-1 and Run-2 and CC addition in Run-3. CSLBP* is computed by first generating depth buffer images from multiple viewpoints

Table 4: Classification of the twenty-six evaluated methods. When classifying Query-by-Sketch methods, we refer to [10] for "Feature type": local or global 2D feature. DFM: direct feature matching, BoW: Bag-of-Words, SVC: super-vector coding, BoF: Bag-of-Features, SL: supervised learning, MR: manifold ranking, LCDR: Locally Constrained Diffusion Ranking, CDMR: Cross-Domain Manifold Ranking.

Index	Evaluated method	Feature type	Feature coding/matching	Learning scheme	Semantic information	Section	Reference(s)
			Query-by-Model				
1	**CSLBP**	hybrid	DFM	no	no	5.1.1	[108, 109]
2	**HSR-DE**	hybrid	DFM	no	no	5.1.1	[110]
3	**KVLAD**	view-based	DFM	SL	yes	5.1.1	[49, 50]
4	**DBNAA_DERE**	hybrid	DFM	no	no	5.1.2	[111]
5	**BF-DSIFT**	view-based	BoW	no	no	5.1.3	[96, 112, 113]
6	**VM-1SIFT**	view-based	DFM	no	no	5.1.3	[96, 112]
7	**MR-BF-DSIFT**	view-based	BoW	MR	no	5.1.3	[96, 112, 113, 114]
8	**MR-D1SIFT**	view-based	BoW + DFM	MR	no	5.1.3	[96, 112, 113, 114]
9	**MR-VM-1SIFT**	view-based	DFM	MR	no	5.1.3	[96, 112, 114]
10	**ZFDR**	hybrid	DFM	no	no	5.1.4	[54]
11	**DBSVC**	view-based	SVC	no	no	5.1.5	[115, 116]
12	**LCDR-DBSVC**	view-based	SVC	MR (LCDR)	no	5.1.5	[115, 116, 117]
13	**MFF-EW**	hybrid	DFM	no	yes	5.1.6	[118, 119, 79]
14	**MSD**	geometric	DFM	no	no	5.1.6	[58]
15	**SDS**	geometric	DFM	no	no	5.1.6	[17]
16	**SHELL**	geometric	DFM	no	no	5.1.6	[17]
17	**SECTOR**	geometric	DFM	no	no	5.1.6	[17]
18	**SECSHELL**	geometric	DFM	no	no	5.1.6	[17]
19	**D2**	geometric	DFM	no	no	5.1.6	[58]
20	**PANORAMA**	hybrid	DFM	no	no	2.1.4	[53]
			Query-by-Sketch				
21	**BF-fGALIF**	local	BoW	no	no	5.2.1	[120, 10]
22	**CDMR**	local	BoW	MR (CDMR)	no	5.2.1	[120, 10]
23	**SBR-VC**	global	DFM	no	no	5.2.2	[121, 5, 10]
24	**OPHOG**	local	DFM	no	no	5.2.3	[122]
25	**SCMR-OPHOG**	local	DFM	MR (SCMR)	no	5.2.3	[122, 123, 117]
26	**BOF-JESC**	local	BoF	no	no	5.2.4	[124, 125, 126]

for a given 3D shape object, then by analyzing gray-scale intensities to produce three-resolution level histograms (in our implementation, 256×256, 128×128, and 64×64), having 16 bins each, after segmenting each depth-buffer image into sub-images (16, 8, 4, respectively). In addition to CSLBP, we have augmented it with "Entropy", trying to capture the randomness of surface shapes, resulting in CSLBP*.

For the fourth run, we applied HSR-DE, another hybrid shape descriptor, composed of multiple Fourier spectra obtained by Hole, Surface-Roughness, Depth-buffer, Contour, Line, Circle, and Edge images, an extension to the method we published in [110]. Figure 3 illustrates the method adopted in Run-4.

For the fifth run, we applied KVLAD, a supervised learning method we developed by combining non-linear scale space [49] with the Vector of Locally Aggregated Descriptor (VLAD) [50]. For the training stage, we employ SHREC2011 data and generate a code book of size 500, which is used for distance computation during the testing stage.

KVLAD is a combination of the KAZE local feature [49], which is supposed to be free from blurring along the sharp edge, with the location sensitive encoding scheme VLAD to produce "Visual Features", which was introduced by Jégou et al. [50]. VLAD differs from the histogram-based bag of visual words (BoVW) model in that it maintains the residual vector during the encoding procedure of visual features. VLAD can be represented by the following formula:

$$\mathbf{v}_i = \sum_{\mathbf{x} \in \Gamma_i} (\mathbf{x} - \mathbf{c}_i), \tag{5}$$

where $i = 1, 2, ..., K$, $\mathbf{c}_i$ is the centroid of the i-th cluster Γ_i, and $\mathbf{x}$ is a local feature in the cluster Γ_i. Each element of vector $\mathbf{v}_i$ has the same dimension of local features. Assume that we have d dimensional local features, then plain VLAD can be regarded as a $d \times K$ dimensional matrix. Although Jégou et al. suggest that dimension reduction of plain VLAD works reasonably well, we keep all the data as they are. The KVLAD visual feature is represented by the following:

$$V \equiv [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_K]. \tag{6}$$

Dissimilarity computation is carried out such that we compute Euclidean distance between the visual features extracted from

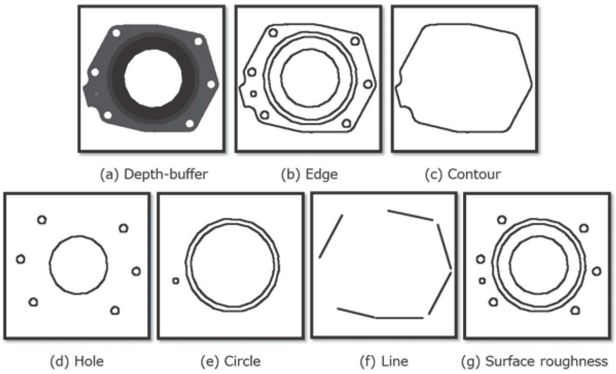


Figure 3: An example of HSR-DE (Hole and Surface-Roughness descriptors with Depth-buffer and Edge features augmented) before conversion to Fourier spectra.

a query and the visual features of each 3D model. Assume that a visual feature for a query is given by **Q**, and an arbitrary 3D model is given by **V**. The distance or the dissimilarity between them is computed as follows:

$$dist(\mathbf{Q},\mathbf{V}) = \sqrt{\sum_{i=1}^{K}\sum_{j=1}^{d}(Q_{i,j} - V_{i,j})^2}. \tag{7}$$

The search results computed from the above equation are ranked in ascending order.

5.1.2. 3D model retrieval descriptor DBNAA_DERE, by Q. Chen and B. Fang [111]

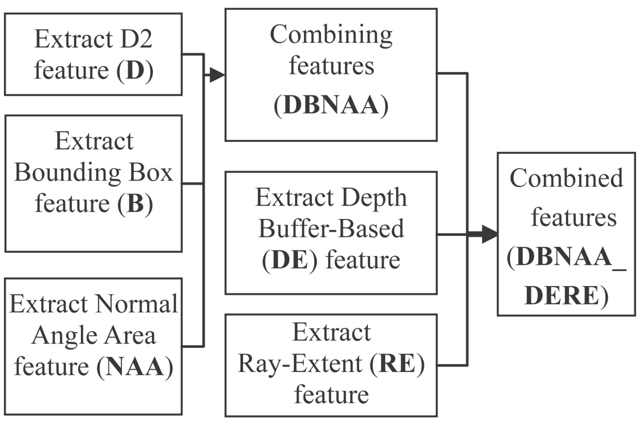


Figure 4: DBNAA_DERE feature extraction procedure.

We propose a combined 3D model feature named DBNAA_DERE which contains five different features: D2 [58], Depth Buffer images (DE) feature, Ray Extent (RE) [59] feature, Bounding Box feature, and Normal Angle Area feature. Based on the analysis on model surfaces, for each vertex we compute the mean angle and the average area of its adjacent faces and then use them to form a joint 2D histogram distribution, which we name Normal Angle Area feature. Then, we extract the D2 [58] feature and Bounding Box feature for each model, followed by linearly combining all the three features together based on fixed weights to form a new feature named D2 Bounding Box Normal Area feature (DBNAA) [111]. At last, we combine our DBNAA feature with Depth Buffer (DE) [59] and Ray Extent (RE) [59] features to build a more powerful feature named DBNAA_DERE [111]. Figure 4 shows the feature extraction procedure.

(1) DBNAA feature extraction. DBNAA comprises three components: D2 feature, Bounding Box feature and Normal Angle Area feature. The well-known D2 feature is first introduced by Osada et al. [58]. Here we use D2 as a component of our combined feature, and choose the parameters as follows: N=1024 samples and B=1024 bins, which means we sample N=1024 sample points and divide the histogram into 1024 bins. Finally, we have a 1024-dimensional vector to represent each model.

Bounding Box feature of a model is extracted after applying Continuous Principle Component Analysis (CPCA) [59] on it for pose normalization.

$$L = \{Z_{max} - Z_{min}, Y_{max} - Y_{min}, X_{max} - X_{min}\}, \tag{8}$$

$$F_{BB} = \{\frac{rank(L,1)}{rank(L,2)}, \frac{rank(L,2)}{rank(L,3)}\}, \tag{9}$$

where Z_{max}/Z_{min} is the maximum/minimum value of the z-axis coordinates of all the vertices of the model. Similar are with Y_{max}/Y_{min} and X_{max}/X_{min}. $rank()$ is a function to sort the vector in ascending order, $rank(L,1)$ means the first number in the sorted vector L. Finally, we get a two-dimensional vector F_{BB} to represent the Bounding Box feature of the model.

NAA feature is based on the mean angle A and average area S of each vertex,

$$A = \frac{1}{N_{vj}} \sum_{\{n_i,n_j\}\subset F_{vj}} n_i \cdot n_j, \tag{10}$$

$$S = \frac{1}{N_{vj}} \sum_{i=1}^{N_{vj}} S_i, \tag{11}$$

where N_{vj} is the number of adjacent faces of the j-th vertex. F_{vj} is a set of the normals of the adjacent faces of the j-th vertex, while n_i/n_j is the normal of face i/j. S_i is the area of the i-th face, and S is the average area of the adjacent faces. An illustration to demonstrate the A and S joint distribution can be found in [111]. After obtaining the mean angle A and average area S, we can use them to form a joint 2D distribution histogram, where both A and S are divided into N bins. N is empirically set to be 16. NAA feature is therefore an N*N feature matrix. According to our experiments, NAA feature is suitable to differentiate models with similar D2 features.

After getting the above three types of features, we combine the three features as below,

$$d_{DBNAA} = \alpha * d_D + \beta * d_B + (1 - \alpha - \beta) * d_{NAA}, \tag{12}$$

where α and β are set as follows: α=0.65, and β=0.15 according to our experiments on the SHREC'12 Track: Generic 3D Shape Retrieval [96] dataset. d_D is a scalar, which means the ℓ_1-norm D2 distance of two models. d_B and d_{NAA} are the Bounding Box and Normal Angle Area feature distance, respectively. We need to mention that when combining features we should first normalize different feature distances, which can be found in [111].

(2) DBNAA_DERE feature combination. Inspired by the idea proposed in Li and Johan [54], we also integrate the Depth Buffer-based (DE) and Ray-Extent (RE) [59] features by adopting a similar framework as DBNAA:

$$d_{DBNAA\_DERE} = \alpha * d_{DBNAA} + \beta * d_{DE} + (1 - \alpha - \beta) * d_{RE}. \quad (13)$$

We set α=0.3 and β=0.35, which are similarly based on the experiments on the SHREC'12 Track: Generic 3D Shape Retrieval [96] dataset.

Since the label information for the test dataset of the benchmark is assumed unknown for the purpose of benchmarking, our class information-based retrieval method is not applicable here. For more details about the shape descriptor computation, please refer to [111].

5.1.3. Visual feature combination for 3D model retrieval, by T. Furuya and R. Ohbuchi

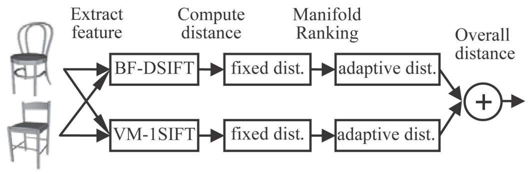


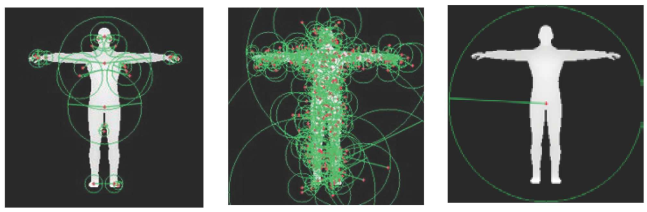
Figure 5: Two feature-adaptive distances computed from two visual features (BF-DSIFT and VM-1SIFT) are fused by summation.

Our algorithm is essentially the same as the one described in [96] and [112]. Figure 5 illustrates overall processing flow of the algorithm. It starts with multi-viewpoint rendering of 3D models, followed by extraction of a global visual feature and a set of local visual features from an image rendered from a view. A distance between a pair of 3D models is computed as a sum of distances learned from two distinct features.

Our algorithm employs a view-based approach for it is able to compare 3D models in almost any shape representations, e.g., polygon soup, open mesh, or point cloud. A set of local features aggregated by using Bag-of-Features (BF) approach (BF-DSIFT below) is known to attain certain invariance against articulation of 3D shapes, e.g., bending of joints. Such a feature, however, is incapable of distinguishing differences among rigid shapes, e.g, pipes bent in U shape and in S shape. Thus, a fusion of an aggregated local feature, which is insensitive to deformation or articulation, with a global feature sensitive to global deformation and articulation (VM-1SIFT below) could improve overall accuracy.

Visual feature extraction. Our method first renders a 3D model into range images from multiple viewpoints spaced uniformly in solid angle space. For the SHREC'14 Comprehensive 3D Shape Retrieval track, we used 42 viewpoints. Image resolution for each range image is 256×256 pixels. Then the algorithm extracts a set of local visual features, Dense SIFT (DSIFT) [113], from each range image. The algorithm also extracts a global visual features, One SIFT (1SIFT) [112] from a range image.

For DSIFT visual feature extraction, we randomly and densely sample feature points on the range image with prior to concentrate feature points on or near 3D model in the image (see Figure 6 (b)). From each feature point sampled on the image, we extract SIFT [127], which is a multi-scale, rotation-invariant local visual feature. The number of feature points per image is set to 300 as in [113], resulting in about 13k DSIFT features per 3D model. The set of dense local features are aggregated into a single feature vector per 3D model by using the BF approach. We use the ERC-Tree algorithm [128] to accelerate both codebook learning (clustering of local features) and vector quantization of local features into visual words. A frequency histogram of vector-quantized DSIFT features becomes a Bag-of-Features DSIFT, or BF-DSIFT feature vector for the 3D model.



(a) Original SIFT [127] (b) DSIFT (c) 1SIFT

Figure 6: Our method combines dense local visual feature (DSIFT) and global visual feature (1SIFT).

For 1SIFT extraction, we sample a feature point at the center of the range image and extract a SIFT feature from a large region covering the entire 3D model (see Figure 6 (c)). The number of 1SIFT per model is equal to the number of rendering viewpoints, i.e., 42. Note that the set of 1SIFT features is not BF-aggregated but is compared per-feature (i.e., per-view). Thus, the matching algorithm by using 1SIFT is called per-View Matching 1SIFT (VM-1SIFT).

Distance computation. Our method uses two different distance metrics for retrieval ranking; (1) fixed distance and (2) feature-adaptive distance learned by using Manifold Ranking (MR) algorithm [114].

