Erschienen in: Computer Vision and Image Understanding ; 119 (2014). - S. 57-80

A comparison of methods for sketch-based 3D shape retrieval ☆

Bo Li [a,b], Yijuan Lu [a,*], Afzal Godil [b], Tobias Schreck [c], Benjamin Bustos [d], Alfredo Ferreira [e], Takahiko Furuya [f], Manuel J. Fonseca [e], Henry Johan [g], Takahiro Matsuda [f], Ryutarou Ohbuchi [f], Pedro B. Pascoal [e], Jose M. Saavedra [d,h]

[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, Germany
[d] Department of Computer Science, University of Chile, Chile
[e] Instituto Superior Técnico/ Technical University of Lisbon/ INESC-ID, Portugal
[f] Computer Science and Engineering Department, University of Yamanashi, Yamanashi, Japan
[g] Visual Computing, Fraunhofer IDM@NTU, Singapore
[h] Computer Vision Research Group, ORAND S.A., Santiago, Chile

ARTICLE INFO

Keywords:
Sketch-based 3D model retrieval
Evaluation
SHREC contest
Large-scale
Benchmark

ABSTRACT

Sketch based 3D shape retrieval has become an important research topic in content based 3D object retrieval. To foster this research area, two Shape Retrieval Contest (SHREC) tracks on this topic have been organized by us in 2012 and 2013 based on a small scale and large scale benchmarks, respectively. Six and five (nine in total) distinct sketch based 3D shape retrieval methods have competed each other in these two contests, respectively. To measure and compare the performance of the top participating and other existing promising sketch based 3D shape retrieval methods and solicit the state of the art approaches, we perform a more comprehensive comparison of fifteen best (four top participating algorithms and eleven additional state of the art methods) retrieval methods by completing the evaluation of each method on both benchmarks. The benchmarks, results, and evaluation tools for the two tracks are publicly available on our websites [1,2].

1. Introduction

Sketch based 3D model retrieval is focusing on retrieving relevant 3D models using sketch(es) as input. This intuitive and convenient scheme is easy for users to learn and use to search for 3D models. It is also popular and important for related applications such as sketch based modeling and recognition, as well as 3D animation production via 3D reconstruction of a scene of 2D storyboard [3].

However, most existing 3D model retrieval algorithms target the Query by Model framework, that is, using existing 3D models as queries. In the areas of content based 2D image retrieval and image synthesis, sketch based methods have been addressed for some time now. In 3D model retrieval, on the other hand, less work has to date considered the Query by Sketch framework. In fact, it is a non trivial task to perform sketch based 3D model retrieval and also more difficult compared with the Query by Model case. This is because there exists a semantic gap between the sketches humans draw and the 3D models in the database, implying that the structure of the query and target objects differ. Specifically, target objects are typically given as precisely modeled objects, while the query sketch may differ drastically in level of detail, abstraction, and precision. In addition, until now there is no comprehensive evaluation or comparison for the large number of available sketch based retrieval algorithms. Considering this, we organized the Shape Retrieval Contest (SHREC) 2012 track on Sketch Based 3D Shape Retrieval [1,4], held in conjunction with the fifth Eurographics Workshop on 3D Object Retrieval, to foster this challenging research area by providing a common small scale sketch based retrieval benchmark and soliciting retrieval results from current state of the art retrieval methods for comparison. We also provided corresponding evaluation code for computing a set of performance metrics similar to those typically used to evaluate Query by Model techniques. The objective of this track was to evaluate the performance of different sketch based 3D model retrieval algorithms using both hand drawn and standard line drawings sketch queries on a watertight 3D model dataset. Every participant performed the queries and sent us their retrieval results. We then did the performance assessment.

☆ This paper has been recommended for acceptance by Nicu Sebe.

* Corresponding author. Address: 601 University Drive, Department of Computer Science, Texas State University, San Marcos, TX 78666, United States. Fax: +1 512 245 8750.

E-mail address: lu@txstate.edu (Y. Lu).

A satisfactory success has been achieved in the SHREC'12 sketch track [4]. However, the contest has limitations in terms of its evaluation of different sketch based retrieval algorithms based on a rather small benchmark and a comparison of a limited number of methods. Eitz et al. [5] provided us the largest sketch based 3D shape retrieval benchmark until 2012, based on the Princeton Shape Benchmark (PSB) [6] with one user sketch for each PSB model. However, until now no comparative evaluation has been done on a very large scale sketch based 3D shape retrieval benchmark. Considering this and encouraged by the successful sketch based 3D model retrieval track in SHREC'12 [4], in 2013 we organized another track [2,7] with a similar topic in SHREC'13 to further foster this challenging research area by building a very large scale benchmark and soliciting retrieval results from current state of the art retrieval methods for comparison. Similarly, we also provided corresponding evaluation code for computing the same set of performance metrics as the SHREC'12 sketch track. For this track, the objective was evaluating the performance of different sketch based 3D model retrieval algorithms using a large scale hand drawn sketch query dataset for querying from a generic 3D model dataset.

After finishing the above two SHREC contests, we have found that the participating methods for the two contests are not completely the same, thus a conclusion of the current state of the art algorithm is still unavailable. In addition, to provide a more complete reference for the researchers in this research direction, it is necessary to perform a more incisive analysis on different participating methods w.r.t their scalability and efficiency performance, as well as the two benchmarks used in the two contest tracks. Motivated by the above two findings, we decided to perform a follow up study by completing a more comprehensive evaluation of currently available top sketch based retrieval algorithms on the two benchmarks such as to perform a more comprehensive comparison on them and solicit the state of the art approaches. Thus, we sent invitations to the participants as well as the authors of recently published related papers (according to our knowledge) to ask them to contribute to the new comprehensive evaluation. Totally, 6 groups accepted our invitations and agreed to submit their results on schedule. Finally, 15 best performing methods (4 top participating algorithms and 11 additional state of the art approaches; totally 17 runs) from 4 groups successfully submitted their results, including running results (e.g. retrieval lists and timing information) and method description, which are also available on the SHREC'12 and SHREC'13 sketch track website [1,2]. After that, we performed a comparative evaluation on them.

In this paper, we first review the related work (w.r.t. techniques and benchmarks, respectively) in Section 2. Then, in Section 3 we introduce the two benchmarks (one small scale and one large scale) used in the two contest tracks. Section 4 gives a brief introduction of the contributors of the paper. A short and concise description for each contributed method is presented in Section 5. Section 6 describes the evaluation results of the 15 sketch based 3D retrieval algorithms on the SHREC'12 small scale benchmark and SHREC'13 large scale benchmark, respectively. Section 7 further comments on the benchmarks and analyzes the contributed algorithms w.r.t the performance they achieved. Section 8 concludes the paper and further lists several future research directions.

2. Related work

2.1. Sketch based 3D model retrieval techniques

Existing sketch based 3D model retrieval techniques can be categorized differently according to dissimilar aspects: *Local versus global 2D features*; *Bag of Words framework versus direct shape feature matching*; *Fixed views versus clustered views*; *With versus without view selection*. In this section, we will review some typical recent work in this field.

In 2003, Funkhouser et al. [8] developed a search engine which supports both 2D and 3D queries based on an extended version of 3D spherical harmonics [9] from 3D to 2D. Yoon et al. [10] and Saavedra et al. [11] developed their sketch based 3D model retrieval algorithms based on suggestive contours [12] feature views sampling and diffusion tensor fields feature representation or structure based local approach (STELA). Aono and Iwabuchi [13] proposed an image based 3D model retrieval algorithm based on the Zernike moments and Histogram of Oriented Gradient (HOG) features. Eitz et al. [14 16,5,17] implemented their sketch based 2D/3D object retrieval algorithms by utilizing the Bag of Words framework and local features including HOG and its modified versions, as well as a feature named Gabor local line based feature (GALIF). Shao et al. [18] developed an efficient and robust contour based shape matching algorithm for sketch based 3D model retrieval. Li and Johan [19] performed "View Context" [20] based 2D sketch 3D model alignment before 2D 3D matching based on relative shape context matching [21]. Li et al. [22] further developed a sketch based 3D model retrieval algorithm based on the idea of performing sketch recognition before sketch model matching.

Recently, 2D line drawings have also been utilized to reconstruct correspondent 3D models, which often involves sketch based 3D shape retrieval techniques. Several line drawing based reconstruction algorithms [23 25] have been proposed based on the idea of 2D parts separation, 3D parts search and combination to create a 3D model based on its 2D line drawing. On the other hand, Xie et al. [26] developed a sketching based 3D modification and variation modeling interface based on the idea of parts assembly. Further, Sketch2Scene [27] builds a 3D scene based on a 2D scene sketch by incorporating an analysis of structural context information among the objects in the 2D scene.