(1) Fixed distance. Symmetric version of Kullback-Leibler Divergence (KLD) is used as fixed distance metric. KLD performs well when comparing a pair of probability distributions, i.e., histograms. For the BF-DSIFT, the distance between a pair of 3D models $\mathbf{x}_i$, $\mathbf{x}_j$ is equivalent to KLD between BF-DSIFT feature vectors of the two models (Equation (14)). For the VM-1SIFT, the distance between a pair of 3D models is calculated by using Equation (15) where N_v is the number of 1SIFT features per model and x_{ip} is 1SIFT feature extracted from the view p of 3D model x_i.

$$d_{BF-DSIFT}(\mathbf{x}_i, \mathbf{x}_j) = d_{KLD}(\mathbf{x}_i, \mathbf{x}_j), \quad (14)$$

$$d_{VM-1SIFT}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{p=1}^{N_v} \min_{1\le q\le N_v} d_{KLD}(\mathbf{x}_{ip}, \mathbf{x}_{jq}). \quad (15)$$

(2) Feature-adaptive distance. To improve distance metric among 3D models, we compute feature-adaptive distances on a manifold of 3D model features. To do so, we apply the MR algorithm to each of the BF-DSIFT feature manifold and the VM-1SIFT feature manifold. For each feature, we first generate a $N_m \times N_m$ affinity matrix $\mathbf{W}$ where N_m is the number of 3D models (N_m=8,987 for Query-by-Model retrieval on **LSB**) and $\mathbf{W}_{ij}$ indicates similarity between a pair of 3D models x_i, x_j. $\mathbf{W}_{ij}$ is computed by using the following equation,

$$\mathbf{W}_{ij} = \begin{cases} \exp(-\frac{d(\mathbf{x}_i,\mathbf{x}_j)}{\sigma}) & \text{if } i \ne j, \\ 0 & \text{otherwise,} \end{cases}$$

where d is fixed distance of either BF-DSIFT (Equation (14)) or VM-1SIFT (Equation (15)).

We normalize $\mathbf{W}$ by computing $\mathbf{S} = \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$ where $\mathbf{D}$ is a diagonal matrix whose diagonal element is $\mathbf{D}_{ii} = \sum_j W_{ij}$.

We use the following closed form solution for the MR to find relevance values in $\mathbf{F}$ given "source" vector $\mathbf{Y}$. In the source vector $\mathbf{Y}$, an element corresponding to the query 3D model is set to 1 to serve as the source of diffusion, while the other elements corresponding to the database 3D models are set to 0. $\mathbf{F}_{ij}$ is the relevance value between 3D models i and j. A higher relevance means a higher similarity, or a smaller diffusion distance.

$$\mathbf{F} = (\mathbf{I} - \alpha\mathbf{S})^{-1}\mathbf{Y}. \quad (16)$$

We add prefix "MR-" before the feature comparison method to indicate MR-processed algorithms (MR-BF-DSIFT and MR-VM-1SIFT). For parameters, we use σ=0.005 and α=0.975 for MR-BF-DSIFT, and use σ=0.0025 and α=0.9 for MR-VM-1SIFT. To further improve retrieval accuracy, we combine diffusion distances of the two features. The diffusion distances of MR-BF-DSIFT and MR-VM-1SIFT are normalized and then summed with equal weight (MR-D1SIFT).

5.1.4. *Hybrid shape descriptor ZFDR, by B. Li, Y. Lu and H. Johan [54]*

The comprehensive 3D model dataset contains both generic and professional (e.g. CAD and architecture models), rigid and non-rigid, articulated and non-articulated, watertight and non-watertight models. Due to the variations in the types and robustness considerations in retrieval performance, we employ the hybrid shape descriptor ZFDR devised in [54] which integrates both visual and geometric information of a 3D model: **Z**ernike moments and **F**ourier descriptor features of 13 cube-based sample views; **D**epth information feature of 6 depth buffer views and **R**ay-based features based on ray shooting from the center of the model to its farthest surface intersection points. Visual information-based features (e.g., **Z** and **F**) have good performance in characterizing some classes like "sea animal", but for some other types of models like "car", depth buffer-based features (e.g., **D** and **R**) are better [83]. We optimally integrate the above four different but complementary features to formulate the hybrid shape descriptor ZFDR to increase its differentiation power.

Figure 7 illustrates the overview of the feature extraction process: 3D model normalization mainly utilizing Continuous Principle Component Analysis (CPCA) [59] and extraction of four component features **Z**, **F**, **D** and **R**. The details of the retrieval algorithm are described as follows.

(1) View sampling. As a tradeoff between efficiency and accuracy, the approach sets cameras on the 4 top corners, 3 adjacent face centers and 6 middle edge points of a cube to generate 13 silhouette views to represent a 3D model.

(2) Zernike moments and Fourier descriptors features (ZF). For each silhouette view, up to 10^{th} order Zernike moments [129] (totally 35 moments) and first 10 centroid distance-based Fourier descriptors [130] are computed to respectively represent the region-based and contour-based visual features of the the silhouette views of the 3D model.

(3) Depth information and Ray-based features (DR). To improve the versatility of the descriptor in characterizing diverse types of models, the depth buffer-based feature and ray-based with spherical harmonic representation feature developed by Vranic [59] are integrated into the hybrid shape descriptor. The executable files [59] are utilized to extract the 438-dimensional **D** and 136-dimensional **R** features.

(4) ZFDR hybrid shape descriptor distance. Scaled-ℓ_1 (scaling each component of two feature vectors by their respective ℓ_1-norm before computing the summed component-wise ℓ_1 distance metric) [59] or Canberra distance (computing the ℓ_1 component-wise distance between any two components of two feature vectors followed by normalizing it by their sum, followed by summing all the component-wise distances) [76] metric is first applied to measure the component distances d_Z, d_F, d_D, and d_R between two models. Then, the hybrid descriptor distance d_{ZFDR} is generated by linearly combining the four component distances.

(5) Distance ranking and retrieval list output. Sort the hybrid distances between the query model and all the models in the dataset in ascending order and then list the models accordingly.

Please refer to the original paper [54] for more details about the feature extraction and retrieval process.

5.1.5. *Unsupervised 3D model retrieval based on Depth Buffered Super-Vector Coding and Locally Constrained Diffusion Ranking, by A. Tatsuma and M. Aono*

Depth Buffered Super-Vector Coding. We propose a new 3D model feature known as Depth Buffered Super-Vector Coding (DBSVC), an approach categorized as a bag-of-features method [131, 113]. DBSVC extracts 3D model features from rendered depth buffer images using a super-vector coding method [115]. Figure 8 illustrates the generation of our proposed DBSVC feature. We first apply Point SVD, a pose normalization method developed previously by the authors [85]. Post pose normalization, we enclose the 3D model with a unit geodesic sphere. From each vertex of the unit geodesic sphere,

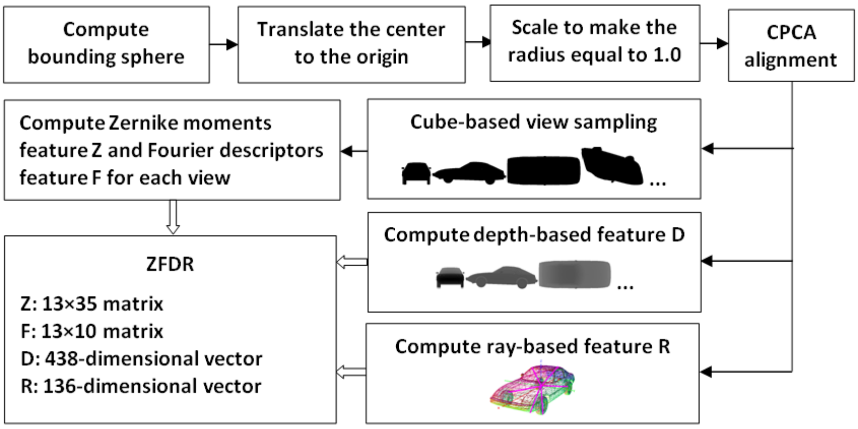


Figure 7: ZFDR feature extraction process [54].

we render depth buffer images with 300×300 resolution, and a total of 38 viewpoints are defined.

After image rendering, we extract local features from each depth buffer image. The SURF-128 descriptor is a well-known local feature vector with outstanding discrimination power [116]. The SURF-128 descriptor outperforms the regular SURF descriptor, but it turns more sparse. Thus, we apply the power and the ℓ_2 normalization, which diminish the sparseness of the SURF-128 descriptor, and call it the Power SURF descriptor. Moreover, we employ feature augmentation with patch coordinates [132]. The Power SURF descriptors are extracted from 98×98 pixel patches arranged every 5 pixels.

To calculate DBSVC, we generate a codebook of visual words in advance. The visual word is thus defined as the center of a cluster obtained by applying K-means clustering to the Power SURF descriptors, which are extracted from 3D models in the training dataset prepared by removing the decimated and the duplicated models from the NTU 3D Model Dataset (NMD) [37]. K-means clustering is performed with $K = 2048$.

We calculate DBSVC with the codebook of K visual words $\mathbf{v}_1, \ldots, \mathbf{v}_K$. Given a set of local features $\mathbf{x}_1, \ldots, \mathbf{x}_N$ extracted from a 3D model, let $a_{ki} = 1$ if $\mathbf{x}_i$ is assigned to $\mathbf{v}_k$ and 0 otherwise. For each $k = 1, \ldots, K$, we define,

$$b_k = \frac{1}{N} \sum_{i=1}^{N} a_{ki}, \tag{17}$$

$$c_k = c\sqrt{b_k}, \tag{18}$$

$$\mathbf{u}_k = \frac{1}{\sqrt{b_k}} \sum_{i=1}^{N} a_{ki}(\mathbf{x}_i - \mathbf{v}_k), \tag{19}$$

where c is a nonnegative constant and is chosen as 0.001 in our implementation. Then the DBSVC feature is obtained by,

$$\mathbf{f}_{DBSVC} = [c_1, \mathbf{u}_1^{\mathrm{T}}, \ldots, c_K, \mathbf{u}_K^{\mathrm{T}}]^{\mathrm{T}}. \tag{20}$$

To diminish the sparseness, the DBSVC feature is normalized using the power and the ℓ_2 normalization. We simply calculate the Euclidean distance for comparing DBSVC features between two 3D models.

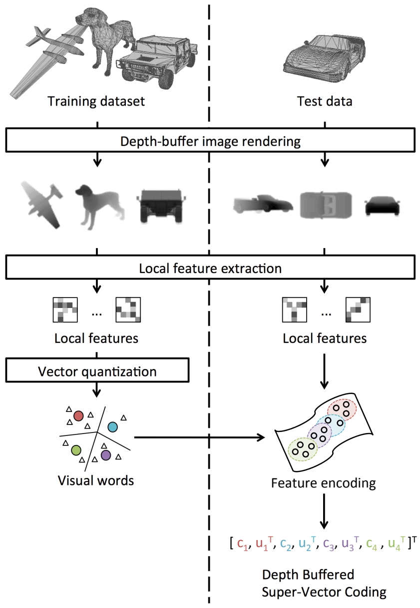


Figure 8: Overview of the Depth Buffered Super-Vector Coding.

Locally Constrained Diffusion Ranking. We calculate ranking scores using our modified manifold ranking algorithm. We use the Locally Constrained Diffusion Process (LCDP) [117] for calculating the affinity matrix in the manifold ranking algorithm [123], and call this method Locally Constrained Diffusion Ranking (LCDR). LCDP aims at capturing the geometric structure of data manifolds, reducing the effect of noisy data points.

Given a set of data points $\mathbf{f}_1, \ldots, \mathbf{f}_n$, the transition probability matrix on the k-nearest neighbor graph is defined by,

$$P = T^{-1}E, \tag{21}$$

where $E_{ij} = \exp(-\|\mathbf{f}_i - \mathbf{f}_j\|^2/\sigma^2)$ if $\mathbf{f}_j$ belongs to the k-nearest neighbors of $\mathbf{f}_i$ and $E_{ij} = 0$ otherwise, and $T_{ii} = \sum_j E_{ij}$. Furthermore, LCDP sets a high value to the transition probability between two data points if all the paths among their k-nearest neighbors are short. This property is implemented in the following update strategy,

$$W(t+1) = PW(t)P^T. \tag{22}$$

For the initial affinity matrix $W(0)$, we use a symmetrically normalized affinity matrix, which is defined as

$$W(0) = Q^{-1/2} A Q^{-1/2}, \tag{23}$$

where $A_{ij} = \exp(-\|\mathbf{f}_i - \mathbf{f}_j\|^2/\sigma^2)$ and $Q_{ii} = \sum_j A_{ij}$.

Our LCDR calculates ranking scores using the manifold ranking algorithm with the affinity matrix W obtained by LCDP. Given a column vector $\mathbf{y} = [y_1, \ldots, y_n]^T$ with $y_i = 1$ if $\mathbf{f}_i$ is a query and $y_i = 0$ otherwise, the ranking score vector $\mathbf{r} = [r_1, \ldots, r_n]^T$ in LCDR is defined by,

$$\mathbf{r} = (I - \alpha M)^{-1}\mathbf{y}, \tag{24}$$

where $M = D^{-1/2} W D^{-1/2}$, $D_{ii} = \sum_j W_{ij}$, and $\alpha \in [0, 1)$ is a tuning parameter.

LCDR allows to calculate the ranking scores, which capture more geometric structure of data manifolds than the conventional manifold ranking methods. However, LCDR requires much execution time because of calculating the matrix product repeatedly. We fixed the LCDR parameters through preliminary experiments with the Princeton Shape Benchmark [7]. We set k to 12, σ to 0.36, α to 0.99, and the maximum number of iterations to 10.

5.1.6. 3D shape retrieval based on MSD, SDS and MFF-EW, by C. Zhang, H. Li, Y. Wan

To accommodate the characteristics of the large-scale benchmark dataset, we adopt two highly time-efficient geometry-based retrieval algorithms, which are modified from Ankerst et al.'s Shape Histogram algorithm [17] and Osada et al.'s Shape Distribution (D2) algorithm (SD) [58]. In addition, to better represent the feature of each category dataset, the multi-feature fusion method based on entropy weight is adopted.

Modified Shape Distribution (MSD). To enhance the performance of the SD, we modify the 3D normalization part in the preprocessing step, and construct a cubic spline interpolation curve to represent the statistical shape distribution histogram.

(1) 3D model normalization and sampling. Firstly, we obtain a model's gravity center by accumulating the gravities of all the faces on the surface of the 3D model. Then, we translate the gravity center to the origin and scale the model to make the radius of its bounding sphere to be 1. Consequently, the D2 distance feature value is compressed into the range of [0, 2], which contributes to the scale invariance property of our algorithm. Finally, we randomly sample 1,024 sample points for each model. Figure 9 shows examples.

(2) Cubic spline interpolation curve construction. To better describe the statistical properties of a Shape Distribution histogram, a cubic spline interpolation curve with 1026 control points, instead of polynomial fitting or piecewise linear function [58], is used to represent the shape distribution. Some examples are listed in Figure 10.

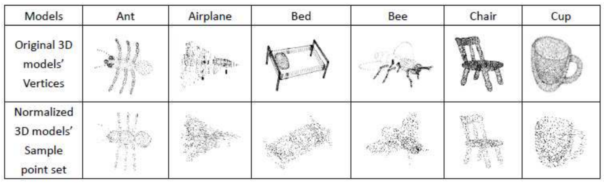


Figure 9: Example sample point sets for normalized 3D models.

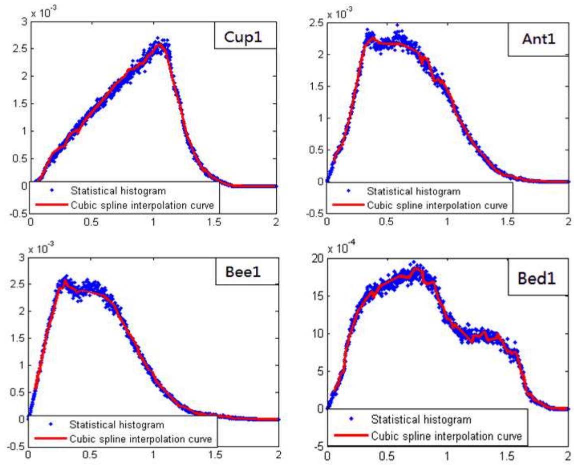


Figure 10: Example cubic spline interpolation curves used to represent the Shape Distribution histograms.

Shell-Distance-Sum (SDS) algorithm. 3D Shape Histogram algorithm [17] can be broadly divided into three types: SHELL, SECTOR and SECSHELL. Our SDS is based on SHELL and makes an improvement in the step of constructing the shape histogram. In our algorithm, we sum the distances between every point in each of 120 bins and the gravity center of the model to represent the feature of that bin, instead of counting the number of points falling into each bin. This improvement enables SDS to describe both the location and the magnitude information of the vertices on a 3D model. In addition, we normalize the 3D model first, as in the corresponding steps described in MSD.

Multi-Feature Fusion Based on Entropy Weights (MFF-EW). Considering the complementarity between the candidate features for fusion, we select the MSD and SDS features in our multi-feature fusion algorithm. We propose a novel multi-feature fusion algorithm by adaptively computing the fusion feature weights using entropy for each query, which is similar to [118, 119].

(1) Information entropy calculation based on a query result. The theoretical basis of this step is to characterize the differentiation ability of a 3D shape feature based on the information entropy of its retrieval results. We need to mention that the classification information of the benchmark is also needed in this step.

1) For each query model $q \in U$, where U represents the target 3D model dataset, we obtain the top k retrieved models R^f_{qk} when using the shape feature f. We set k=10 based on experimental results as well as by referring to the approach in [79].

2) Counting the number of models in the top k models that belong to the same category, denoted as R^f_{qki}, where $i = 1, 2, ...n$ and n is the number of categories. Then we calculate the probability distribution of R^f_{qki}, denoted as $\{p_1, p_2, ..., p_i, ..., p_n\}$,

$$p_i = \frac{R^f_{qki}}{R^f_{qk}} \tag{25}$$

3) Computing the entropy of R^f_{qk},

$$E(R^f_{qk}) = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i. \tag{26}$$

(2) Calculating the weight of feature. Based on the analysis of Step (1), a smaller entropy demonstrates that the corresponding 3D feature can better describe the models, and we should assign a large weight for it. Therefore, we formulate their relationship as follows,

$$W^f_{qk} = \frac{1 - E(R^f_{qk})}{m - \sum_{f=1}^{m} E(R^f_{qk})}. \tag{27}$$

where m is the total number of the 3D features, and $\sum_{f=1}^{m} W^f_{qk} = 1$.

(3) Computing fusion dissimilarity distance. First, we normalize each row of the dissimilarity distance matrices resulting from different features,

$$d^{f'}(i, j) = \frac{d^f(i, j) - \min_i}{\max_i - \min_i}, j = 1, 2, .., n, \tag{28}$$

where $d^f(i, j)$ and $d^{f'}(i, j)$ are the pre-normalized and normalized distances between model i and model j respectively, while $\max_i$ and $\min_i$ are the maximum and minimum distances in the i^{th} row. Finally, the fusion dissimilarity distance is,

$$D_{fusion}(i, j) = \sum_{f=1}^{m} d^{f'}(i, j) \cdot W^f_{qk}. \tag{29}$$

In the experiments, we also provide the performance of our implementations of the original D2, and three types of 3D Shape Histograms (SHELL, SECTOR and SECSHELL) as a baseline for reference.

5.2. Query-by-Sketch retrieval methods

5.2.1. Ranking on Cross-Domain Manifold for sketch-based 3D model retrieval, by T. Furuya and R. Ohbuchi

To compare a hand-drawn sketch to a 3D model, most of existing methods compare a sketch with a set of multi-view rendered images of a 3D model. However, there is a gap between sketches and rendered images of 3D models. As hand-drawn sketches contain "noise", such as shape abstraction, semantic influence, stylistic variation, and wobbly lines, these sketches are often dissimilar to rendered images of 3D models.

Our algorithm employs an unsupervised distance metric learning to partially overcome the gap between sketches and 3D models [10][120]. Our algorithm called Cross-Domain Manifold Ranking, or CDMR [120], tries to bridge the gap between features extracted in two heterogeneous domains, i.e., domain of sketches and domain of rendered images of 3D models. While the CDMR algorithm could perform in either an unsupervised, semi-supervised, or supervised mode, we use unsupervised CDMR in this paper.

Figure 11 shows an overview of the CDMR. It first creates two separate manifolds of features, i.e., a manifold of sketch features and a manifold of 3D model features. The feature manifolds are computed by using an algorithm best suited for each of the domains; BF-fGALIF [120] (slightly modified BF-GALIF [133]) is used to compare sketches and BF-DSIFT [113] is used to compare 3D models. These two feature manifolds are then inter-linked to form a Cross-Domain Manifold (CDM) by using an algorithm capable of sketch-to-3D comparison, that is, the BF-fGALIF. Using the CDM, similarity values between a sketch query and 3D models are computed by diffusing relevance on the CDM. The relevance originates from the query, and it diffuses towards 3D models via edges of the CDM by using a process identical to Manifold Ranking [123]. The higher the relevance value of a 3D model, the closer it is to the query.

Unlike previous sketch-to-3D model comparison algorithms, the CDMR tries to maintain manifolds of sketches and 3D models. This often positively contributes to ranking accuracy. Also, if a large enough number of sketches and their inter-similarity values are available, the CDMR performs a form of automatic query expansion on the manifold of sketches.

Forming a Cross Domain Manifold. A CDM is a graph, whose vertices are either sketches or 3D models. The CDM graph $\mathbf{W}$ is represented by a matrix having size $(N_s + N_m) \times (N_s + N_m)$, where N_s and N_m are the number of sketches and 3D models in a database respectively. For Query-by-Sketch retrieval on **LSB**, $N_s = 13{,}680$ and $N_m = 8{,}987$.

The element of the matrix $\mathbf{W}$, i.e., $\mathbf{W}_{ij}$, indicates similarity between a sketch (or a 3D model) i and a sketch (or a 3D model) j. (For details, please refer to [120].) Distances are computed for each pair of vertices i and j by using the feature comparison methods i.e., BF-fGALIF and BF-DSIFT. The distances are

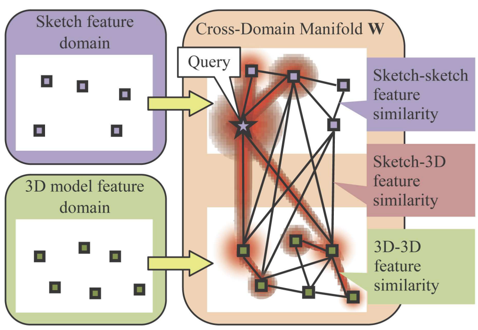


Figure 11: Feature comparison using Unsupervised Cross-Domain Manifold Ranking (CDMR).

then converted into similarities by using the following equation where $d(i, j)$ is the distance between vertices i and j.

$$\mathbf{W}_{ij} = \begin{cases} \exp(-d(i, j)/\sigma) & \text{if } i \neq j, \\ 0 & \text{otherwise.} \end{cases}$$

The parameter σ controls diffusion of relevance value across the CDM. We use different values σ_{SS}, σ_{MM}, and σ_{SM} to compute sketch-to-sketch similarity, 3D model-to-3D model similarity, and sketch-to-3D model similarity, respectively. These similarity values must be computed either by feature similarity or semantic similarity (if available.)

As mentioned above, sketch-to-3D model comparison uses BF-fGALIF algorithm [10][120], which is a slightly modified version of BF-GALIF [133]. BF-fGALIF compare a sketch and multi-view rendered images of a 3D model by using sets of Gabor filter-based local features. A 3D model is rendered into Suggestive Contour (SC) [134] images from multiple viewpoints. The sketch image and the SC images of the 3D model are rotation-normalized by using responses of multi-orientation Gabor filters computed of the image. After normalizing for rotation, fGALIF features are densely extracted from the image. The set of fGALIF features are integrated into a feature vector per image by using Bag-of-Features (BF) approach. A BF feature of the sketch is compared against a set of per-view BF features of the 3D model to find a distance between the sketch and the 3D model.

For sketch-to-sketch comparison, BF-fGALIF features are extracted from the sketches. Unlike the BF-fGALIF for sketch-to-3D model comparison, the BF-fGALIF for sketch-to-sketch comparison does not perform rotation normalization.

To compare 3D models, we use the BF-DSIFT [113] algorithm. It is also a view-based algorithm. A set of multi-scale, rotation-invariant local visual features is densely extracted from multi-view rendered range images of a 3D model. The set of local visual features is then BF-integrated per 3D model for comparison. A little more detail on the BF-DSIFT is found in Section 5.1.3.

Ranking on the Cross Domain Manifold. After generating $\mathbf{W}$ representing a CDM, Manifold Ranking (MR) algorithm [123] is applied on $\mathbf{W}$ to diffuse relevance value over the CDM from a query. We use the closed form of the MR (Equation (30)) to find relevance values in $\mathbf{F}$ given "source" matrix $\mathbf{Y}$. In Equation (30), $\mathbf{I}$ is an identity matrix and $\mathbf{S}$ is a symmetrically normalized matrix of $\mathbf{W}$ and α is a parameter. $\mathbf{F}_{ij}$ is the relevance value of the 3D model j given the sketch i. A higher relevance means a smaller distance.

$$\mathbf{F} = (\mathbf{I} - \alpha\mathbf{S})^{-1}\mathbf{Y}. \tag{30}$$

Using a naive algorithm, CDMR requires time complexity $O((N_s + N_m)^2)$ for generating the CDM graph $\mathbf{W}$ and $O((N_s + N_m)^3)$ for diffusing relevance over the CDM (Equation (30)). As shown in the experiments, computing CDMR is slower than other Query-by-Sketch retrieval algorithms. Among the parameters for the CDMR (i.e., σ_{SS}, σ_{MM}, σ_{SM} and α), we fixed σ_{SS} to 0.02 and σ_{MM} to 0.005 through preliminary experiments. For σ_{SM} and α), we tried the following combinations of the parameters; (σ_{SM}, α) = (0.1, 0.6), (0.1, 0.3), (0.05, 0.6), (0.05, 0.3).

5.2.2. *Efficient sketch-based 3D model retrieval based on view clustering and parallel shape context matching (SBR-VC) [121] [5] [10], by B. Li, Y. Lu, H. Johan, and M. Burtscher*

The SBR-VC algorithm first clusters a set of sample views of each model into an appropriate number of representative views according to its visual complexity, which is defined as the viewpoint entropy distribution of its sample views. Next, a parallel relative frame-based shape context (referred as relative shape context) matching [135] algorithm is employed to compute the distances between a 2D sketch and the representative silhouette views of a 3D model. Before retrieval, the relative shape context features of the representative views of all 3D target models are precomputed. Figure 12 presents an overview of the algorithm, which is described in more detail below.

Precomputation. **(1) Viewpoint entropy-based adaptive view clustering**. This clustering is performed in four steps. For each 3D model, the first step computes the viewpoint entropy of 81 views that are sampled by subdividing a regular icosahedron using the Loop subdivision [136] rule. The second step calculates the viewpoint entropy-based 3D visual complexity for each model. The mean and standard deviation entropies m and s of all sample views of each 3D model are computed first. The 3D visual complexity of each model is defined as

$$C = \sqrt{\frac{\widehat{s}^2 + \widehat{m}^2}{2}}, \tag{31}$$

where $\widehat{s}$ and $\widehat{m}$ are the entropies s and m normalized relative to their maximum and minimum over all the models. Hence, $C \in [0, 1]$. This metric has the ability to quantitatively measure the visual complexity difference between models belonging to different categories. In the third step, the visual complexity C of a 3D model is utilized to determine the number of representative views

$$N_c = \lceil \alpha \cdot C \cdot N_0 \rceil, \tag{32}$$

where α is a constant and N_0 is the number of sample views for each 3D model. N_0 is 81 in the presented SBR-VC algorithm. For large-scale retrieval, α is chosen as 1 or $\frac{1}{2}$, which corresponds to an average of 18.5 or 9.5 representative views, respectively, for each model in the dataset. The fourth step applies Fuzzy C-Means [137] view clustering to the viewpoint entropy values of the 81 sample views, together with their viewpoint locations, to generate the representative views for each model.

(2) Feature view generation. Outline feature views for the 2D sketches and the 3D models are generated. In the 3D case, silhouette views are first rendered followed by outline feature extraction. In the 2D case, silhouette views are generated based on binarization, Canny edge detection, closing (once), dilation (7 times in this case), and hole filling.

(3) Relative shape context computation. Rotation-invariant relative shape context features [135] are extracted to represent

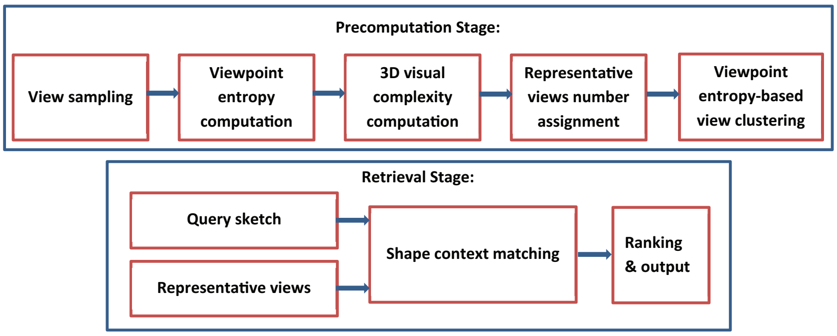


Figure 12: Overview of the SBR-VC algorithm: the first row is for the precomputation whereas the second row is for the retrieval stage [5] [10].

both sketches and sample views. 50 feature points are uniformly sampled for each outline feature view based on cubic B-Spline interpolation.

Online retrieval. With a 2D query sketch, a target 3D database, and the precomputed relative shape context features of the representative views of each model, the online retrieval algorithm works as follows.

(1) Sketch feature extraction. First, an outline feature view of the 2D sketch is generated. Then, its relative shape context features are computed in parallel within the following three steps: outline magnitude computation, log-polar histogram generation and normalization.

(2) 2D-3D distance computation. The relative shape context matching is performed between the sketch and each representative view of a model and the minimum 2D-3D matching cost is chosen as the sketch-model distance. The computation of 2D-3D distances between the sketch and all the 3D models is also performed in parallel.

(3) 2D-3D distance ranking. The sketch-model distances are sorted in ascending order and the models are ranked accordingly.

SBR-VC ($\alpha = 1$) and SBR-VC ($\alpha = \frac{1}{2}$) represent two runs of the SBR-VC algorithm with corresponding α values. The 70x performance speedup achieved over the serial code [5] is mainly due to the parallelization and code optimization of the relative shape context matching algorithm.

5.2.3. Unsupervised sketch-based 3D model retrieval based on Overlapped Pyramid of HOG and Similarity Constrained Manifold Ranking , by A. Tatsuma and M. Aono

Overlapped Pyramid of HOG. We propose a new feature vector known as Overlapped Pyramid of Histograms of Orientation Gradients (OPHOG) which is an extended version of the Pyramid of Histograms of Orientation Gradients [122] proposed in the field of image classification. An overview of the proposed OPHOG is illustrated in Figure 13. OPHOG divides an image into overlapped cells by stages, and extracts an orientation histogram from each cell.