2.1.1. 2D feature views

Matching 3D models with a 2D sketch requires us to sample and render appropriate 2D feature views of a 3D model for an as accurate as possible feature correspondence between the 2D and 3D information. In this section, we review typical feature views (some examples are shown in Fig. 1) that have often been used or are promising in sketch based 3D retrieval algorithms.

2.1.1.1. Silhouette view. This is to simply render the black and white image to represent a view of a 3D model. It has been adopted in several latest sketch based 3D model retrieval algorithms, such as Kanai [28], Ohbuchi et al. [4], Aono and Iwabuchi [13].

2.1.1.2. Contour or outline feature view. Contours are a serial of points where the surface blends sharply and becomes invisible to the viewer [12]. They have been utilized in Tatsuma and Aono [29], Aono and Iwabuchi [13], and Li and Johan [19].

2.1.1.3. Suggestive contours feature view [12]. Suggestive contours are contours in the nearby views, that is, they will become contours after rotating the model a little bit. They have been used in the following sketch based 3D model retrieval algorithms: Yoon et al. [10,30], Saavedra et al. [31] and Eitz et al. [5,4] (using both occluding contours and suggestive contours).

2.1.1.4. Apparent ridges. Apparent ridges [32] are defined as the loci of points that maximize a view dependent curvature and they are extensions beyond ridge and valleys [33]. They have been utilized by Eitz et al. [14] in their retrieval algorithm.

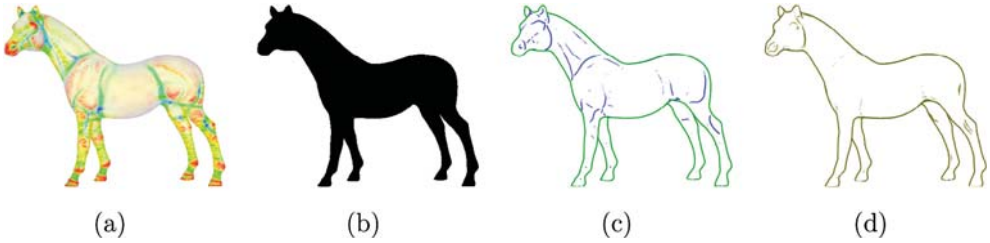


Fig. 1. Example feature views: (a) A horse model (in curvature view); (b) Silhouette feature view; (c) Contours (green) and suggestive contours (blue) feature views; (d) Apparent ridges feature view. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.1.1.5. Other more sophisticated 3D line drawings. Recently, quite a few new and more sophisticated 3D line drawings have been proposed. We regard them as promising in achieving even better results compared with those features mentioned above. They include photic extremum lines (PEL) [34] and its GPU accelerated version GPEL [35], demarcating curves [36], perceptual saliency extremum lines [37], Laplacian lines [38], Difference of Gaussian (DoG) based 3D line drawing [39], as well as the latest multi scale curves on 3D surface [40]. For the classification and characteristics of the above methods, please refer to the survey written by Doug DeCarlo [41].

2.1.2. 2D shape descriptors

For a sketch based 3D model retrieval algorithm, developing or selecting an appropriate 2D shape descriptor is an important part to represent a 2D sketch as well as the 2D feature views of a 3D model, such as those mentioned in Section 2.1.1. In this section, we present several typical and promising 2D shape descriptors for sketch based retrieval.

Fourier descriptor (FD) is an important shape descriptor and has been successfully applied in many pattern recognition related applications such as shape analysis, classification and retrieval as well as character recognition [42]. However, it assumes that we can get the boundary information of a shape beforehand and it does not consider the internal information of the shapes. Considering the above limitations, Zhang and Lu [43,44] extended the Fourier descriptor and proposed a more robust and accurate shape descriptor called generic Fourier descriptor (GFD) which applies Fourier transform on a polar raster sampled shape image.

Zernike moments feature [45] is one typical moment descriptor that outperforms other moments in terms of performance in different applications. For example, 3D Zernike moments [46] feature has been developed to deal with 3D model retrieval. Revaud et al. [47] proposed an improved Zernike moments [45] comparator which considers not only the magnitude of the moments (classic Zernike moments comparator) but also their phase information. They demonstrated its better performance than the classic one.

Local binary pattern [28,48] divides the surrounding regions of any pixel in a binary image into eight directions, computes the percentages of the pixels falling in each bin and regards this distribution information as a local binary pattern (LBP) encoded using an 8 bit binary number, and finally represents the whole image based on the statistical distribution of all the local binary patterns. It can be used to measure the similarity between the 2D sketch after a pre processing and the rendered feature images of a 3D model.

Shape context [21] is a log polar histogram and defines the relative distribution of other sample points with respect to a sample point. It has been successfully applied in diverse tasks. The default shape context definition partitions the surrounding area of a sample point of a 2D shape into 5 distance bins and 12 orientation bins. Thus, the shape context is represented by a 5×12 matrix. Different points have different shape context features in one shape and similar points in two similar shapes usually have similar shape context features. Shape context is scale and transformation invariant but not rotation invariant. To achieve the property of rotation invariance, in [21] a relative frame is defined by adopting the local tangent vector at each point as the reference x axis for angle computation and we refer to it as relative shape context. In addition, Edge histogram [49] can be regarded as an alternative of shape context for sketch representation.

Scale invariant feature transform (SIFT) [50] feature together with the Bag of Features (BoF) framework has many applications in various computer vision research fields. To optimize the search accuracy, efficiency and memory usage in a large scale image retrieval scenario which utilizes SIFT features and BoF framework, Jégou et al. [51] proposed a new compact image representation to aggregate SIFT local descriptors. It achieves a significantly better performance than BoF on condition that the feature vectors used have the same size. In addition, Ohbuchi et al. [52,53] proposed several extended versions of SIFT feature for 3D retrieval, such as Dense SIFT (DSIFT), Grid SIFT (GSIFT) and One SIFT (1SIFT) which also have achieved good retrieval performance.

Histogram of Oriented Gradients (HOG) [54] was first proposed for human detection based on the local and combinational orientation and magnitude distribution of the gradients in each grid of an image. According to the characteristics of a sketch, HOG has been modified and applied in sketch based 2D and 3D object retrieval. For example, to perform a large scale 3D model retrieval, Eitz et al. [17] utilized a simplified HOG (SHOG) feature (first proposed in [16]), which only concerns the orientation information. HOG was also successfully used in sketch based image retrieval, for instance, like [49]. Eitz et al. [16] also performed a comparative evaluation on several 2D shape descriptors for sketch based image retrieval, including HOG, SHOG, local shape context and a modified version of shape context named "Spark" feature.

Other recent 2D shape representations or transforms include tensor representation, which was used in Yoon et al. [10] and Eitz et al. [49], as well as the latest feature Gabor local line based feature (GALIF) [5] which has demonstrated outperforming performance than SIFT [50], Spherical Harmonics [8], and Diffusion Tensor representation [10]. Motivated by the idea of Curvelet transform [55], GALIF is a transformation type feature and it approximates Curvelet by utilizing Gabor filters which only respond to some special frequency and orientation.

2.2. Sketch based 3D shape retrieval benchmarks

2.2.1. Snograss and Vanderwart's standard line drawings (1980)

Snograss and Vanderwart [56] built a dataset of 260 standard line drawings. These sketches were originally designed for experiments in cognitive psychology. They were carefully designed to be comparable regarding four variables fundamental to memory and cognitive processing, including name agreement, image agreement, familiarity, and visual complexity. Their main target is to explore the correlation among those four cognition factors and this pioneer work was followed by several research work with respect

to different languages such as French [57,58], Spanish [59] and Portuguese [60].

2.2.2. Cole et al.'s line drawing benchmark (2008)

Cole et al. [61] built a line drawing benchmark (together with corresponding 3D models) such as to study the relationship between human drawn sketches and computer graphics feature lines. They created line drawings of 12 models including bones, mechanical parts, tablecloths and synthetic shapes. However, either the number of sketches or that of models is very small.

2.2.3. Saavedra and Bustos's sketch dataset (2010)

Saavedra and Bustos [62] built a small sketch dataset (rotation variations of 53 sketches) to test the performance of their sketch based image retrieval algorithm.

2.2.4. Yoon et al.'s sketch based 3D model retrieval benchmark (2010)

To perform sketch based 3D model retrieval and evaluate their algorithm, Yoon et al. [10] built a benchmark which contains 250 sketches for the 260 models of the Watertight Model Benchmark (WMB) dataset [63] and the sketches and models are categorized into 13 classes.

2.2.5. Eitz et al.'s sketch based shape retrieval benchmark

Eitz et al. [5] built a sketch dataset containing one sketch for each of the 1814 models in the Princeton Shape Benchmark (PSB) [6] dataset.