We perform preprocessing to a 3D model and a sketch image before extracting OPHOG features as shown in Figure 14. In the preprocessing of the 3D model, we generate depth buffer images with 300×300 resolution from the 102 viewpoints that are composed of the vertices of a unit geodesic sphere. To obtain a sketch-like image, we apply Laplacian filtering, thinning transformation and Gaussian filtering to the depth buffer image. Similarly, in the preprocessing of the sketch image, we resize it to 300×300 resolution, and employ thinning transformation and Gaussian filtering.

After preprocessing, OPHOG divides a given image into cells using a regular sliding window determined by the spatial level. The window size w and stride size s are defined by the image size h and spatial level l as follows:

$$w = h/2^l, \quad s = w/2. \tag{33}$$

The OPHOG feature is obtained by concatenating all of the orientation histograms calculated for each cell. The orientation histogram is constructed by voting gradient magnitude to the corresponding orientation bin. The gradient magnitude g and orientation θ are defined as follows:

$$g(x,y) = \sqrt{u_x(x,y)^2 + u_y(x,y)^2}, \tag{34}$$

$$\theta(x,y) = \tan^{-1}\frac{u_x(x,y)}{u_y(x,y)}, \tag{35}$$

where,

$$u_x(x,y) = L(x+1,y) - L(x-1,y),$$
$$u_y(x,y) = L(x,y+1) - L(x,y-1),$$

and $L(x,y)$ denotes the image value at pixel (x,y).

Finally, to decrease the influence of the noise in a sketch image, we transform the OPHOG feature vector into its rank order vector and apply the ℓ_2 normalization.

During implementation, we set the number of histogram bins to 40 and limit the number of levels to 3. For comparing a sketch image to a 3D model, we calculate the minimum Euclidean distance, which is denoted by the following equation:

$$d(s,m) = \min_{i=1,\cdots,102} \|\mathbf{f}^{(s)} - \mathbf{f}_i^{(m)}\|, \tag{36}$$

where $\mathbf{f}^{(s)}$ is the feature vector of sketch image s, and $\mathbf{f}_i^{(m)}$ denotes the feature vector of the ith depth buffer image rendered from 3D model m.

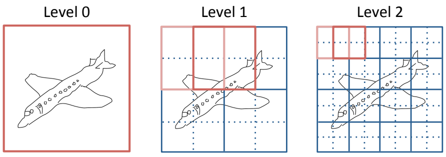


Figure 13: Overview of the Overlapped Pyramid of HOG.

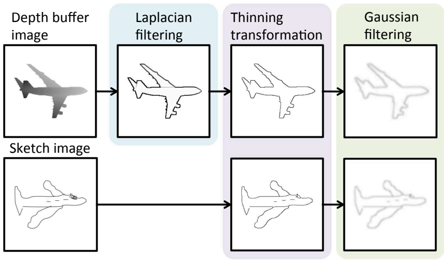


Figure 14: Preprocessing steps of the Overlapped Pyramid of HOG.

Similarity Constrained Manifold Ranking. We also propose an extended manifold ranking method [123] constrained by the similarity between a sketch image and a 3D model. In the following, we call this method Similarity Constrained Manifold Ranking (SCMR).

Suppose we have feature vectors of 3D model $\mathbf{f}_1, \ldots, \mathbf{f}_n$. SCMR aims to assign to each feature vector $\mathbf{f}_i$ a ranking score r_i which reflects the non linear structure of the data manifold. To reflect the data relations represented with the affinity matrix W within the ranking scores, we defined the following cost function:

$$\frac{1}{2}\sum_{i,j=1}^{n}\left(\frac{r_i}{\sqrt{D_{ii}}} - \frac{r_j}{\sqrt{D_{jj}}}\right)^2 W_{ij}, \qquad (37)$$

where $D_{ii} = \sum_j W_{ij}$. To preserve the similarity between a query sketch-image and a target 3D model in the ranking score, we add the following fitting constraint term:

$$\sum_{i=1}^{n}(r_i - z_i)^2, \qquad (38)$$

where $z_i = \exp(-d(s, m_i)^2/\sigma^2)$ is the similarity between the query sketch-image and ith target 3D model.

The optimal ranking score is obtained by minimizing following cost function:

$$J(r) = \frac{1}{2}\sum_{i,j=1}^{n}\left(\frac{r_i}{\sqrt{D_{ii}}} - \frac{r_j}{\sqrt{D_{jj}}}\right)^2 W_{ij} + \mu\sum_{i=1}^{n}(r_i - z_i)^2, \qquad (39)$$

where $\mu > 0$ is a regularization parameter. Differentiating $J(r)$ with respect to r and rearranging, we obtain

$$\mathbf{r} = (I - \alpha M)^{-1}\mathbf{z}, \qquad (40)$$

where $M = D^{-1/2}WD^{-1/2}$, $\mathbf{r} = [r_1, \ldots, r_n]^{\mathrm{T}}$, $\mathbf{z} = [z_1, \ldots, z_n]^{\mathrm{T}}$, and $\alpha \in [0, 1)$ is a tuning parameter. Clearly, the matrix $(I - \alpha M)^{-1}$ can be calculated off-line. The ranking score can be obtained by simple matrix-vector multiplication.

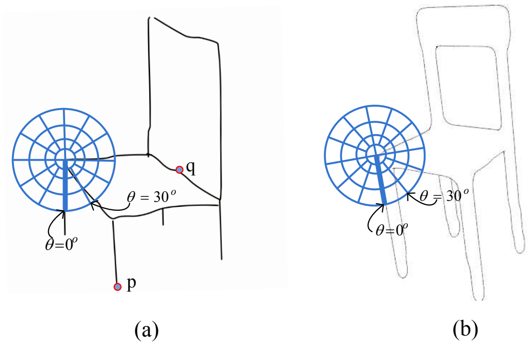


Figure 15: Illustration for the junction-based extended shape context feature descriptor. Two local patches on a junction of a query sketch and a model view are shown in (a) and (b), respectively.

In SCMR, we use the DBSVC as the feature vector for a 3D model. Furthermore, we calculate the affinity matrix using the LCDP [117]. We fixed the SCMR parameters through preliminary experiments with the SHREC'13 Sketch Track Benchmark [5]. For the SCMR, we set σ to 0.1 and α to 0.85. For the LCDP, we set the number of nearest neighbors to 10, the Gaussian width to 0.45, and the maximum number of iterations to 10.

5.2.4. BOF-JESC based descriptor, by C. Zou , H. Fu, and J. Liu

BOF-JESC follows the bag-of-features framework. It employs a junction-based extended shape context to characterize the local details within the four concentric circles centered at the key points. The motivation of the BOF-JESC descriptor comes from two aspects: 1) the local patch centered at a junction takes into account contour salience, hence can capture important cues for perceptual organization and shape discrimination, as discussed in [124], and 2) the local descriptor shape context [125] is tailored for the images in this work (i.e., the sketches or model views) since they only contain contours. It has been evaluated by [138] to have a high discrimination performance.

BOF-JESC extracts a global histogram for each image M (M denotes a binary image obtained from a query sketch/model view in this work). Edge point location in a local patch of BOF-JESC is quantized into 40 bins as shown in Fig. 15 (i.e. the number of points is recorded in each bin). In our experiments, the best performance is achieved by setting the radius of the log-polar coordinate to 0.075, 0.15, 0.25 and 0.35 of R_M ($R_M = \sqrt{W * H}$ where W and H is the width and height of the bounding box of M). The circle with the shortest radius is divided into four bins, as shown in Fig. 15, which is based on the fact that the bins with small areas are more sensitive to the statistics of the edge points.

The 40 dimensional local feature of BOF-JESC has the following characteristics:

- BOF-JESC selects all the junctions (we uses the method in [124] to extract the junctions in M, and the points with degree one, e.g. the point p in Fig. 15 (a), are also treated as junctions), and the mid-points in the lines connecting two adjacent junctions (e.g. the point q in Fig. 15 (a)) into the key-point set to generate local features;

- BOF-JESC aligns the reference axis with $\theta = 0$ of the log-polar coordinate system to the average direction of the tangent lines of the ten nearest points in the longest edge connecting the corresponding key-point, this step obtains a rotation invariance;
- BOF-JESC quantizes the edge points on the boundary of two neighboring bins into the bin with a greater angle (relative to the the reference axis in the anti-clockwise direction);
- BOF-JESC normalizes a 40 dimensional local feature with ℓ_1-norm regularization.

After the local features based on key-points are extracted from all the model views in a database, BOF-JESC employs K-means to obtain d "visual words" and finally builds a global ℓ_2-normalized histogram (i.e. a d dimensional feature vector) for each model view in the off-line stage.

5.2.5. *Implementation*

We sample 42 views for each 3D model uniformly on the unit viewpoint sphere. The vocabulary is obtained by the following steps: 1) concentrating the local features of all the model views in the database, 2) sampling 1 million local features from concentrated features, 3) utilizing KNN to obtain N words. The query-to-model distance metric is based on the nearest neighbor (NN) strategy, which finds the closest view to the query in the feature space, and treats such a minimum query-to-view distance as the query-to-model distance. The vocabulary sizes are set to 800 and 1000. Besides the standard framework of the bag-of-feature method using k-means, we also evaluate the performance of the Fisher Vector [126] combined with JESC features.

6. Results

6.1. *Query-by-Model retrieval*

In this section, we perform a comparative evaluation of the results of the twenty-two runs submitted by the seven groups based on the 3D target dataset of **LSB**. To provide a comprehensive comparison, we measure the retrieval performance based on the 7 metrics mentioned in Section 3.5: PR, NN, FT, ST, E, DCG, and AP, as well as the proportionally and reciprocally weighted NN, FT, ST, E, and DCG.

Figure 16 shows the Precision-Recall performance of the twenty-two runs whereas Figure 17 compares the best runs of each group. Tables 5 through 7 list the other six non-weighted and weighted performance metrics, together with their ranking orders (R). As can be seen from Figure 17 and Tables 5 through 7, Tatsuma's LCDR-DBSVC performs best, followed by Furuya's MR-D1SIFT. The top five methods are the same for the non-weighted and weighted performance metrics. We further find that the rank order in Table 7 is more similar to that in Table 5 than in Table 6, which shows that the reciprocally weighed metrics correlate better with the non-weighted definitions. However, because they also consider the difference in the number of models in different classes, they are more accurate in real applications. Based on the three jumps ahead in the ranking order of PANORAMA in Table 6, it can be deduced that it provides superior performance in retrieving classes with more variations. From this result, we can say that using view-based features in combination with advanced feature coding and adaptive ranking yields the best performance among the set of submitted methods.

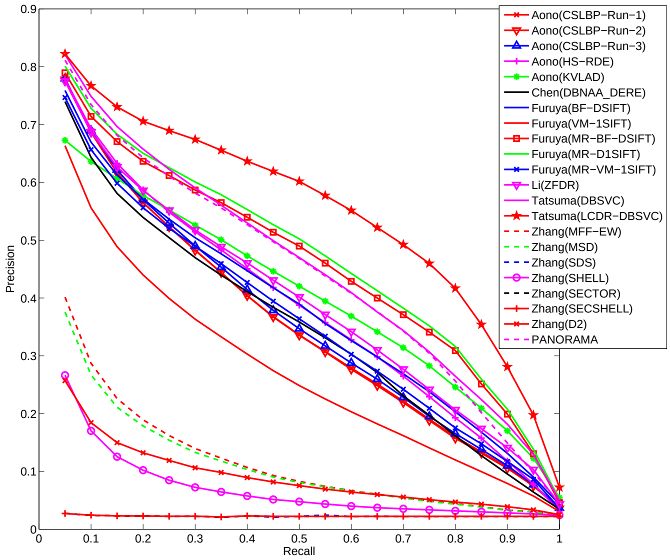


Figure 16: Precision-Recall plot performance comparison of all the twenty-two runs of the seventeen Query-by-Model retrieval algorithms from the seven groups.

As can be seen from Figure 16, if we compare approaches without employing a machine learning approach (see the R_p values in the tables), including manifold ranking, overall PANORAMA, Li's ZFDR, Aono's HSR-DF and Furuya's BF-DSIFT are comparable to Tatsuma's DBSVC approach. However, by applying a manifold ranking learning method, Tatsuma et al. achieve an apparent performance improvement, which can be validated by the resulting LCDR-DBSVC method. Compared to DBSVC, LCDR-DBSVC has a 20.6%, 17.4%, 9.0%, 4.2%, and 21.3% gain in terms of non-weighted FT, ST, E, DCG, and AP, respectively. In fact, Furuya et al.'s three "MR-" runs also have adopted a manifold ranking method to improve the retrieval performance. This indicates the advantage of employing machine learning approaches in the 3D model retrieval research field. We should mention that the above finding is consistent with the three types of metrics, including standard, proportionally, and reciprocally weighted ones.

To perform an approximate efficiency performance comparison, we asked the contributors to provide timing information in terms of average response time per query, as listed in Table 8. Obviously, ZFDR and BF-DSIFT are the most efficient ones, followed by the Shape Histogram methods (SECTOR, SHELL, SECSHELL, SDS), MSD, MFF-EW, and VM-1SIFT, whereas

Table 5: Performance metrics for the performance comparison of the twenty-two runs of the seventeen Query-by-Model retrieval algorithms from the seven groups. "R" denotes the ranking order of all the twenty-two runs, while "R_p" denotes the ranking order of all the runs that do not utilize any machine learning techniques or class information, that is, the runs of the pure shape descriptors themselves.

Contributor	**Method**	**NN**	**FT**	**ST**	**E**	**DCG**	**AP**	**R**	**R_p**
Aono	CSLBP-Run-1	0.840	0.353	0.452	0.197	0.736	0.349	12	7
	CSLBP-Run-2	0.842	0.352	0.450	0.197	0.735	0.347	13	8
	CSLBP-Run-3	0.840	0.359	0.459	0.200	0.740	0.355	11	6
	HSR-DE	0.837	0.381	0.490	0.203	0.752	0.378	8	4
	KVLAD	0.605	0.413	0.546	0.214	0.746	0.396	6	-
Chen	DBNAA_DERE	0.817	0.355	0.464	0.188	0.731	0.344	14	9
Furuya	BF-DSIFT	0.824	0.378	0.492	0.201	0.756	0.375	9	5
	VM-1SIFT	0.732	0.282	0.380	0.158	0.688	0.269	15	10
	MR-BF-DSIFT	0.845	0.455	0.567	0.229	0.784	0.453	3	-
	MR-D1SIFT	0.856	0.465	0.578	0.234	0.792	0.464	2	-
	MR-VM-1SIFT	0.812	0.368	0.467	0.194	0.737	0.357	10	-
Li	ZFDR	0.838	0.386	0.501	0.209	0.757	0.387	7	3
Tatsuma	DBSVC	**0.868**	0.438	0.563	0.234	0.790	0.446	4	**1**
	LCDR-DBSVC	0.864	**0.528**	**0.661**	**0.255**	**0.823**	**0.541**	**1**	-
Zhang	MFF-EW	0.566	0.138	0.204	0.076	0.570	0.114	16	-
	MSD	0.504	0.132	0.196	0.071	0.562	0.109	17	11
	SDS	0.486	0.074	0.114	0.041	0.511	0.023	20	14
	SHELL	0.483	0.078	0.119	0.043	0.513	0.069	19	13
	SECTOR	0.398	0.062	0.098	0.035	0.495	0.023	20	14
	SECSHELL	0.469	0.079	0.118	0.045	0.511	0.023	20	14
	D2	0.232	0.103	0.168	0.046	0.527	0.089	18	12
[53]	PANORAMA	0.859	0.436	0.560	0.225	0.783	0.437	5	2

the other methods are much slower. We also note that the best-performing method LCDR-DBSVC is slower by an order of magnitude. This also raises the issue of scalability of existing or new Query-by-Model retrieval algorithms to large corpuses, and it deserves further efforts.

Among the seven group contributors, one group (Zhang) adopts geometry-based techniques, two groups (Furuya and Tatsuma) utilize view-based techniques, while four groups (Aono, Chen, Li, and PANORAMA [53]) follow a hybrid approach. If we consider the above evaluation results as well, this demonstrates the popularity and superiority of hybrid techniques.