2.2.6. Eitz et al.'s sketch recognition benchmark (2012)

Eitz et al. [17] also built a sketch recognition benchmark which contains 20000 sketches, divided into 250 classes, each with 80 sketches. Currently, it is the most comprehensive sketch dataset.

2.2.7. Comparison with our two benchmarks

The first three datasets or benchmarks cannot be used directly for our purpose while the fourth also has its limitations, such as the bias of different number of sketches per class and lacking of comprehensiveness. Considering these, we have built the SHREC'13 Sketch Track Benchmark using 7200 sketches selected from the large sketch collection presented in [17] as query objects, and the SHREC'12 Sketch Track Benchmark which was extended from Yoon et al.'s benchmark. These two benchmarks either eliminate certain bias or add new evaluation datasets, thus are more comprehensive and objective when used to evaluate existing or newly developed sketch based 3D model retrieval algorithms.

3. Benchmarks

In the SHREC'12 and SHREC'13 sketch tracks, we have built two sketch based 3D model retrieval benchmarks, featuring small scale and large scale benchmarks, and sketches without and with internal features, respectively. In this section, we also introduce several evaluation metrics that are generally used to measure the retrieval performance of a sketch based 3D model retrieval algorithm.

3.1. Small scale benchmark: SHREC'12 Sketch Track Benchmark

3.1.1. 3D target dataset

The 3D benchmark dataset is built based on the Watertight Model Benchmark (WMB) dataset [63] which has 400 watertight models, divided into 20 classes, with 20 models each. The 3D target dataset contains two versions: Basic and Extended. The **Basic version** comprises 13 selected classes from the WMB dataset with each 20 models (in summary, 260 models). In the basic version, all 13 classes are considered relevant for the retrieval challenge. Fig. 2(a) shows one typical example for each class of the basic benchmark. The **Extended version** adds to the basic version all remaining 7 classes of the WMB dataset (each 20 models). These additional classes, however, are not considered relevant for the retrieval challenge but added to increase the retrieval difficulty of the basic version. Fig. 2(b) illustrates typical examples for these remaining 7 irrelevant classes. The extended version is utilized to test the scalability of a sketch based retrieval algorithm.

3.1.2. 2D query set

The 2D query set comprises two subsets, falling into two different types.

- **Hand-drawn sketches.** We utilize the hand drawn sketch data compiled by TU Darmstadt and Fraunhofer IGD [10]. It contains 250 hand drawn sketches, divided into 13 classes. The query sketches were produced by a number of students asked to draw objects from the given categories without any further instructions. The sketches represent a spectrum of different sketching styles and qualities and are used to simulate retrieval by non expert users. They feature sketches with few internal feature lines. One typical example for each class is shown in Fig. 2(c).
- **Standard line drawings.** We also select 12 relevant sketches from the Snograss and Vanderwart's standard line drawings dataset [56]. Note that just one sketch per query class is available in these drawings. Note that these queries are meant as a preliminary first step in eventually building a benchmark which controls for sketch standardization. Owing to their professional design quality the sketches can be considered representing "ideal" queries. Some examples are shown in Fig. 2(d).

In the SHREC'12 sketch track, the two subsets were needed to be tested separately. However, users can also form a query set by combining these two to form a query set which contains diverse types of sketches.

3.2. Large scale benchmark: SHREC'13 Sketch Track Benchmark

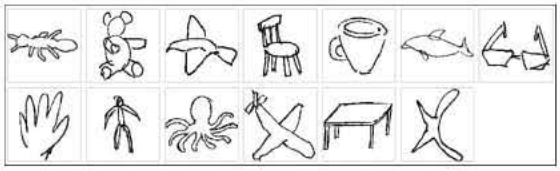
3.2.1. Overview

Our large scale sketch based 3D model retrieval benchmark [2] is built on the latest large collection of human sketches collected by Eitz et al. [17] and the well known Princeton Shape Benchmark (PSB) [6]. To explore how humans draw sketches and for the purpose of human sketch recognition using a crowdsourcing approach, they collected 20000 human drawn sketches, categorized into 250 classes, each with 80 sketches. This sketch dataset is regarded as exhaustive in terms of the number of object categories. Furthermore, it represents a basis for a benchmark which can provide an equal and sufficiently large number of query objects per class, avoiding query class bias. In addition, the sketch variation within each class is high. Thus, we believe a new sketch based 3D model retrieval benchmark built on [17] and the PSB benchmark [6] can foster the research of sketch based 3D object retrieval methods. This benchmark presents a natural extension of the benchmark proposed in [5] for very large scale 3D sketch based retrieval.

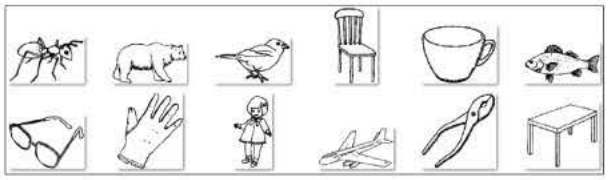
PSB is the most well known and frequently used 3D shape benchmark and it also covers many commonly occurring objects. It contains two datasets: "test" and "train", each has 907 models, categorized into 92 and 90 distinct classes, respectively. Most of the 92 and 90 classes share the same categories with each other. However, PSB has quite different numbers of models for different classes, which is a "target class" bias for retrieval performance evaluation. For example, in the "test" dataset, the "fighter_jet" class has 50 models while the "ant" class only has 5 models. In [5] the query sketch dataset and the target model dataset share the same distribution in terms of number of models in each class.

(a) 13 relevant 3D watertight models classes

(b) 7 irrelevant 3D watertight models classes



(c) Hand-drawn sketches



(d) Standard line drawings

Fig. 2. Typical 3D model and 2D sketch for each class of the SHREC'12 Sketch Track Benchmark.

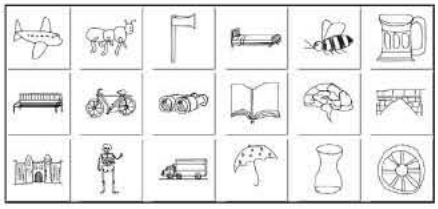
Considering the above fact and analysis, we build the bench mark (available in [2]) by finding common classes in both the sketch [17] and the 3D model [6] datasets. We search for the rele vant 3D models (or classes) in PSB and the acceptance criterion is as follows: for each class in the sketch dataset, if we can find the relevant models and classes in PSB, we keep both sketches and models, otherwise we ignore both of them. In total, 90 of 250 clas ses, that is 7200 sketches, in the sketch dataset have 1258 relevant models in PSB. The benchmark is therefore composed of 7200 sketches and 1258 models, divided into 90 classes. Fig. 3 shows example sketches and their relevant models of 18 classes in the benchmark. We randomly select 50 sketches from each class for training and use the remaining 30 sketches per class for testing, while the 1258 relevant models as a whole are remained as the target dataset. The SHREC'13 sketch track participants need to sub mit results on the training and testing datasets, respectively. To provide a complete reference for the future users of our bench mark, we evaluate the contributed algorithms on both the testing dataset (30 sketches per class, totally 2700 sketches) and the com plete benchmark (80 sketches per class, 7200 sketches).

3.2.2. 2D sketch dataset

The 2D sketch query set comprises the selected 7200 sketches (90 classes, each with 80 sketches), which have relevant models in PSB [6], from Eitz et al.'s [17] human sketch recognition dataset. These sketches often contain internal feature lines. One example indicating the variations within one class is demonstrated in Fig. 4.

3.2.3. 3D model dataset

The 3D model dataset is built on the PSB dataset [6]. The target 3D model dataset comprises 1258 selected models distributed on 90 classes.



(a) Example hand-drawn 2D sketches in Eitz et al.'s sketch dataset

(b) Example relevant 3D models in PSB benchmark

Fig. 3. Example 2D sketches and their relevant 3D models in the SHREC'13 Sketch Track Benchmark.

Fig. 4. An example of intra-class variations of hand-drawn sketches: 35 sketches of the "cell phone" class.

3.3. Evaluation metrics

To have a comprehensive evaluation of a sketch based 3D model retrieval algorithm based on the above two benchmarks, we employ seven commonly adopted performance metrics in Query by Model retrieval techniques. They are Precision Recall plot (*PR*), Nearest Neighbor (*NN*), First Tier (*FT*), Second Tier (*ST*), E Measures (*E*), Discounted Cumulated Gain (*DCG*) [6] and Average Precision (*AP*) [64]. We also have developed the code [1,2] to compute them for the two benchmarks. Their meaning and definitions are listed below.