However, if we classify the contributing methods based on the properties of the features used, we find that two groups (Aono and Tatsuma) employ a local shape descriptor, four groups (Chen, Li, Zhang, and PANORAMA [53]) adopt a global feature, and one group (Furuya) adopts both local and global features. The two groups (Tatsuma and Furuya) that extract local features have applied the Bag-of-Words framework and K-means clustering on the local features. Within the submitted methods for Query-by-Model retrieval, this shows the popularity of global shape descriptors and the Bag-of-Words technique in dealing with local features.

6.2. Query-by-Sketch retrieval

This section presents a comparative evaluation of the twelve runs of the six methods submitted by the four groups based on **LSB**. We measure the retrieval performance using the seven metrics mentioned in Section 3.5: PR, NN, FT, ST, E, DCG, and AP.

As described in Section 3.3.4, the complete query sketch dataset is divided into "Training" and "Testing" datasets as needed by machine learning-based retrieval algorithms. To provide complete reference performance data for learning-based methods as well as non-learning based approaches (including all of the six participating methods), we evaluate the submitted results on the "Training", the "Testing", and the complete datasets. Figure 18 compares their PR performance, while Tables 9 and 10 compare the other six general and reciprocally weighted performance metrics on these three datasets.

As shown in the figure and tables, Tatsuma's SCMR-OPHOG is the best by a large margin, followed by their OPHOG and Furuya's CDMR. Nevertheless, the overall performance of the top methods from other groups are very close, while the closeness appearance of the other methods in the Precision-Recall plots is partially because of the distinct disparity between the best method and others. It appears that the other groups could catch up with OPHOG in terms of overall

Table 6: Proportionally weighted performance metrics for the performance comparison of the twenty-two runs of the seventeen Query-by-Model retrieval algorithms from the seven groups. “R” denotes the ranking order of all the twenty-two runs, while “R_p” denotes the ranking order of all the runs that do not utilize any machine learning techniques or class information, that is, the runs of the pure shape descriptors themselves.

Contributor	Method	NN	FT	ST	E	DCG	R	R_p
Aono	CSLBP-Run-1	0.880	0.379	0.502	0.145	0.800	11	7
	CSLBP-Run-2	0.881	0.375	0.495	0.145	0.798	13	9
	CSLBP-Run-3	0.878	0.381	0.505	0.146	0.802	10	6
	HSR-DE	0.882	0.405	0.539	0.148	0.812	6	3
	KVLAD	0.617	0.418	0.574	0.144	0.806	9	-
Chen	DBNAA_DERE	0.859	0.398	0.544	0.136	0.799	12	8
Furuya	BF-DSIFT	0.868	0.392	0.529	0.143	0.809	7	4
	VM-1SIFT	0.797	0.290	0.406	0.120	0.753	15	10
	MR-BF-DSIFT	0.877	0.464	0.607	0.156	0.834	5	-
	MR-D1SIFT	0.895	0.473	0.611	0.160	0.839	3	-
	MR-VM-1SIFT	0.868	0.388	0.501	0.142	0.798	13	-
Li	ZFDR	0.879	0.398	0.535	0.148	0.809	7	4
Tatsuma	DBSVC	**0.898**	0.444	0.604	0.162	0.839	3	2
	LCDR-DBSVC	0.892	**0.541**	**0.723**	**0.169**	**0.872**	**1**	-
Zhang	MFF-EW	0.582	0.159	0.252	0.056	0.654	16	-
	MSD	0.544	0.157	0.249	0.054	0.652	17	11
	SDS	0.485	0.085	0.146	0.029	0.596	21	15
	SHELL	0.486	0.091	0.153	0.031	0.600	20	14
	SECTOR	0.446	0.071	0.124	0.028	0.587	22	16
	SECSHELL	0.503	0.091	0.150	0.034	0.601	19	13
	D2	0.281	0.139	0.234	0.038	0.632	18	12
[53]	PANORAMA	0.891	0.472	0.636	0.158	0.840	2	**1**

performance (e.g., see the R_p values in Table 9, but after employing the manifold ranking-based method SCMR, Tatsuma’s group achieved much better performance. For example, compared to OPHOG, SCMR-OPHOG achieves a gain of 77.3%, 74.5%, 52.94%, 10.3%, and 116.4% in FT, ST, E, DCG, and AP, respectively. Compared to the performance obtained in the SHREC’12 and SHREC’13 sketch-based 3D model retrieval tracks [4][5], the performance of all approaches has decreased sharply due to the much more challenging data in the new **LSB** benchmark. In fact, there is an additional drop when compared to the performance achieved by the evaluated Query-by-Model retrieval algorithms in Section 6.1, which again demonstrates the challenges and semantic gaps that exist in sketch-based 3D model retrieval. It also seems worthwhile to pay more attention to scalability issues when developing sketch-based 3D retrieval algorithms, especially for large-scale retrieval applications. More details about the retrieval performance with respect to different classes for each participating method can be found on the SHREC’14 sketch track homepage [2].

For the proportionally weighted metrics, we find that the results of the evaluated methods are very close. For example, the proportionally weighted (FT, ST, E, DCG, AP) of SBR-VC (α=1) are 1.0e-05*(1.25, 1.25, 1.25, 0.00, 3.75, 1.25), while for SCMR-OPHOG, they are 1.0e-05*(2.50, 1.25, 2.50, 1.25, 5.00, 1.25). Hence, the performance of the contributed methods in retrieving classes with more variations/models is very close. If we consider the comparison and analysis results of the three types of metrics based on the Query-by-Model retrieval results in Section 6.1 as well, we regard the set of reciprocally weighted metrics as the more accurate and robust weighted version to evaluate either 2D or 3D query-based retrieval algorithms.

In addition, rather than having a consistent evaluation result as in the Query-by-Model retrieval algorithms evaluation, we find there is some discrepancy in the case of sketch-based 3D retrieval evaluation: the ranking results of the methods are somehow different when based on the reciprocally weighted metrics. For example, if we compare the ranking results in Tables 9 and 10, we find the ranking order of OPHOG and CDMR (σ_{SM}=0.05, α=0.3) to be flipped. The reciprocal version is to alleviate the bias influence due to the differences in the number of models that each class contains by proportionally weighting the performance per query by the reciprocal of the number of relevant models for the query. Therefore, it highlights the performance of classes with fewer models/variations, which is usually even lower than the average performance. This results in the even smaller performance values in Table 10. We further find that this helps differentiate the performance of the various

Table 7: Reciprocally weighted performance metrics for the performance comparison of the twenty-two runs of the seventeen Query-by-Model retrieval algorithms from the seven groups. “R” denotes the ranking order of all the twenty-two runs, while “R_p” denotes the ranking order of all the runs that do not utilize any machine learning techniques or class information, that is, the runs of the pure shape descriptors themselves.

Contributor	**Method**	**NN**	**FT**	**ST**	**E**	**DCG**	**R**	$\mathbf{R}_p$
Aono	CSLBP-Run-1	0.663	0.303	0.359	0.180	0.571	10	7
	CSLBP-Run-2	0.668	0.304	0.359	0.180	0.571	10	7
	CSLBP-Run-3	0.658	0.310	0.365	0.183	0.573	9	6
	HSR-DE	0.656	0.318	0.380	0.189	0.582	8	5
	KVLAD	0.480	0.323	0.434	0.213	0.564	12	-
Chen	DBNAA_DERE	0.626	0.281	0.339	0.169	0.552	14	9
Furuya	BF-DSIFT	0.645	0.321	0.389	0.192	0.588	6	3
	VM-1SIFT	0.547	0.235	0.290	0.142	0.510	15	10
	MR-BF-DSIFT	0.680	0.376	0.444	0.221	0.619	4	-
	MR-D1SIFT	0.689	0.383	0.455	0.227	0.627	3	-
	MR-VM-1SIFT	0.626	0.300	0.359	0.179	0.564	12	-
Li	ZFDR	0.659	0.326	0.392	0.194	0.588	6	3
Tatsuma	DBSVC	0.707	0.371	0.445	0.224	0.628	2	**1**
	LCDR-DBSVC	**0.718**	**0.428**	**0.506**	**0.255**	**0.658**	**1**	-
Zhang	MFF-EW	0.446	0.139	0.172	0.078	0.418	16	-
	MSD	0.395	0.124	0.157	0.070	0.400	17	11
	SDS	0.397	0.097	0.113	0.047	0.364	18	12
	SHELL	0.392	0.097	0.114	0.048	0.362	19	13
	SECTOR	0.300	0.063	0.080	0.035	0.327	22	16
	SECSHELL	0.370	0.095	0.111	0.047	0.357	20	14
	D2	0.160	0.069	0.102	0.046	0.338	21	15
[53]	PANORAMA	0.687	0.350	0.421	0.210	0.612	5	2

methods.

Similarly, we conducted an approximate efficiency evaluation. The average response time per query based on the “Testing” dataset using a modern computer is compared in Table 11. Obviously, BF-fGALIF is the most efficient, followed by BOF-JESC and SBR-VC ($\alpha = \frac{1}{2}$). OPHOG, SCMR-OPHOG, and SBR-VC ($\alpha = 1$) are comparable in terms of speed, while CDMR is the slowest algorithm by an order of magnitude. We believe this timing information is useful for an approximate comparison of the runtime requirements of the algorithms even though they were obtained on different computers.

Finally, we classify all participating methods with respect to the techniques employed according to the classification standards described in [10]: local/global 2D features, Bag-of-Words framework or direct feature matching, fixed/clustered views, and with/without view selection. Three groups (Furuya, Tatsuma, and Zou) utilize local features while one group (Li) employs a global feature. Two (Furuya and Zou) of the three methods based on local features apply the Bag-of-Features framework while manifold ranking is also used in two (Furuya and Tatsuma) of the three local feature-based algorithms. Only one group (Li) performs view clustering while the others employ a fixed view sampling. No group includes a view selection process in their methods.

7. Conclusions and future work

7.1. Conclusions

The LSB benchmark. This paper describes the building process of **LSB**, a large-scale 3D model retrieval benchmark supporting both 3D model and 2D sketch queries. Compared to other multimodal query-supported 3D retrieval benchmarks, its 13,680 sketches and 8,987 models of 171 classes make it the currently largest scale benchmark in terms of the number of models and sketches as well as the most comprehensive benchmark in terms of the number of object classes and variations within a class. Compared to previous sketch-based 3D retrieval benchmarks, it is not only the largest and most comprehensive but also the only currently available comprehensive 3D model benchmark. Even compared to prior generic benchmarks, it is still among the largest and most comprehensive in terms of the number of categories. In addition to the **LSB** benchmark, we also developed two versions of commonly used performance metrics, proportionally-weighted and reciprocally-weighted, by incorporating the model variations in each class based on the number of available models it contains. We regard the reciprocally-weighted version as more accurate than its original form in terms of reflecting the real performance of a 3D shape retrieval algorithm either using model or sketch queries.

Table 8: Available timing information comparison of the seventeen Query-by-Model retrieval algorithms: T is the average response time (in seconds) per query. "R" denotes the ranking order of all the seventeen runs, while "R_p" denotes the ranking order of all the runs that do not utilize any machine learning techniques or class information, that is, the runs of the pure shape descriptors themselves. For PANORAMA [53], we collected the timing information based on the publically available executable [107].

Contributor (with computer configuration)	**Method**	**Language**	T	**R**	$\mathbf{R}_p$
Chen (CPU: Intel(R) Core i3-2350M @2.3GHz (only using one thread); Memory: 6 GB; OS: Windows 2003 32-bit)	DBNAA_DERE	C#, Matlab	58.82	11	10
Furuya (CPU: Intel(R) Core i7 3930K @3.20 GHz, GPU: NVIDIA GeForce GTX 670 (the programs ran on a single thread); Memory: 64 GB; OS: Ubuntu 12.04)	BF-DSIFT	C++, CUDA	1.94	2	2
	VM-1SIFT	C++	9.60	10	9
	MR-BF-DSIFT	C++, CUDA	65.17	13	-
	MR-VM-1SIFT	C++, CUDA	65.87	14	-
	MR-D1SIFT	C++, CUDA	131.04	15	-
Li (CPU: Intel(R) Xeon(R) CPU X5675 @3.07 GHz (2 processors, 12 cores); Memory: 20 GB; OS: Windows 7 64-bit)	ZFDR	C/C++	**1.77**	**1**	**1**
Tatsuma (CPU: Intel(R) Xeon(R) CPU E5-2630 @2.30GHz (2 processors, 12 cores); Memory: 64 GB; OS: Debian Linux 7.3)	DBSVC	C++, Python	62.66	12	11
	LCDR-DBSVC	C++, Python	668.61	17	-
Zhang (CPU: Intel(R) Xeon(R) E5620 @ 2.40 GHz; Memory: 12.00 GB; OS: Windows 7 64-bit)	MFF-EW	C++, Matlab	8.05	9	-
	MSD	C++, Matlab	4.10	8	8
	SECSHELL	C++, Matlab	3.48	4	4
(CPU: Intel(R) Core(TM) i5-2450M @ 2.50 GHz; Memory: 2.45 GB; OS: Windows 7 32-bit)	SDS	C++, Matlab	3.91	6	6
	SHELL	C++, Matlab	3.65	5	5
	SECTOR	C++, Matlab	3.29	3	3
	D2	C++, Matlab	4.00	7	7
[53] (CPU: Intel(R) Xeon(R) CPU X5675 @3.07 GHz (2 processors, 12 cores); Memory: 20 GB; OS: Windows 7 64-bit)	PANORAMA	C++	370.2	16	12

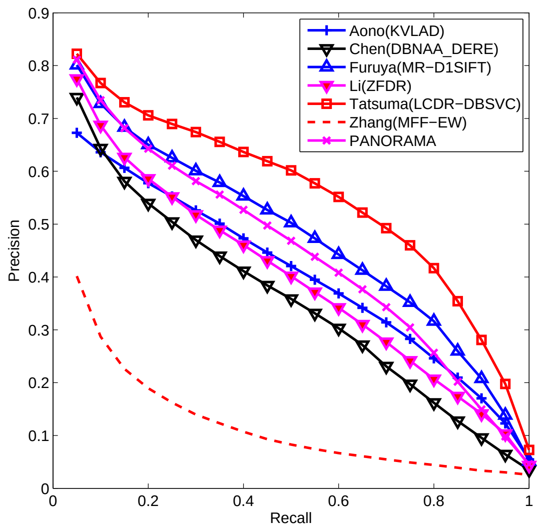


Figure 17: Precision-Recall plot performance comparison of the best runs of the Query-by-Model retrieval algorithms from each group.

We also hope that the large-scale sketch retrieval benchmark will prove useful for other researchers in our community.

Evaluation of Query-by-Model retrieval algorithms. Based on the 3D model dataset of the **LSB** benchmark, we organized the SHREC'14 large scale comprehensive 3D model retrieval track. In this paper, a comprehensive evaluation of twenty (twelve track participating and eight state-of-the-art or new) Query-by-Model retrieval algorithms has been conducted based on both non-weighted and weighted performance metrics. A comparison of approximate runtime information was also performed to provide a reference on the efficiency of the evaluated methods, which also serves as evaluation of the scalability of each method w.r.t large-scale retrieval scenarios or real applications. According to the evaluation results, among the submitted algorithms, hybrid methods, manifold ranking learning methods, and Bag-of-Words approaches are more popular and promising in the scenario of Query-by-Model retrieval, which partially illustrates a current research trend in the field of comprehensive 3D model retrieval.