- **Precision-Recall plot (*PR*):** Precision measures the percentage of the relevant models in the top K ($1 \leqslant K \leqslant n$) retrieval list, where n is the total number of models in the dataset. Recall calculates how much percentage of the relevant class in the database has been retrieved in the top K retrieval list.
- **Nearest Neighbor** (***NN***): *NN* is the precision of top 1 retrieval list.
- **First Tier** (***FT***): Assume there are C relevant models in the database, *FT* is the recall of the top C 1 retrieval list.
- **Second Tier** (***ST***): Similarly, *ST* is the recall of the top $2(C\ 1)$ retrieval list.
- **E-Measure** (***E***): E Measure is motivated by the fact that people are more interested in the retrieval results in the first page. Thus, it is defined [6] to measure the retrieval performance of the top 32 models in a retrieval list,

$$E \quad \frac{2}{\frac{1}{P}+\frac{1}{R}}. \tag{1}$$

- **Discounted Cumulated Gain** (***DCG***): Since relevant models appear in the front of the retrieval list are more important than those in the rear of the list, *DCG* is defined as the normalized summed weighted value related to the positions of the relevant models. A retrieval list R is first transformed into a list G, where $G_i \quad 1$ if R_i is a relevant model, otherwise $G_i \quad 0$. DCG is then defined as follows.

$$DCG_i \quad \begin{cases} G_1 & i \quad 1 \\ DCG_{i\ 1} + \frac{G_i}{\lg_2 i} & \text{otherwise} \end{cases} \tag{2}$$

Finally, it is normalized by the optimal *DCG*,

$$DCG \quad \frac{DCG_n}{1+\sum_{j\ 2}^{C} \frac{1}{\lg_2 j}} \tag{3}$$

where n is the total number of models in the dataset and C is the total number of relevant models in the class.

- **Average Precision** (***AP***): *AP* is to measure the overall performance and it combines precision, recall as well as ranking positions. It can be computed by counting the total area under the Precision Recall plot curve. A good *AP* needs both high recall and precision.

4. Contributors

The first four authors of this paper built the above two benchmarks, and organized the SHREC'12 and SHREC'13 tracks on the topic of sketch based 3D retrieval and this follow up study. Totally, 4 groups successfully contributed the following 15 methods (17 runs), including 4 top algorithms in the SHREC'12 and SHREC'13 sketch tracks (performance of other participating methods can be found in [1,4,2,7]) which are SBR 2D 3D, SBR VC, BF fDSIFT (a modified version of DSIFT) and FDC, as well as 11 additional state of the art methods.

- **BF-fDSIFT**, **BF-fGALIF**, **BF-fGALIF + BF-fDSIFT**; **CDMR-BF-fDSIFT**, **CDMR-BF-fGALIF**, **CDMR-BF-fGALIF + CDMR-BF-fDSIFT**; **UMR-BF-fDSIFT**, **UMR-BF-fGALIF** and **UMR-BF-fGALIF + UMR-BF-fDSIFT** submitted by Takahiko Furuya, Takahiro Matsuda, and Ryutarou Ohbuchi from the University of Yamanashi, Japan (Section 5.1).
- **SBR-2D-3D_NUM_100**, **SBR-2D-3D_NUM_50**, **SBR-VC_NUM_100** and **SBR-VC_NUM_50** submitted by Bo Li and Yijuan Lu from Texas State University, USA; and Henry Johan from Fraunhofer IDM@NTU, Singapore (Sections 5.2 and 5.3).
- **Hierarchical Topology 3D Descriptor** submitted by Pedro B. Pascoal, Alfredo Ferreira and Manuel J. Fonseca from Instituto Superior Técnico/ Technical University of Lisbon/ INESC ID, Portugal (Section 5.4).
- **HELO-SIL**, **HOG-SIL**, and **FDC** submitted by Jose M. Saavedra from University of Chile, Chile and ORAND S.A., Chile and Benjamin Bustos from University of Chile, Chile (Section 5.5).

5. Methods

5.1. Distance metric learning on Bag of Densely sampled local features for sketch based 3D shape retrieval [65,4], by T. Furuya, T. Matsuda and R. Ohbuchi

5.1.1. Method overview

To compare a hand drawn sketch to a 3D model, most of existing methods compare a human drawn 2D 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 human drawn sketches contain stylistic variation, abstraction, inaccuracy and instability, these sketches are often dissimilar to rendered images of 3D models. The entries of their methods employ *unsupervised distance metric learning* to overcome this gap.

First approach, called *Uniform Manifold Ranking*, or *UMR*, is of unsupervised kind. It treats a feature extracted from a sketch and a feature (e.g., BF GALIF [5]) extracted from a view of a 3D model on the same ground. From a set of features, which include features of both sketches and 3D models, the UMR learns a graph structure or a *Uniform Manifold* (*UM*) that reflects low dimensional structure of features. (It is called "uniform" as same feature extraction algo rithm is used for both sketches and rendered images of 3D models, and these features are meshed into a single manifold graph.) Assuming N_m 3D models rendered from N_v viewpoints, and N_s sketches, total of $N_s + (N_m \times N_v)$ features are connected to form the UM. Then diffusion distance from the feature of the sketch query to the features of multi view renderings of 3D models are computed by using *Manifold Ranking* (*MR*) algorithm proposed by Zhou et al. [66].

In the experiments, they use either the *BF fGALIF*, which is a modified version of BF GALIF by Eitz et al. [5] (Fig. 5(a)), or the *BF fDSIFT*, which is a regressed version of BF DSIFT [4] (Fig.5(b)), to form the UM.

Second approach, called *Cross Domain Manifold Ranking*, or *CDMR*, may be either unsupervised, semi supervised, or supervised [65]. For the experiment described in this paper, they used the CDMR in the unsupervised mode. It is called cross domain since it tries to bridge the gap between features extracted from two het erogeneous domains, i.e., hand drawn sketch images and multi view rendered images of 3D models. Unlike UMR, which forms a manifold of features by using single feature, CDMR allows for the use of multiple measures of similarities, both feature based and semantic label based, to form an integrated *Cross Domain Manifold* (*CDM*) that spans heterogeneous domains (See Fig. 6). A set of sketch images are formed into manifold of sketch images by using a feature (e.g., BF GALIF [5]) optimal for sketch to sketch compar ison. Another manifold that connects 3D models are formed by using a feature (e.g., BF DSIFT [52]) that is optimal for comparison among 3D models. These two manifolds are then cross linked by using a feature that is adept at comparing a sketch image to a view of 3D model, e.g., [5]. The CDM is a graph containing $N_s + N_m$ ver tices. Additionally, if available, class labels may also be used for cross linking the domains. Semantic labels help significantly if a sketch (e.g., a stick figure human) is dissimilar to multi view ren derings of 3D models (e.g., a realistic 3D model of human). Similar ity from a sketch query to a 3D model is computed by using MR [66] algorithm, as is the case with the UMR. The diffusion of rele vance value originates from the query, and spreads via edges of the CDM to 3D models. The diffusion occurs among 2D sketches, from 2D sketches to 3D models across the domain, and among 3D models. Note that, if a corpus of sketches is available in the database, the CDMR automatically performs a form of query expansion. The relevance is first diffused from the query to its neighboring sketches. Then, these sketches (expanded query set) behave as multiple secondary sources of diffusion.

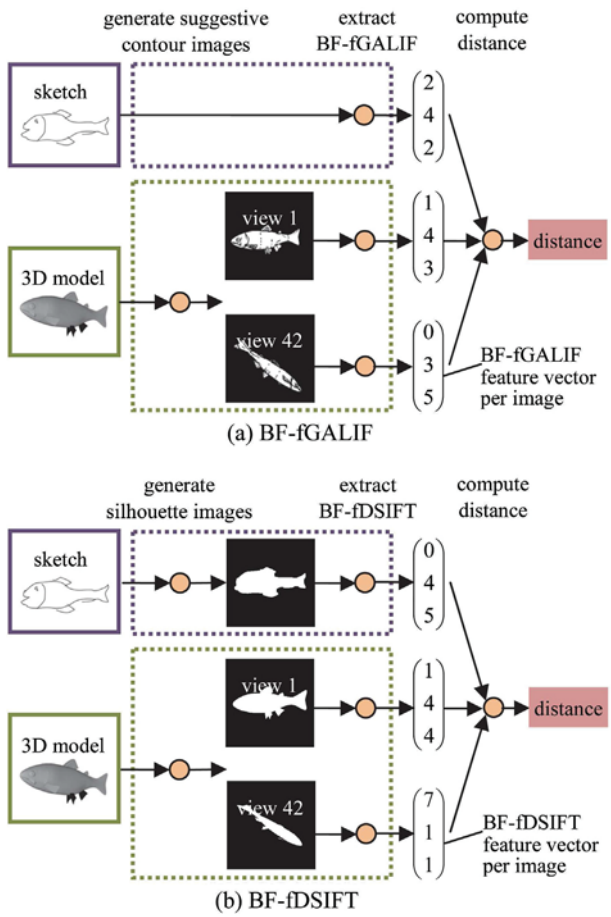


Fig. 5. Image-based feature comparison processes (applicable to fixed distance, UMR, and CDMR).