Evaluation of Query-by-Sketch retrieval algorithms. Based on the complete **LSB** benchmark, we organized another SHREC'14 track on large scale sketch-based 3D retrieval. The second track is meant to foster this challenging and interesting research direction, encouraged by the success of the SHREC'12 and SHREC'13 sketch-based 3D shape retrieval tracks. Though the latest benchmark is by far the most challenging so far, we still attracted four groups who have successfully participated in the track and contributed twelve runs of six methods, which have been comparatively evaluated in this paper as well. We have noticed that the obtained retrieval performance is far from satisfactory, and the performance of existing sketch-based retrieval methods apparently drops when scaled to a significantly larger collection. Local feature and manifold ranking based approaches also dominate the evaluated methods and often achieve superior retrieval accuracy, but their performance leaves room for further improvements.

7.2. *Future work*

The **LSB** benchmark provides a common platform to evaluate 3D model retrieval approaches in the context of a large-scale retrieval scenario. It helps identify state-of-the-art methods as well as future research directions in this area. For promising future work on sketch-based 3D retrieval algorithms, please refer to [10]. Here, we mainly list several important research directions that apply to both sketch and model query based 3D retrieval algorithms.

- **Benchmark.** Since the current version of our **LSB** benchmark contains only 171 of the full set of 250 classes from Eitz et al.'s sketch dataset, there is still room for further improvement by finding models from additional sources such as the Trimble 3D Warehouse (formerly the Google 3D Warehouse) [139], to make it more complete and comprehensive in terms of class variations. In addition, making each class contain the same number of sketches/models will help eliminate any bias, which we currently cope with using the weighted metrics.

- **Increasing amounts of 3D data.** We expect that in the future, even more 3D object data will become available, due to technical advantages of 3D acquisition devices, cloud services and social media networks. In particular, the latter may include large amounts of noisy data, e.g., from handheld and mobile devices. Then, the problem to retrieve among sets of 3D data of varying quality properties will become a challenge. Compiling benchmarks that control for varying levels of quality of the 3D models will be helpful to foster research in this direction.

- **Scalability of retrieval algorithms.** Building scalable 3D retrieval systems is of utmost importance for related interactive applications. For Query-by-Sketch retrieval, an important direction for future research in this area is to develop more robust algorithms that scale to different sizes and diverse types of sketch queries and models. For Query-by-Model retrieval, though the performance is relatively speaking much better, it still requires further effort to develop an interactive system for existing or new retrieval algorithms w.r.t a large corpus by adopting additional techniques, such as parallelization (i.e., using multi-core CPUs or GPUs), as well as algorithm and code optimizations.

- **Feature coding.** Among the main parameters of 3D retrieval algorithms, the coding of features has recently come into the focus of researchers. Techniques like sparse coding, Fisher coding, VLAD coding, etc. may provide for both efficient and effective retrieval. More systematic studies are needed to assess the contribution of specific coding techniques to the overall method performance. In particular, it would be interesting to study if particular codings could be recommended for particular types of 3D features.

- **Semantics-based 3D retrieval.** As we saw, manifold learning and attribute-based semantic retrieval approaches have become more and more important to bridge the gap in the pure content-based 3D model retrieval framework to achieve satisfactory accuracy. Therefore, we recommend utilizing techniques from other related disciplines, such as machine learning, especially representation learning [140] including manifold learning and deep learning (i.e., Caffe [141]), image retrieval (i.e., ImageNet [142]), and pattern recognition (i.e., [143], to develop higher level knowledge-based 3D retrieval algorithms.

Acknowledgments

The work of Bo Li and Yijuan Lu is supported by the Texas State University Research Enhancement Program (REP), Army Research Office grant W911NF-12-1-0057, and NSF CRI 1305302 to Dr. Yijuan Lu.

Henry Johan is supported by Fraunhofer IDM@NTU, which is funded by the National Research Foundation (NRF) and managed through the multi-agency Interactive & Digital Media Programme Office (IDMPO) hosted by the Media Development Authority of Singapore (MDA).

We would like to thank Yuxiang Ye and Natacha Feola who helped us build the **LSB** benchmark.

We would like to thank Mathias Eitz, James Hays and Marc Alexa who collected the 250 classes of sketches. We would also like to thank the following authors for building the 3D benchmarks:

- Philip Shilane, Patrick Min, Michael M. Kazhdan, and Thomas A. Funkhouser who built the Princeton Shape Benchmark (**PSB**);

- Atsushi Tatsuma, Hitoshi Koyanagi, and Masaki Aono who built the Toyohashi Shape Benchmark (**TSB**);

- Dejan Vranic and colleagues who built the Konstanz 3D Model Benchmark (**CCCC**);

- Daniela Giorgi who built the Watertight Shape Benchmark (**WMB**);

- Kaleem Siddiqi, Juan Zhang, Diego Macrini, Ali Shokoufandeh, Sylvain Bouix, and Sven Dickinson who built the McGill 3D Shape Benchmark (**MSB**);

- Raoul Wessel, Ina Blümel, and Reinhard Klein from the University of Bonn and the TIB Hannover who built the Bonn Architecture Benchmark (**BAB**);

- Subramaniam Jayanti, Yagnanarayanan Kalyanaraman, Natraj Iyer, and Karthik Ramani who built the Engineering Shape Benchmark (**ESB**).

References

[1] http://www.itl.nist.gov/iad/vug/sharp/contest/2014/Generic3D/, 2014.

[2] http://www.itl.nist.gov/iad/vug/sharp/contest/2014/SBR/, 2014.

[3] A.-P. Ta, C. Wolf, G. Lavoué, A. Baskurt, 3D object detection and viewpoint selection in sketch images using local patch-based Zernike moments, in: S. D. Kollias, Y. S. Avrithis (Eds.), CBMI, IEEE Computer Society, 2009, pp. 189–94.
[4] B. Li, T. Schreck, A. Godil, M. Alexa, T. Boubekeur, B. Bustos, J. Chen, M. Eitz, T. Furuya, K. Hildebrand, S. Huang, H. Johan, A. Kuijper, R. Ohbuchi, R. Richter, J. M. Saavedra, M. Scherer, T. Yanagimachi, G.-J. Yoon, S. M. Yoon, SHREC'12 track: Sketch-based 3D shape retrieval, in: Eurographics Workshop on 3D Object Retrieval (3DOR), 2012, pp. 109–18.
[5] B. Li, Y. Lu, A. Godil, T. Schreck, M. Aono, H. Johan, J. M. Saavedra, S. Tashiro, SHREC'13 track: Large scale sketch-based 3D shape retrieval, in: Eurographics Workshop on 3D Object Retrieval (3DOR), 2013, pp. 89–96.
[6] M. Eitz, J. Hays, M. Alexa, How do humans sketch objects?, ACM Trans. Graph. 31 (2012) 44:1–44:10.
[7] P. Shilane, P. Min, M. M. Kazhdan, T. A. Funkhouser, The Princeton shape benchmark, in: SMI, 2004, pp. 167–78.
[8] B. Li, Y. Lu, C. Li, A. Godil, T. Schreck, M. Aono, M. Burtscher, H. Fu, T. Furuya, H. Johan, et al., Shrec'14 track: Extended large scale sketch-based 3D shape retrieval, in: Eurographics Workshop on 3D Object Retrieval, 2014, pp. 121–30.
[9] B. Li, Y. Lu, C. Li, A. Godil, T. Schreck, M. Aono, Q. Chen, N. Chowdhury, B. Fang, T. Furuya, et al., Shrec'14 track: Large scale comprehensive 3D shape retrieval, in: Eurographics Workshop on 3D Object Retrieval, 2014, pp. 131–40.
[10] B. Li, Y. Lu, A. Godil, T. Schreck, B. Bustos, A. Ferreira, T. Furuya, M. J. Fonseca, H. Johan, T. Matsuda, R. Ohbuchi, P. B. Pascoal, J. M. Saavedra, A comparison of methods for sketch-based 3D shape retrieval, Computer Vision and Image Understanding 119 (2014) 57–80.
[11] Z. Liu, S. Bu, K. Zhou, S.-M. Gao, J. Han, J. Wu, A survey on partial retrieval of 3D shapes, J. Comput. Sci. Technol. 28 (2013) 836–51.
[12] B. Li, A. Godil, H. Johan, Hybrid shape descriptor and meta similarity generation for non-rigid and partial 3D model retrieval, Multimedia Tools and Applications (Online First version)) (2013) 1–30.
[13] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, K. Ramani, Three-dimensional shape searching: state-of-the-art review and future trends, Computer-Aided Design 37 (2005) 509–30.
[14] B. Bustos, D. Keim, D. Saupe, T. Schreck, D. Vranić, Feature-based Similarity Search in 3D Object Databases, ACM Computing Surveys 37 (2005) 345–87.
[15] J. W. H. Tangelder, R. C. Veltkamp, A survey of content based 3D shape retrieval methods, Multimedia Tools Appl. 39 (2008) 441–71.
[16] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, Matching 3D models with shape distributions, in: Proc. of Shape Modeling and Applications, 2001, pp. 154–66.
[17] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, T. Seidl, 3D shape histograms for similarity search and classification in spatial databases, in: R. H. Güting, D. Papadias, F. H. Lochovsky (Eds.), SSD, volume 1651 of *Lecture Notes in Computer Science*, Springer, 1999, pp. 207–26.
[18] A. Frome, D. Huber, R. Kolluri, T. Bulow, J. Malik, Recognizing objects in range data using regional point descriptors, in: Proc. of the European Conference on Computer Vision (ECCV), 2004.
[19] K. S. Huang, M. M. Trivedi, 3D shape context based gesture analysis integrated with tracking using omni video array, in: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, p. 80.
[20] M. Kortgen, G.-J. Park, M. Novotni, R. Klein, 3D shape matching with 3D shape contexts, in: Proceedings of the 7th Central European Seminar on Computer Graphics(CESCG), 2003.
[21] B. Horn, Extended gaussian images, Proc. of the IEEE 72 (1984) 1671–86.
[22] M. Ben-Chen, C. Gotsman, Characterizing shape using conformal factors, in: Eurographics Workshop on 3D Object Retrieval (3DOR), 2008, pp. 1–8.
[23] M. M. Kazhdan, T. A. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3D shape descriptors, in: Symposium on Geometry Processing, 2003, pp. 156–64.
[24] X. Pan, Q. You, Z. Liu, Q. H. Chen, 3D shape retrieval by poisson histogram, Pattern Recognition Letters 32 (2011) 787–94.
[25] I. Sipiran, B. Bustos, T. Schreck, Data-aware 3D partitioning for generic shape retrieval, Computers & Graphics 37 (2013) 460–72.
[26] K. sheng Zou, W.-H. Ip, C.-H. Wu, Z. Chen, K.-L. Yung, C.-Y. Chan, A novel 3D model retrieval approach using combined shape distribution, Multimedia Tools Appl. 69 (2014) 799–818.
[27] H. Sundar, D. Silver, N. Gagvani, S. J. Dickinson, Skeleton based shape matching and retrieval, in: Shape Modeling International, 2003, pp. 130–9.
[28] C. Li, A. B. Hamza, Symmetry discovery and retrieval of nonrigid 3D shapes using geodesic skeleton paths, Multimedia Tools and Applications (2013) 1–21.
[29] M. Hilaga, Y. Shinagawa, T. Komura, T. L. Kunii, Topology matching for fully automatic similarity estimation of 3D shapes, in: SIGGRAPH 2001, 2001, pp. 203–12.
[30] V. Barra, S. Biasotti, 3D shape retrieval using kernels on extended Reeb graphs, Pattern Recognition 46 (2013) 2985–99.
[31] H. Edelsbrunner, D. Letscher, A. Zomorodian, Topological persistence and simplification, Discrete and Computational Geometry 28 (2002) 511–33.
[32] P. Frosini, Measuring shapes by size functions, in: Intelligent Robots and Computer Vision X: Algorithms and Techniques, International Society for Optics and Photonics, 1992, pp. 122–33.
[33] S. Biasotti, A. Cerri, D. Giorgi, M. Spagnuolo, PHOG: Photometric and geometric functions for textured shape retrieval, in: Computer Graphics Forum, volume 32, Wiley Online Library, 2013, pp. 13–22.
[34] C. Li, M. Ovsjanikov, F. Chazal, Persistence-based structural recognition, in: Computer Vision and Pattern Recognition, IEEE, 2014.
[35] G. Carlsson, Topology and data, Bulletin of the American Mathematical Society 46 (2009) 255–308.
[36] Q. Liu, A survey of recent view-based 3D model retrieval methods, CoRR abs/1208.3670 (2012).
[37] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, M. Ouhyoung, On visual similarity based 3D model retrieval, Computer Graphics Forum 22 (2003) 223–32.
[38] M. Chaouch, A. Verroust-Blondet, A new descriptor for 2D depth image indexing and 3D model retrieval, in: ICIP (6), 2007, pp. 373–6.
[39] R. Ohbuchi, K. Osada, T. Furuya, T. Banno, Salient local visual features for shape-based 3D model retrieval, in: Shape Modeling International, 2008, pp. 93–102.
[40] P. Daras, A. Axenopoulos, A 3D shape retrieval framework supporting multimodal queries, International Journal of Computer Vision 89 (2010) 229–47.
[41] B. Li, H. Johan, View context based 2D sketch-3D model alignment, in: WACV, IEEE Computer Society, 2011, pp. 45–50.
[42] Z. Lian, A. Godil, X. Sun, Visual similarity based 3D shape retrieval using Bag-of-Features, in: Shape Modeling International, 2010, pp. 25–36.
[43] Y. Gao, Y. Yang, Q. Dai, N. Zhang, 3D object retrieval with bag-of-region-words, in: ACM Multimedia, 2010, pp. 955–8.
[44] A. Axenopoulos, P. G. Litos, 3D model retrieval using accurate pose estimation and view-based similarity, in: Proceedings of the 1st ACM International Conference on Multimedia Retrieval, 2011, pp. 1–8.
[45] K. Ding, Y. Liu, Sphere image for 3-D model retrieval, IEEE Transactions on Multimedia 16 (2014) 1369–76.
[46] X. Bonaventura, J. Guo, W. Meng, M. Feixas, X. Zhang, M. Sbert, 3D shape retrieval using viewpoint information-theoretic measures, Computer Animation and Virtual Worlds (2013) 1:1–1:13.
[47] L. Li, S. Zhang, X. Bai, L. Shao, Retrieving 3D model using compound-eye visual representation, in: Computer-Aided Design and Computer Graphics (CAD/Graphics), 2013 International Conference on, 2013, pp. 172–9.
[48] K. Sfikas, T. Theoharis, I. Pratikakis, 3D object retrieval via range image queries in a bag-of-visual-words context, The Visual Computer 29 (2013) 1351–61.
[49] P. F. Alcantarilla, A. Bartoli, A. J. Davison, KAZE Features, in: ECCV (6), 2012, pp. 214–27.
[50] H. Jegou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: CVPR, 2010, pp. 3304–11.
[51] D. V. Vranic, DESIRE: a composite 3D-shape descriptor, in: ICME, 2005, pp. 962–5.
[52] P. Papadakis, I. Pratikakis, T. Theoharis, G. Passalis, S. J. Perantonis, 3D object retrieval using an efficient and compact hybrid shape descriptor, in: Eurographics Workshop on 3D Object Retrieval (3DOR), 2008, pp.

9–16.

[53] P. Papadakis, I. Pratikakis, T. Theoharis, S. Perantonis, PANORAMA: A 3D shape descriptor based on panoramic views for unsupervised 3D object retrieval, International Journal of Computer Vision 89 (2010) 177–92.

[54] B. Li, H. Johan, 3D model retrieval using hybrid features and class information, Multimedia Tools Appl. 62 (2013) 821–46.

[55] P. Li, H. Ma, A. Ming, Combining topological and view-based features for 3D model retrieval, Multimedia Tools Appl. 65 (2013) 335–61.

[56] Z. Liu, S. Bu, J. Han, J. Wu, Sparse patch coding for 3D model retrieval, in: C. Gurrin, F. Hopfgartner, W. Hürst, H. D. Johansen, H. Lee, N. E. O'Connor (Eds.), MMM (2), volume 8326 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 116–27.

[57] H. Tabia, H. Laga, D. Picard, P. H. Gosselin, Covariance descriptors for 3d shape matching and retrieval, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014, IEEE, 2014, pp. 4185–92. URL: `http://dx.doi.org/10.1109/CVPR.2014.533`. doi:`10.1109/CVPR.2014.533`.

[58] R. Osada, T. A. Funkhouser, B. Chazelle, D. P. Dobkin, Shape distributions, ACM Trans. Graph. 21 (2002) 807–32.

[59] D. Vranic, 3D Model Retrieval, PhD thesis, University of Leipzig, 2004.

[60] C. Li, A. B. Hamza, Spatially aggregating spectral descriptors for nonrigid 3D shape retrieval: a comparative survey, Multimedia Syst. 20 (2014) 253–81.

[61] E. Boyer, A. M. Bronstein, M. M. Bronstein, B. Bustos, T. Darom, R. Horaud, I. Hotz, Y. Keller, J. Keustermans, A. Kovnatsky, R. Litman, J. Reininghaus, I. Sipiran, D. Smeets, P. Suetens, D. Vandermeulen, A. Zaharescu, V. Zobel, SHREC 2011: robust feature detection and description benchmark, Eurographics Workshop on Shape Retrieval 2 (2011) 79–86.

[62] R. R. Coifman, S. Lafon, Diffusion maps, Applied and computational harmonic analysis 21 (2006) 5–30.

[63] B. Lévy, Laplace-Beltrami eigenfunctions towards an algorithm that "understands" geometry, in: SMI, IEEE Computer Society, 2006, pp. 13:1–8.

[64] M. Reuter, F.-E. Wolter, N. Peinecke, Laplace–beltrami spectra as 'shape-dna' of surfaces and solids, Computer-Aided Design 38 (2006) 342–66.

[65] J. Sun, M. Ovsjanikov, L. Guibas, A concise and provably informative multi-scale signature based on heat diffusion, in: Computer Graphics Forum, volume 28, 2009, pp. 1383–92.

[66] K. Gebal, J. A. Bærentzen, H. Aanæs, R. Larsen, Shape analysis using the auto diffusion function, in: Computer Graphics Forum, volume 28, Wiley Online Library, 2009, pp. 1405–13.

[67] M. Aubry, U. Schlickewei, D. Cremers, The wave kernel signature: A quantum mechanical approach to shape analysis, in: Proc. of ICCV Workshop on Dynamic Shape Capture and Analysis, 2011, pp. 1626–33.

[68] C. Li, A. B. Hamza, A multiresolution descriptor for deformable 3D shape retrieval, The Visual Computer 29 (2013) 513–24.

[69] A. M. Bronstein, M. M. Bronstein, L. J. Guibas, M. Ovsjanikov, Shape google: Geometric words and expressions for invariant shape retrieval, ACM Transactions on Graphics (TOG) 30 (2011) 1.

[70] C. Li, A. B. Hamza, Intrinsic spatial pyramid matching for deformable 3D shape retrieval, International Journal of Multimedia Information Retrieval 2 (2013) 261–71.

[71] B. Gao, H. Zheng, S. Zhang, An overview of semantics processing in content-based 3D model retrieval, in: Artificial Intelligence and Computational Intelligence, 2009. AICI '09. International Conference on, volume 2, 2009, pp. 54–9.

[72] G. Leifman, R. Meir, A. Tal, Semantic-oriented 3D shape retrieval using relevance feedback, The Visual Computer 21 (2005) 865–75.

[73] S. Hou, K. Lou, K. Ramani, SVM-based semantic clustering and retrieval of a 3D model database, Computer-Aided Design and Applications 2 (2005) 155–64.

[74] D. Xu, H. Li, 3D shape retrieval integrated with classification information, in: Image and Graphics, 2007. ICIG 2007. Fourth International Conference on, 2007, pp. 774 –9.

[75] H. Laga, M. Nakajima, Supervised learning of salient 2D views of 3D models, Journal of Society for Art and Sciences 7 (2008) 124–31.

[76] H. Laga, M. Nakajima, Supervised learning of similarity measures for content-based 3D model retrieval, in: LKR, 2008, pp. 210–25.

[77] H. Laga, Semantics-driven approach for automatic selection of best views of 3D shapes, in: 3DOR, 2010, pp. 15–22.

[78] R. Wessel, R. Klein, Learning the Compositional Structure of Man-Made Objects for 3D Shape Retrieval, in: 3DOR, 2010, pp. 39–46.

[79] B. Bustos, D. A. Keim, D. Saupe, T. Schreck, D. V. Vranic, Using entropy impurity for improved 3D object similarity search, in: ICME, 2004, pp. 1303–6.

[80] R. Ohbuchi, A. Yamamoto, J. Kobayashi, Learning semantic categories for 3D model retrieval, in: Multimedia Information Retrieval, 2007, pp. 31–40.

[81] A. Yamamoto, M. Tezuka, T. Shimizu, R. Ohbuchi, SHREC'08 entry: Semi-supervised learning for semantic 3D model retrieval, in: Shape Modeling International, 2008, pp. 241–3.

[82] H. Laga, M. Nakajima, A boosting approach to content-based 3D model retrieval, in: GRAPHITE, 2007, pp. 227–34.

[83] S. Biasotti, D. Giorgi, S. Marini, M. Spagnuolo, B. Falcidieno, 3D classification via structural prototypes, in: B. Falcidieno, M. Spagnuolo, Y. S. Avrithis, I. Kompatsiaris, P. Buitelaar (Eds.), SAMT, volume 4816 of *Lecture Notes in Computer Science*, Springer, 2007, pp. 140–3.

[84] C. Goldfeder, P. K. Allen, Autotagging to improve text search for 3D models, in: JCDL, 2008, pp. 355–8.

[85] A. Tatsuma, M. Aono, Multi-fourier spectra descriptor and augmentation with spectral clustering for 3D shape retrieval, The Visual Computer 25 (2008) 785–804.

[86] R. Ohbuchi, T. Shimizu, Ranking on semantic manifold for shape-based 3D model retrieval, in: Multimedia Information Retrieval, 2008, pp. 411–8.

[87] T. yang Lv, G. Liu, S. Huang, Z. Wang, Semantic 3D model retrieval based on semantic tree and shape feature, in: Y. Chen, H. Deng, D. Zhang, Y. Xiao (Eds.), FSKD (5), IEEE Computer Society, 2009, pp. 452–7.

[88] R. Ohbuchi, M. Tezuka, T. Furuya, T. Oyobe, Squeezing bag-of-features for scalable and semantic 3D model retrieval, in: G. Quénot (Ed.), CBMI, IEEE, 2010, pp. 1–6.

[89] L. Chen, B. Leng, Z. Xiong, C. Chen, A new framework for composing vectorial semantic labels in 3D model retrieval, in: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, volume 8205 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 2011.

[90] T. Lv, S. Huang, P. Wu, D. Lang, Clustering analysis and semantics annotation of 3D models based on users' implicit feedbacks, in: J. Wang, H. Xiong, Y. Ishikawa, J. Xu, J. Zhou (Eds.), WAIM, volume 7923 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 757–68.

[91] H. Laga, M. Mortara, M. Spagnuolo, Geometry and context for semantic correspondences and functionality recognition in man-made 3D shapes, ACM Trans. Graph. 32 (2013) 150.

[92] J.-Y. Wang, Y. Sun, X. Gao, Sparse structure regularized ranking, Multimedia Tools and Applications (2014) 1–20.

[93] M. A. Kassimi, O. E. Beqqali, 3D model retrieval based on semantic and shape indexes, CoRR abs/1111.6387 (2011).

[94] H. Zhang, Z.-J. Zha, Y. Yang, S. Yan, Y. Gao, T.-S. Chua, Attribute-augmented semantic hierarchy: towards bridging semantic gap and intention gap in image retrieval, in: ACM Multimedia, 2013, pp. 33–42.

[95] B. Gong, J. Liu, X. Wang, X. Tang, Learning semantic signatures for 3D object retrieval, IEEE Transactions on Multimedia 15 (2013) 369–77.

[96] B. Li, A. Godil, M. Aono, X. Bai, T. Furuya, L. Li, R. J. López-Sastre, H. Johan, R. Ohbuchi, C. Redondo-Cabrera, A. Tatsuma, T. Yanagimachi, S. Zhang, SHREC'12 track: Generic 3D shape retrieval, in: M. Spagnuolo, M. M. Bronstein, A. M. Bronstein, A. Ferreira (Eds.), 3DOR, Eurographics Association, 2012, pp. 119–26.

[97] A. Tatsuma, H. Koyanagi, M. Aono, A large-scale shape benchmark for 3D object retrieval: Toyohashi Shape Benchmark, in: Proc. of 2012 Asia Pacific Signal and Information Processing Association (APSIPA2012), 2012.

[98] R. C. Veltkamp, F. B. ter Haar, SHREC 2007 3D Retrieval Contest, Technical Report UU-CS-2007-015, Department of Information and Computing Sciences, Utrecht University, 2007.

[99] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, S. J. Dickinson, Retrieving articulated 3-D models using medial surfaces, Mach. Vis. Appl. 19 (2008) 261–75.

[100] R. Wessel, I. Blümel, R. Klein, A 3D shape benchmark for retrieval

and automatic classification of architectural data, in: M. Spagnuolo, I. Pratikakis, R. C. Veltkamp, T. Theoharis (Eds.), 3DOR, Eurographics Association, 2009, pp. 53–6.

[101] S. Jayanti, Y. Kalyanaraman, N. Iyer, K. Ramani, Developing an engineering shape benchmark for CAD models, Computer-Aided Design 38 (2006) 939–53.

[102] R. Fang, A. Godil, X. Li, A. Wagan, A new shape benchmark for 3D object retrieval, in: ISVC (1), 2008, pp. 381–92.

[103] `http://shapes.aimatshape.net/`, 2014.

[104] `http://www.aimatshape.net/event/SHREC/`, 2014.

[105] A. Bronstein, M. Bronstein, R. Kimmel, Numerical Geometry of Non-Rigid Shapes, 1 ed., Springer Publishing Company, Incorporated, 2008. URL: `http://tosca.cs.technion.ac.il/book/resources_data.html`.

[106] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, C. Schmid, Evaluation of GIST descriptors for web-scale image search, in: S. Marchand-Maillet, Y. Kompatsiaris (Eds.), Proceedings of the 8th ACM International Conference on Image and Video Retrieval, CIVR 2009, Santorini Island, Greece, July 8-10, 2009, ACM, 2009.

[107] `https://sites.google.com/site/pgpapadakis/software/`, 2014.

[108] M. Heikkilä, M. Pietikäinen, C. Schmid, Description of interest regions with center-symmetric local binary patterns, in: P. K. Kalra, S. Peleg (Eds.), ICVGIP, volume 4338 of *Lecture Notes in Computer Science*, Springer, 2006, pp. 58–69.

[109] C. H. Chen, L. F. Pau, P. S. P. Wang, Handbook of Pattern Recognition and Computer Vision (2nd Edition), World Scientific Publishing Co., Inc., 1998.

[110] M. Aono, H. Koyanagi, A. Tatsuma, 3D shape retrieval focused on holes and surface roughness, in: Proc. of 2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013, pp. 1–8.

[111] Q. Chen, B. Fang, Y.-M. Yu, Y. Tang, 3D CAD model retrieval based on the combination of features, Multimedia Tools and Applications (2014) (Online first version).

[112] R. Ohbuchi, T. Furuya, Distance metric learning and feature combination for shape-based 3D model retrieval, in: Proceedings of the ACM workshop on 3D object retrieval, 3DOR '10, 2010, pp. 63–8.

[113] T. Furuya, R. Ohbuchi, Dense sampling and fast encoding for 3D model retrieval using bag-of-visual features, in: Proceedings of the ACM International Conference on Image and Video Retrieval, CIVR '09, 2009, pp. 26:1–8.

[114] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, in: S. Thrun, L. K. Saul, B. Schölkopf (Eds.), NIPS, MIT Press, 2003.

[115] X. Zhou, K. Yu, T. Zhang, T. Huang, Image classification using super-vector coding of local image descriptors, in: K. Daniilidis, P. Maragos, N. Paragios (Eds.), Computer Vision - ECCV 2010, volume 6315 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2010, pp. 141–54.

[116] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (SURF), Computer Vision Image Understanding 110 (2008) 346–59.

[117] X. Yang, S. Köknar-Tezel, L. J. Latecki, Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval, in: Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 357–64.

[118] Busto, Benjamin, Keim, D. A, Saupe, Dietmar, Schreck, Tobias, Vranic, Dejan, Automatic selection and combination of descriptors for effective 3D similarity search, in: Proc. IEEE Sixth International Symposium on Multimedia Software Engineering, IEEE, 2004, pp. 514–21.

[119] J. Chen, B. He, X. Wang, Hpal information entropy based combination methods for 3D model retrieval, in: Journal of System Simulation, 2012, pp. 1777–9.

[120] T. Furuya, R. Ohbuchi, Ranking on cross-domain manifold for sketch-based 3D model retrieval, in: Cyberworlds 2013, 2013, pp. 274–81.

[121] B. Li, Y. Lu, H. Johan, Sketch-based 3D model retrieval by viewpoint entropy-based adaptive view clustering, in: Eurographics Workshop on 3D Object Retrieval (3DOR), 2013, pp. 49–56.

[122] A. Bosch, A. Zisserman, X. Munoz, Representing shape with a spatial pyramid kernel, in: Proceedings of the 6th ACM International Conference on Image and Video Retrieval, CIVR '07, 2007, pp. 401–8.

[123] D. Zhou, J. Weston, A. Gretton, O. Bousquet, B. Schölkopf, Ranking on data manifolds, in: S. Thrun, L. Saul, B. Schölkopf (Eds.), Advances in Neural Information Processing Systems 16, MIT Press, Cambridge, MA, 2004.

[124] M. Maire, P. Arbelaez, C. Fowlkes, J. Malik, Using contours to detect and localize junctions in natural images, in: CVPR, IEEE Computer Society, 2008.

[125] A. C. Berg, T. L. Berg, J. Malik, Shape matching and object recognition using low distortion correspondences, in: Proc. CVPR, 2005, pp. 26–33.

[126] F. Perronnin, Y. Liu, J. Sánchez, H. Poirier, Large-scale image retrieval with compressed fisher vectors, in: CVPR, 2010, pp. 3384–91.

[127] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2004) 91–110.

[128] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Machine Learning 63 (2006) 3–42.

[129] A. Khotanzad, Y. Hong, Invariant image recognition by Zernike moments, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (1990) 489–97.

[130] D. Zhang, G. Luo, A comparative study on shape retrieval using Fourier Descriptors with different shape signatures, in: Proc. of International Conference on Intelligent Multimedia and Distance Education (ICIMADE01), 2001, pp. 1–9.

[131] A. M. Bronstein, M. M. Bronstein, L. J. Guibas, M. Ovsjanikov, Shape google: Geometric words and expressions for invariant shape retrieval, ACM Transactions on Graphics 30 (2011) 1–20.

[132] J. Sánchez, F. Perronnin, T. de Campos, Modeling the spatial layout of images beyond spatial pyramids, Pattern Recognition Letters 33 (2012) 2216–23.

[133] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, M. Alexa, Sketch-based shape retrieval, ACM Trans. Graph. 31 (2012) 31:1–31:10.

[134] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, A. Santella, Suggestive contours for conveying shape, ACM Trans. Graph. 22 (2003) 848–55.

[135] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 509–22.

[136] C. Loop, Smooth Subdivision Surfaces Based on Triangles, Master's thesis, University of Utah, 1987.

[137] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[138] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Trans. PAMI 27 (2005) 1615–30.

[139] `https://3dwarehouse.sketchup.com/`, 2014.