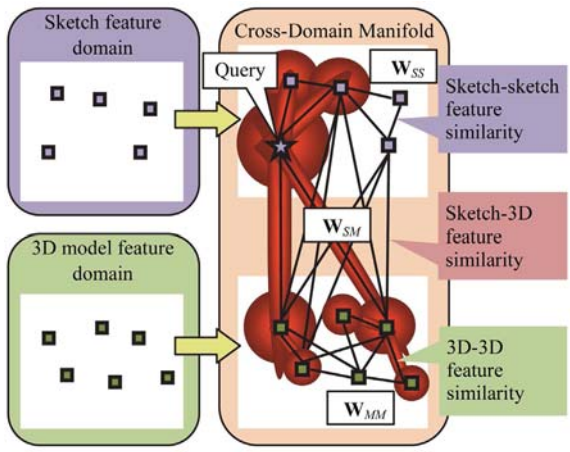


Fig. 6. Feature comparison using Unsupervised Cross-Domain Manifold Ranking (CDMR).

For CDMR, they use two different features fit for the purposes to form subparts of the CDM. To form a manifold of 2D sketches, they use BF fGALIF. To form a manifold of 3D models, they use BF DSIFT [52]. To link sketches with 3D models using feature similarity, they use either BF fGALIF or BF fDSIFT (Fig.5).

Descriptions of the BF fGALIF and BF fDSIFT features will be presented in the next section. For the BF DSIFT, please refer to the original paper by Furuya et al. [52]. They then briefly describe the UMR and the CDMR [65] algorithms.

5.1.2. Features used in the experiments

5.1.2.1. BF fGALIF. BF GALIF [5] proposed by Eitz et al. is designed for sketch based 3D model retrieval. Their variation called *BF fGALIF* (Fig.5(a)) is similar but not identical to the original.

For each 3D model, the model is rendered into Suggestive Contour (SC) [12] images from multiple viewpoints and a set of fGALIF features is computed for each view. They use 42 viewpoints spaced uniformly in solid angle and image resolution of 256×256 pixels. Unlike original GALIF, their fGALIF uses black background for the SC images (see Fig.5(a)).

Each rendered image is then normalized for rotation. To do so, they exploit response images produced by Gabor filtering on the rendered image. Gabor filter captures orientation of lines and intensity gradient in the image. For each pixel in the response image, a response vector is calculated according to the direction of Gabor filter and response magnitude at the pixel. The response vectors calculated at all the pixels in the image are voted against a histogram. The histogram comprises 18 orientation bins and voting is done according to orientation and magnitude of the response vectors. After voting, the image is rotated to the direction of the most populated orientation bin.

After normalizing for the rotation, fGALIF features are extracted densely at regular grid points on the image. They extract 1024 fGALIF features per image. Bandwidth and other parameters for the Gabor filter are determined through preliminary experiments so that the retrieval accuracy is the highest among the combinations of parameters they tried.

For each sketch image, fGALIF features are computed after the image is resized to 256×256 pixels. Computation of fGALIF is carried out in the same manner as for a sketch image and for a rotation normalized SC image of a 3D model.

The set of 1024 fGALIF features extracted from an image is integrated into a BF fGALIF feature vector per image by using a standard Bag of Features (BF) approach. This integration reduces cost of image to image matching significantly compared with directly comparing a set of features to another set of features. They used vocabulary size of 2500. They used k means clustering to learn the vocabulary, and used kd tree to accelerate vector quantization of fGALIF features into words of the vocabulary.

5.1.2.2. BF fDSIFT. Original Bag of Features Dense SIFT (BF DSIFT) [52] computes a feature per 3D model for comparison among 3D models. Here, they use a variant of it, called *BF fDSIFT* [4] (Fig.5(b)) to compare a sketch image with multiple images rendered from multiple views of a 3D model.

The BF fDSIFT turns both sketch and 3D model into silhouette images for comparison. To turn a line drawing sketch with possible gaps in its circumference into a silhouette, dilation operation to close the gaps is followed by area filling. Some of the sketches fail to become silhouettes, but they tolerate them. To turn a 3D model into a set of silhouette images, it is rendered from 42 viewpoints into silhouettes of 256×256 pixels each.

On each silhouette image, SIFT [50] features are densely and randomly sampled. They extract 1200 SIFT features per image. There is no need to normalize images for rotation as SIFT is inherently invariant (to some extent) to rotation, translation, and scaling. The set of 1200 SIFT features extracted from an image is integrated into a BF fDSIFT feature vector per image by using the BF approach. They used vocabulary size of about 10000. They use ERC Tree [67] algorithm to accelerate both vocabulary learning (clustering) and vector quantization of SIFT features.

5.1.3. Ranking retrieval results

For the experiments, similarity ranking of retrieval results are performed by using three different algorithms; fixed distance, the UMR, and the CDMR.

5.1.3.1. Fixed distance. Symmetric version of Kullback Leibler Divergence (KLD) is used as fixed distance metric between a pair of BF features. KLD performs well when comparing a pair of probability distributions, i.e., histograms. Distance between a sketch and a 3D model is the minimum of the 42 distances computed from a BF feature of the sketch and a set of 42 BF features of the 3D model.

5.1.3.2. Uniform Manifold Ranking (UMR). Input of the UMR is the BF fGALIF or the BF fDSIFT features of the sketches and the rendered images of 3D models. A graph is represented as a sparse matrix $\mathbf{W}$ of size $(N_s + N_m \quad N_v) \times (N_s + N_m \quad N_v)$ where N_s and N_m are the number of sketches and 3D models in a database respectively, and N_v is the number of views for rendering (i.e., $N_v \quad 42$). The similarity between vertices i and j is computed by Eq. (4) where $d\,(\mathbf{x}_i, \mathbf{x}_j)$ is KLD between feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, and kNN $(\mathbf{x}_i)$ is a set of k nearest neighbors of $\mathbf{x}_i$.

$$\mathbf{W}_{ij} \quad \begin{cases} \exp\left(\frac{d(\mathbf{x}_i, \mathbf{x}_j)^2}{\sigma} \right) & \text{if } \mathbf{x}_j \in k\text{NN}(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

They normalize $\mathbf{W}$ for $\mathbf{S}$,

$$\mathbf{S} \quad \mathbf{D}^{\frac{1}{2}}\mathbf{W}\mathbf{D}^{\frac{1}{2}} \tag{5}$$

where $\mathbf{D}$ is a diagonal matrix whose diagonal element is $\mathbf{D}_{ij} \quad \sum_j \mathbf{W}_{ij}$. They use the following iterative form of the MR to find relevance values in $\mathbf{F}$ given initial value, or "source" matrix $\mathbf{Y}$. A higher relevance means a smaller distance.

$$\mathbf{F}_{t+1} \quad \alpha\mathbf{S}\mathbf{F}_t + (1 \quad \alpha)\mathbf{Y} \tag{6}$$

$\mathbf{Y}$ is a diagonal matrix of size $(N_s + N_m \quad N_v) \times (N_s + N_m \quad N_v)$ that defines source(s) of relevance value diffusion. If a vertex i is the source of diffusion $\mathbf{Y}_{ii} \quad 1$ and, if not, $\mathbf{Y}_{ii} \quad 0$. In their case, the vertex corresponding to the query sketch becomes the source of diffusion. $\mathbf{F}_{ij}$ is the relevance score of the rendered image j given the sketch i. Hence $\mathbf{F}_{ij}$ is the adaptive distance derived from the MR. Final relevance score between the sketch and the 3D model is the maximum of the 42 scores computed between the sketch and a set of 42 rendered images of the 3D model.

They add prefix "UMR " before the feature extraction method (e.g., UMR BF fGALIF or UMR BF fDSIFT) to indicate UMR processed algorithms. Parameters for the UMR (i.e., k, σ, α) are determined through preliminary experiments. Table 1 shows combination of the parameters for the UMR BF fGALIF and the UMR BF fDSIFT.

To further improve retrieval accuracy, they experimented with combining the BF fGALIF and BF fDSIFT features via a late fusion approach, that is, simply adds distances due to BF fGALIF and BF fDSIFT. Here, each distance may be treated with UMR or not. They denote the combination of fixed distances by "*BF fGALIF + BF fDSIFT*" and the combination of adaptive distances by "*UMR BF fGALIF + UMR BF fDSIFT*".

5.1.3.3. Cross Domain Manifold Ranking (CDMR). The CDM graph is represented as a matrix $\mathbf{W}_{CDM}$ whose vertices are the features from the sketch domain and the 3D model domain. $\mathbf{W}_{CDM}$ is size of $(N_s + N_m) \times (N_s + N_m)$ where N_s and N_m are the number of sketches and 3D models in a database respectively.

$$\mathbf{W}_{CDM} \quad \begin{pmatrix} \mathbf{W}_{SS} & \mathbf{W}_{SM} \\ \mathbf{W}_{MS} & \mathbf{W}_{MM} \end{pmatrix} \tag{7}$$

Table 1
Parameters for the UMR.

Benchmark	Method	k	σ	α
SHREC'12	UMR-BF-fGALIF	30	0.0001	0.9
	UMR-BF-fDSIFT	140	0.0050	0.9
SHREC'13	UMR-BF-fGALIF	100	0.0025	0.9
	UMR-BF-fDSIFT	160	0.0050	0.9

The submatrix $\mathbf{W}_{SS}$ having size $N_s \times N_s$ is the manifold of sketch features. Similarity between a pair of sketches is computed by using the BF fGALIF. For sketch to sketch feature comparison, they do not normalize for image rotation as most sketches drawn by human are already aligned to a canonical orientation. The submatrix $\mathbf{W}_{MM}$ having size $N_M \times N_M$ is a manifold of features of 3D models. Similarity between a pair of 3D models is computed by using their 3D model to 3D model comparison method BF DSIFT [52]. The submatrix $\mathbf{W}_{SM}$ of size $N_S \times N_M$ couples two submanifolds $\mathbf{W}_{MM}$ and $\mathbf{W}_{SS}$ that lie in different domains, that are, sketch feature domain and 3D model feature domain. Similarity between a pair of a sketch and a 3D model is computed by using the BF fGALIF or the BF fDSIFT described in Section 5.1.2. The submatrix $\mathbf{W}_{MS}$ of size $N_M \times N_S$ is a zero matrix as they assume no diffusion of similarity occurs from 3D models to sketches.

For each submatrix, the similarity between vertices i and j is computed by Eq. (8) where $d(\mathbf{x}_i, \mathbf{x}_j)$ is KLD between feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. The parameter σ controls diffusion of relevance value across the CDM. They use different values σ_{SS}, σ_{MM}, and σ_{SM} for each of the submatrices $\mathbf{W}_{SS}, \mathbf{W}_{MM}$, and $\mathbf{W}_{SM}$.

$$\mathbf{W}_{ij} \quad \begin{cases} \exp\left(\frac{d(\mathbf{x}_i,\mathbf{x}_j)}{\sigma}\right) & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

After generating the CDM graph $\mathbf{W}_{CDM}$, the MR is applied on $\mathbf{W}_{CDM}$ to diffuse relevance value from the sketch query to the 3D models over the CDM across the domain boundary.

They normalize $\mathbf{W}_{CDM}$ for $\mathbf{S}_{CDM}$ by Eq. (5). They use the following closed form of the MR to find relevance values in $\mathbf{F}$ given source matrix $\mathbf{Y}$. $\mathbf{F}_{ij}$ is the relevance value of the 3D model j given the sketch i. A higher relevance means a smaller distance.

$$\mathbf{F} \quad (\mathbf{I} \quad \alpha\mathbf{S}_{CDM})^{1}\mathbf{Y} \tag{9}$$

They add prefix "CDMR " before the feature comparison method used for computing $\mathbf{W}_{SM}$ (e.g., CDMR BF fGALIF or CDMR BF fDSIFT) to indicate CDMR processed algorithms.

Parameters for the CDMR (i.e., $\sigma_{SS}, \sigma_{MM}, \sigma_{SM}$ and α) are determined through preliminary experiments. Table 2 summarizes combination of the parameters for the CDMR BF fGALIF and the CDMR BF fDSIFT.

They also experimented with combining the CDMR BF fGALIF and CDMR BF fDSIFT. They employ a simple late fusion approach identical to the one used for the UMR. They denote the combination of relevance values derived from the two features as "*CDMR BF fGALIF + CDMR BF fDSIFT*".

5.2. Sketch based 3D model retrieval based on 2D 3D alignment and shape context matching (SBR 2D 3D) [19,4,7], by B. Li, Y. Lu and H. Johan

The main idea of the sketch based retrieval algorithm proposed in [19] is that they want to maximize the chances that they have selected the most similar or optimal corresponding views for computing the distances between a 2D sketch and a set of selected sample views of a 3D model, while not adding additional online computation and avoiding the brute force comparison between the sketch and many sample views of the model. They implemented the idea by utilizing a 3D model feature named View Context [20], which has a capability of differentiating different sample views of a 3D model. The candidate views selection rule is as follows: a sample view is replaced with the sketch and if its new View Context is very similar to the original one, then it is regarded as a candidate view. During online retrieval, for each 3D model, a set of candidate views are efficiently shortlisted in the 2D 3D alignment according to their top View Context similarities as that of the sketch. Finally, a more accurate shape context matching [21] algorithm is employed to compute the distances between the query sketch and the candidate sample views. The algorithm is composed of precomputation and online retrieval stages, which are illustrated in Fig. 7. Some important details and modifications about the algorithm are first given below.

Silhouette and outline feature views are respectively selected for View Context feature extraction and shape context based 2D 3D matching. Two sets of examples are shown in Fig. 8. For a query sketch, a silhouette feature view is generated based on the following six steps: binarization, Canny edge detection, morphological closing (infinite times, which means repeating until the image does not change), and filling holes, inversion and resizing into a 256×256 image. The corresponding outline feature view is very easy to obtain based on the silhouette feature view. An integrated image descriptor, which contains region, contour, and geometrical information of the silhouette and outline feature views, is utilized to compute View Context. Considering the large scale retrieval scenario, to reduce computational cost, they set the number of sample points to represent a contour feature view to 50 and only keep the top 4 candidate views during 2D 3D alignment. On the other hand, to save the memory needed to load the shape context features during online retrieval, they use the short integers to code the locations of the 5×12 bins and values during the loading of the precomputed shape context features.

For clarity, the main steps of the algorithm are further described as follows.

Table 2
Parameters for the CDMR.

Benchmark	Method	σ_{SS}	σ_{MM}	σ_{SM}	α
SHREC'12	CDMR-BF-fGALIF	0.010	0.025	0.040	0.9
	CDMR-BF-fDSIFT	0.040	0.015	0.030	0.8
SHREC'13	CDMR-BF-fGALIF	0.006	0.010	0.025	0.9
	CDMR-BF-fDSIFT	0.005	0.010	0.025	0.9

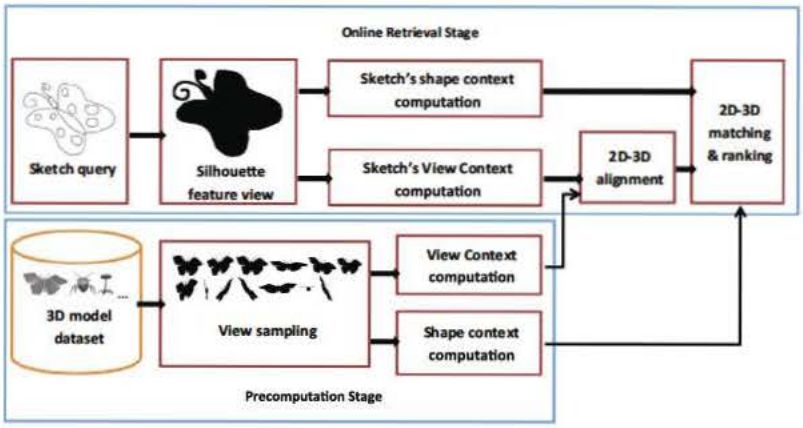


Fig. 7. Flow chart of the sketch-based 3D model retrieval algorithm based on 2D–3D alignment and shape context matching.

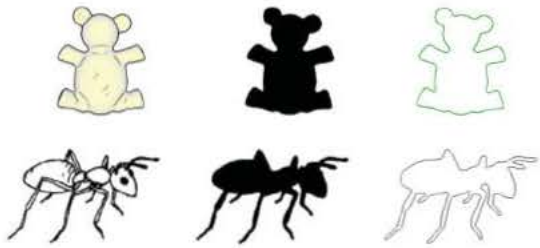


Fig. 8. The feature views of a 3D teddy model and a 2D ant standard line drawing sketch. For each row, from left to right: model/sketch, silhouette view; outline view.

5.2.1. Feature extraction

Silhouette and outline feature views are generated for both 2D sketches and 3D models to effectively and efficiently measure the differences among them.