[140] Y. Bengio, A. C. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1798–828.

[141] Y. Jia, Caffe: An open source convolutional architecture for fast feature embedding, `http://caffe.berkeleyvision.org/`, 2013.

[142] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, in: CVPR09, 2009.

[143] B. Li, Y. Lu, R. Fares, Semantic sketch-based 3D model retrieval, in: ICME Workshops, IEEE, 2013, pp. 1–4.

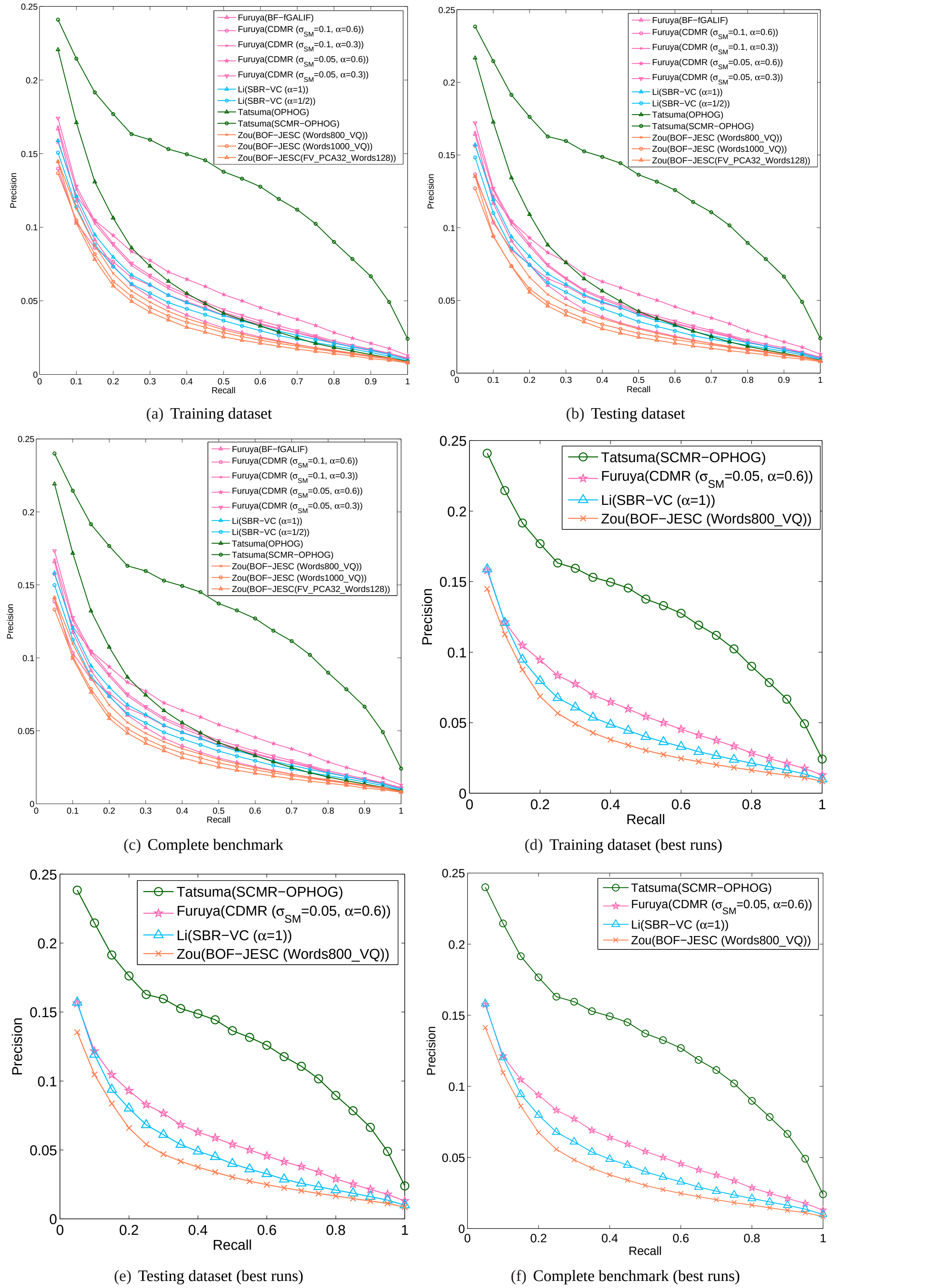


Figure 18: Precision-Recall plot performance comparisons on different datasets of our **LSB** benchmark for the twelve runs of six Query-by-Sketch retrieval methods from the four participating groups. Please note that the range of the precision axis is [0, 0.25].

Table 9: Performance metrics comparison on different datasets of our **LSB** benchmark for the twelve runs of six Query-by-Sketch retrieval methods from the four participating groups. "R" denotes the ranking order of all the twelve runs, while "R_p" denotes the ranking order of all the runs that do not utilize any machine learning techniques, that is, the runs of the pure shape descriptors themselves.

Contributor	**Method**	**NN**	**FT**	**ST**	**E**	**DCG**	**AP**	**R**	**R_p**
Training dataset									
Furuya	BF-fGALIF	0.113	0.050	0.079	0.036	0.321	0.045	9	4
	CDMR (σ_{SM}=0.1, α=0.6)	0.069	0.046	0.074	0.031	0.308	0.048	7	-
	CDMR (σ_{SM}=0.1, α=0.3)	0.104	0.055	0.087	0.039	0.324	0.053	5	-
	CDMR (σ_{SM}=0.05, α=0.6)	0.085	0.058	0.094	0.040	0.325	0.060	2	-
	CDMR (σ_{SM}=0.05, α=0.3)	0.109	0.057	0.090	0.041	0.329	0.055	4	-
Li	SBR-VC (α=1)	0.097	0.050	0.081	0.038	0.320	0.050	6	2
	SBR-VC ($\alpha = \frac{1}{2}$)	0.094	0.047	0.077	0.035	0.316	0.046	8	3
Tatsuma	OPHOG	**0.158**	0.066	0.097	0.051	0.340	0.060	2	**1**
	SCMR-OPHOG	**0.158**	**0.118**	**0.172**	**0.078**	**0.375**	**0.132**	**1**	-
Zou	BOF-JESC (Words800_VQ)	0.107	0.043	0.068	0.031	0.312	0.042	10	5
	BOF-JESC (Words1000_VQ)	0.101	0.040	0.064	0.028	0.307	0.039	11	6
	BOF-JESC (FV_PCA32_Words128)	0.099	0.040	0.062	0.027	0.304	0.038	12	7
Testing dataset									
Furuya	BF-fGALIF	0.115	0.051	0.078	0.036	0.321	0.044	9	4
	CDMR (σ_{SM}=0.1, α=0.6)	0.065	0.046	0.075	0.031	0.308	0.047	7	-
	CDMR (σ_{SM}=0.1, α=0.3)	0.100	0.056	0.087	0.039	0.325	0.052	5	-
	CDMR (σ_{SM}=0.05, α=0.6)	0.081	0.058	0.094	0.040	0.326	0.060	3	-
	CDMR (σ_{SM}=0.05, α=0.3)	0.109	0.057	0.089	0.041	0.328	0.054	4	-
Li	SBR-VC (α=1)	0.095	0.050	0.081	0.037	0.319	0.050	6	2
	SBR-VC ($\alpha = \frac{1}{2}$)	0.083	0.047	0.075	0.035	0.315	0.046	8	3
Tatsuma	OPHOG	**0.160**	0.067	0.099	0.052	0.341	0.061	2	**1**
	SCMR-OPHOG	**0.160**	**0.115**	**0.170**	**0.079**	**0.376**	**0.131**	**1**	-
Zou	BOF-JESC (Words800_VQ)	0.086	0.043	0.068	0.030	0.310	0.041	10	5
	BOF-JESC (Words1000_VQ)	0.082	0.038	0.062	0.027	0.304	0.037	11	6
	BOF-JESC (FV_PCA32_Words128)	0.089	0.038	0.060	0.026	0.302	0.036	12	7
Complete benchmark									
Furuya	BF-fGALIF	0.114	0.050	0.079	0.036	0.321	0.045	9	4
	CDMR (σ_{SM}=0.1, α=0.6)	0.068	0.046	0.074	0.031	0.308	0.048	7	-
	CDMR (σ_{SM}=0.1, α=0.3)	0.102	0.055	0.087	0.039	0.324	0.053	5	-
	CDMR (σ_{SM}=0.05, α=0.6)	0.084	0.058	0.094	0.040	0.325	0.060	3	-
	CDMR (σ_{SM}=0.05, α=0.3)	0.109	0.057	0.090	0.041	0.329	0.054	4	-
Li	SBR-VC (α=1)	0.096	0.050	0.081	0.038	0.319	0.050	6	2
	SBR-VC ($\alpha = \frac{1}{2}$)	0.090	0.047	0.077	0.035	0.316	0.046	8	3
Tatsuma	OPHOG	**0.159**	0.066	0.098	0.051	0.341	0.061	2	**1**
	SCMR-OPHOG	0.158	**0.117**	**0.171**	**0.078**	**0.376**	**0.132**	**1**	-
Zou	BOF-JESC (Words800_VQ)	0.099	0.043	0.068	0.031	0.311	0.042	10	5
	BOF-JESC (Words1000_VQ)	0.094	0.039	0.063	0.028	0.306	0.039	11	6
	BOF-JESC (FV_PCA32_Words128)	0.095	0.039	0.061	0.027	0.303	0.037	12	7

Table 10: Reciprocally weighted performance metrics comparison on different datasets of the **LSB** benchmark for the twelve runs of six Query-by-Sketch retrieval methods from the four participating groups. "R" denotes the ranking order of all the twelve runs, while "R_p" denotes the ranking order of all the runs that do not utilize any machine learning techniques, that is, the runs of the pure shape descriptors themselves.

Contributor	Method	NN	FT	ST	E	DCG	AP	R	R_p
Training dataset					1.0e-05∗				
Furuya	BF-fGALIF	0.435	0.274	0.414	0.175	2.038	0.344	4	2
	CDMR (σ_{SM}=0.1, α=0.6)	0.186	0.140	0.222	0.126	1.693	0.159	11	-
	CDMR (σ_{SM}=0.1, α=0.3)	0.389	0.259	0.382	0.183	1.951	0.304	6	-
	CDMR (σ_{SM}=0.05, α=0.6)	0.336	0.273	0.408	0.187	1.930	0.316	5	-
	CDMR (σ_{SM}=0.05, α=0.3)	0.442	0.301	0.454	0.201	2.055	0.369	2	-
Li	SBR-VC (α=1)	0.259	0.145	0.267	0.164	1.868	0.198	8	4
	SBR-VC ($\alpha = \frac{1}{2}$)	0.259	0.158	0.277	0.155	1.872	0.195	9	5
Tatsuma	OPHOG	0.528	0.295	0.458	0.233	2.089	0.348	3	**1**
	SCMR-OPHOG	**0.526**	**0.399**	**0.615**	**0.318**	**2.173**	0.490	**1**	-
Zou	BOF-JESC (Words800_VQ)	0.334	0.149	0.260	0.137	1.884	0.221	7	3
	BOF-JESC (Words1000_VQ)	0.312	0.139	0.203	0.124	1.824	0.189	10	6
	BOF-JESC (FV_PCA32_Words128)	0.327	0.146	0.199	0.103	1.746	0.157	12	7
Testing dataset					1.0e-05∗				
Furuya	BF-fGALIF	0.802	0.520	0.735	0.289	3.408	0.596	4	2
	CDMR (σ_{SM}=0.1, α=0.6)	0.299	0.237	0.406	0.222	2.861	0.281	11	-
	CDMR (σ_{SM}=0.1, α=0.3)	0.679	0.467	0.719	0.308	3.323	0.553	6	-
	CDMR (σ_{SM}=0.05, α=0.6)	0.576	0.467	0.782	0.318	3.305	0.583	5	-
	CDMR (σ_{SM}=0.05, α=0.3)	0.789	0.526	0.773	0.330	3.430	0.626	2	-
Li	SBR-VC (α=1)	0.449	0.264	0.425	0.264	3.051	0.291	9	5
	SBR-VC ($\alpha = \frac{1}{2}$)	0.414	0.265	0.405	0.259	3.088	0.311	8	4
Tatsuma	OPHOG	0.917	0.509	0.777	0.396	3.539	0.615	3	**1**
	SCMR-OPHOG	**0.993**	**0.743**	**1.035**	**0.541**	**3.676**	**0.886**	**1**	-
Zou	BOF-JESC (Words800_VQ)	0.462	0.271	0.467	0.236	3.149	0.370	7	3
	BOF-JESC (Words1000_VQ)	0.403	0.208	0.356	0.194	3.020	0.286	10	6
	BOF-JESC (FV_PCA32_Words128)	0.455	0.225	0.336	0.170	2.910	0.254	12	7
Complete benchmark					1.0e-05∗				
Furuya	BF-fGALIF	0.283	0.180	0.265	0.109	1.275	0.218	4	2
	CDMR (σ_{SM}=0.1, α=0.6)	0.078	0.065	0.109	0.058	0.760	0.073	12	-
	CDMR (σ_{SM}=0.1, α=0.3)	0.247	0.167	0.250	0.115	1.229	0.196	6	-
	CDMR (σ_{SM}=0.05, α=0.6)	0.212	0.172	0.269	0.118	1.219	0.206	5	-
	CDMR (σ_{SM}=0.05, α=0.3)	0.284	0.192	0.286	0.125	1.285	0.232	2	-
Li	SBR-VC (α=1)	0.164	0.094	0.164	0.101	1.159	0.118	9	5
	SBR-VC ($\alpha = \frac{1}{2}$)	0.160	0.099	0.161	0.097	1.166	0.120	8	4
Tatsuma	OPHOG	0.335	0.187	0.288	0.147	1.314	0.223	3	**1**
	SCMR-OPHOG	**0.345**	**0.260**	**0.386**	**0.200**	**1.366**	**0.316**	**1**	-
Zou	BOF-JESC (Words800_VQ)	0.196	0.097	0.167	0.087	1.179	0.138	7	3
	BOF-JESC (Words1000_VQ)	0.179	0.084	0.129	0.076	1.137	0.114	10	6
	BOF-JESC (FV_PCA32_Words128)	0.192	0.089	0.125	0.064	1.091	0.097	11	7

Table 11: Timing information comparison of the six Query-by-Sketch retrieval algorithms: T is the average response time (in seconds) per query based on the "Testing" dataset. "R" denotes the ranking order of all the twelve runs, while "R_p" denotes the ranking order of all the runs that do not utilize any machine learning techniques, that is, the runs of the pure shape descriptors themselves.

Contributor (with computer configuration)	**Method**	**Language**	T	**R**	R_p
Furuya (CPU: Intel(R) Core i7 3930K @3.20 GHz, GPU: NVIDIA GeForce GTX 670 (on a single thread); Memory: 64 GB; OS: Ubuntu 12.04)	BF-fGALIF	C++	**1.82**	**1**	**1**
	CDMR	C++, CUDA	126.81	7	-
Li (CPU: Intel(R) Xeon(R) CPU X5675 @3.07 GHz (2 processors, 12 cores); Memory: 20 GB; OS: Windows 7 64-bit)	SBR-VC (α=1)	C/C++	27.49	6	5
	SBR-VC ($\alpha = \frac{1}{2}$)	C/C++	15.16	3	3
Tatsuma (CPU: Intel(R) Xeon(R) CPU E5-2630 @2.30GHz (2 processors, 12 cores); Memory: 64 GB; OS: Debian Linux 7.3)	OPHOG	C++, Python	23.85	4	4
	SCMR-OPHOG	C++, Python	25.67	5	-
Zou (CPU: Intel(R) Xeon(R) W3550@3.07GHz (the programs ran on a single thread); Memory: 24 GB; OS: Windows 7 64-bit)	BOF-JESC	Matlab	6.10	2	2