5.2.2. Feature distance

A computationally efficient integrated image descriptor named *ZFEC* is adopted for View Context computation. It contains a region based Zernike moments feature Z for the silhouette view and a contour based Fourier descriptor feature F for the outline view. Additionally, eccentricity feature E and circularity feature C are also utilized to extract the geometric feature of the outline view. To more accurately measure the difference between the sketch and each candidate view, the relative shape context matching method [21] is adopted.

5.2.3. Sketch's View Context feature extraction

The integrated image descriptor distances between the sketch and all the base views of the target model are computed and the resulting distance vector $D^k \quad \langle d_1, d_2, \ldots, d_m \rangle$ is named sketch's View Context.

5.2.4. 2D 3D alignment

To align the 2D sketch and a 3D model, some candidate views are short listed by keeping a certain percentage (e.g. 20% or 16 sample views for the track) of the sample views with top View Context similarities as the sketch, in terms of correlation similarity S_i,

$$S_i \quad \frac{D_i^s \cdot D^k}{\|D_i^s\| \|D^k\|}, \tag{10}$$

where D_i^s and D^k are the View Contexts of the ith sample view V_i^s of the 3D model and the 2D sketch, respectively.

5.2.5. Sketch Model distance computation

Comparing the sketch with every candidate outline view using the relative shape context matching and regarding the minimum relative shape context distance obtained as the sketch model distance.

5.2.6. Ranking and output

Sorting all the sketch model distances between the sketch and the models in an ascending order and listing the retrieved models accordingly.

The two runs, SBR 2D 3D_NUM_100 (for small scale benchmark, Section 6.1) and SBR VC_NUM_50 (for large scale benchmark, Section 6.2), are two variations of the original SBR 2D 3D by setting the number of sample points for the contour(s) of each sketch, referred to as NUM, to 100 and 50, respectively.

5.3. Sketch based 3D model retrieval based on view clustering and shape context matching (SBR VC) [68,7], by B. Li, Y. Lu and H. Johan

3D models often differ in their visual complexities, thus there is no need to sample the same number of views to represent each model. Motivated by this, a Sketch Based Retrieval algorithm based on adaptive View Clustering and Shape Context matching, named SBR VC, has been proposed. Based on the viewpoint entropy distribution of a set of sample views of a model, they propose a 3D model visual complexity metric, based on which the number of the representative views of the 3D model is adaptively assigned. Then, a Fuzzy C Means view clustering is performed on the sample views based on their viewpoint entropy values and viewpoint locations. Finally, shape context matching [21] is utilized during online retrieval for the matching between a query sketch and the representative views for each target model. The retrieval algorithm comprises precomputation and online retrieval stages. An overview of the algorithm is shown in Fig. 9.

The key component of the retrieval algorithm is viewpoint entropy based adaptive view clustering, which comprises the following three steps.

5.3.1. Viewpoint entropy distribution

For each model, they sample a set of viewpoints by setting the cameras on the vertices of a subdivided icosahedron L_n obtained by n times Loop subdivision on a regular icosahedron L_0. Viewpoint entropy distributions of three models utilizing L_3 for view sampling are demonstrated in Fig. 10. It can be seen that for a 3D model, the complexity of its entropy distribution pattern is highly related to the complexity of its geometry. For instance, the two

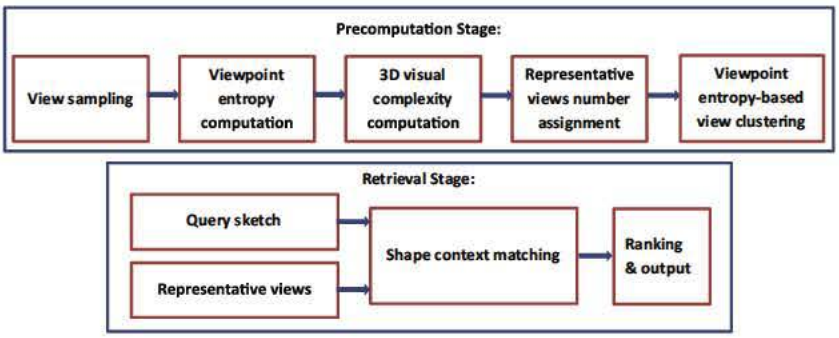


Fig. 9. An overview of the SBR-VC algorithm. The first row is for precomputation while the second row is for retrieval stage.

complex models, horse and Lucy, have a more complicated pattern than the relatively simpler model fish.

5.3.2. Viewpoint entropy based 3D visual complexity

The visual complexity metric is defined based on a class level entropy distribution analysis on a 3D dataset. Mean and standard deviation entropy values m and s among all the sample views of a 3D model are first computed, followed by an average over all the models for each class. 3D visual complexity C is defined as $C = \sqrt{\hat{s}^2 + \hat{m}^2}$, where $\hat{s}$ and $\hat{m}$ are the normalized s and m by their respective maximums over all the classes. The metric is capable of reasonably reflecting the semantic distances among different classes of models.

5.3.3. Viewpoint entropy based adaptive views clustering

Utilizing the visual complexity value C of a model, the number of representative outline feature views N_c is adaptively assigned: $N_c = \frac{C}{2} N_0$ (for small scale benchmark, Section 6.1) or $N_c = \frac{C}{6} N_0$ (for large scale benchmark, Section 6.2), where N_0 is the total number of sample views and it is set to 81 in the algorithm. To speed up the retrieval process on the large scale benchmark in Section 6.2, they choose the parameter setting of $\frac{1}{6}$, compared with the selection of $\frac{1}{2}$ in the originally proposed algorithm. Finally, a Fuzzy C Means view clustering is performed to obtain the representative views.

The two runs, SBR VC_NUM_50 and SBR VC_NUM_100, are two variations of the original SBR VC by setting the number of sample points for the contour(s) of each sketch, referred to as NUM, to 50 and 100, respectively.

5.4. Hierarchical Topology 3D Descriptor (HTD) for sketch based 3D shape retrieval [69,70], by P.B. Pascoal, A. Ferreira and M.J. Fonseca

In order to compare a 3D object with a 2D sketch, it is required to extract 2D views of the object. In the first step, the 3D object is rendered using the Cel shading technique so that the object is drawn with a black outline and interior contour lines.

The black outline is drawn slightly larger than the object itself, and then using backface culling, back facing triangles are hidden due to rendering the object as solid filled. Afterwards, they extract 20 views of the 3D model from different camera positions. This approach is based on the method presented by Chen et al. [71], but instead of using 10 silhouettes they use all the 20 positions of the dodecahedron to get all possible sketches. An overview of the Hierarchical Topology 3D Descriptor process is demonstrated in Fig. 11.

For each image, they then use an algorithm proposed by Ferreira et al. [69] that detects polygons defined by a set of line segments and saves them in a vector format image. The algorithm can be summarized in 4 major steps, as illustrated in Fig. 12.

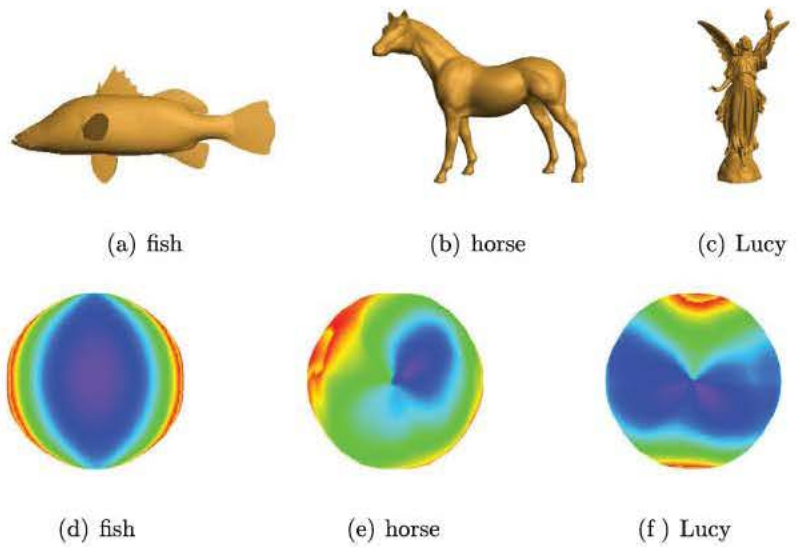


Fig. 10. Viewpoint entropy distribution examples: First row shows the models (front views); Second row demonstrates the viewpoint entropy distribution of each model seen from the viewpoint with respect to its front view. Entropy values are mapped as colors on the surface of the spheres based on HSV color model and smooth shading. Red: small entropy; green: mid-size entropy; blue: large entropy. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

First, it detects the line segment intersections using the Bent ley Ottmann algorithm [72]. Then, creates a graph induced by the drawing, where vertices represent endpoints or proper inter section points of line segments and edges represent maximal rela tively open subsegments that contain no vertices. The third step finds the Minimum Cycle Basis (MCB) [73] of the graph induced in the previous step, using the algorithm proposed by Horton [74]. Last step constructs a set of polygons based on cycles in the previously found MCB. This is straight forward if we transform each cycle into a polygon, where each vertex in the cycle repre sents a vertex in the polygon and each edge in the cycle represents an edge in the polygon.

Finally, for classification, they used a method proposed by Sousa and Fonseca [70] which uses a graph based technique to describe the spatial arrangement of drawing components, combined with geometric information.

Their process starts by applying a simplification step, to remove small visual details while retaining dominant shapes in a drawing. After simplification, they identify visual elements, namely poly gons and lines, and extract geometric and topological information from drawings.

The topology is simplified into the eight topological relation ships defined by Egenhofer and Al Taha [75] (Disjoint, Meet, Over lap, Contain, Inside, Cover, Covered By and Equal), starting from their neighborhood graph for topological relationships. This graph has a well defined structure, with a root node representing the whole drawing and each next level of the graph describing poly gons contained in the blocks identified before, adding more draw ing details. Therefore, by going down in the depth of the graph, we are "zooming in" in drawing details as illustrated in Fig. 13.

The resulting descriptor is a multidimensional vector, whose size depends on graph complexity. Very complex drawings will yield descriptors with higher dimensions, while simple drawings will result in descriptors with lower sizes. To solve this issue, they use the graph spectra to convert graphs into feature vectors, solv ing the problem of isomorphism between topology graphs to the much simpler computation of distances between descriptors. To generate the graph spectrum, firstly it creates the adjacency matrix

Fig. 12. Polygon detection process [69].

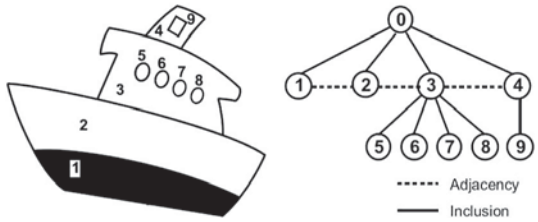


Fig. 13. Drawing and topology graph.

of the graph; secondly calculates its eigenvalues; and finally sorts the absolute values to obtain the topology descriptor as shown in Fig. 14.

As for the geometric information, it uses a general shape recog nition library called CALI [76]. By applying this method to each geometric entity in the figure, it provides complete description of the geometry of a drawing. The geometry and topology descriptors thus computed are combined and used as the descriptor for the respective image.

5.5. Sketch feature extractors based on 3D models silhouettes [7], by J.M. Saavedra and B. Bustos Saavedra

One of the most critical problems when we face the sketch based 3D model retrieval is trying to obtain a good sketch like rep resentation from objects that are not sketches by themselves. Therefore, considering the importance of a good sketch like repre sentation from 3D models, they propose a minimal strategy based

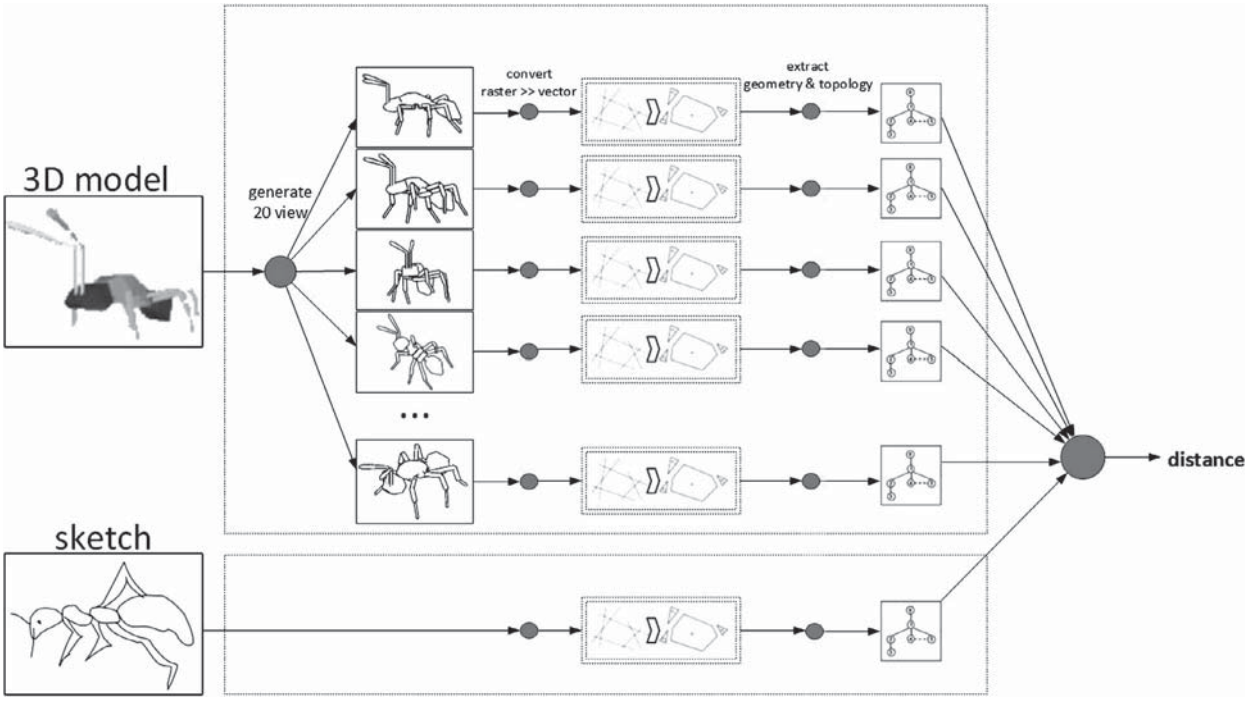


Fig. 11. Overview of the Hierarchical Topology 3D Descriptor process.

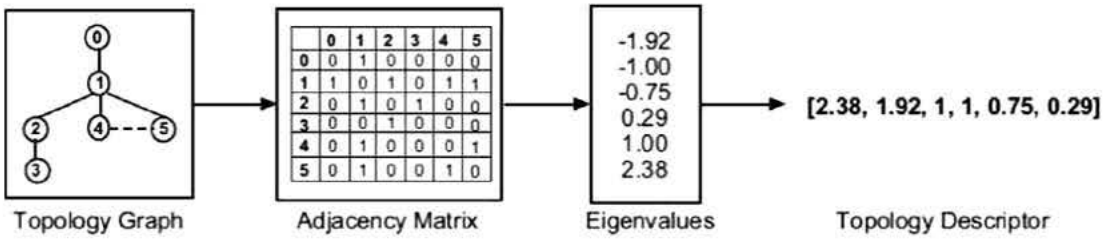


Fig. 14. Block diagram for topology descriptor computation.

on computing an external contour (Silhouette) of 3D models from a defined number of viewpoints. After this stage, they compute a set of low level features for each contour. Finally, they apply a similarity search to get a ranking under an input sketch. These involved processes lead to a framework composed of three stages: (1) Pre processing, (2) Feature extraction, and (3) Similarity search. They describe each one of these stages as follows.

5.5.1. Pre processing

They divide this stage into two sub stages aiming to pre process a 3D model and pre process an input sketch. The goal of both sub stages is to obtain a simple representation that allows the next stage to compute a low level feature in an easier way.

5.5.1.1. 3D model pre processing. They compute 2D projections from each 3D model using six defined viewpoints (top, bottom, right, left, back, front). The external contour of each projection is then extracted discarding internal holes in the underlying image. An example of this representation is shown in Fig. 15.

5.5.1.2. Sketch pre processing. Considering that a sketch is commonly drawn roughly, which produces disconnected strokes, their first stage is to connect all the strokes using a sequence of morphological dilation operations. After that, they extract the external contour of each sketch. Similar to the 3D model pre processing, internal holes of sketches will be discarded. An example of their results in this stage is presented in Fig. 16.

5.5.2. Feature extraction

They take the result produced by the previous stage to compute low level features. In this stage, they propose to use three feature extraction methods: Histogram of Edge Local Orientations (HELO) proposed by Saavedra et al. [62], Histogram of Oriented Gradients (HOG) proposed by Dalal et al. [54], and a Fourier descriptor using the approach presented by Zhang et al. [77].

5.5.2.1. Histogram of Edge Local Orientations based on Silhouettes (HELO SIL). HELO, proposed by Saavedra et al. [62], is a method for computing a histogram of edge orientations in the context of sketch based image retrieval. HELO computes a K bin histogram based on local edge orientations. To get the HELO feature vector, the sketch is divided into a $W \times W$ grid. Then, an edge orientation is estimated for each cell in the grid. Compared with other orientation based approaches, the main difference is that HELO estimates

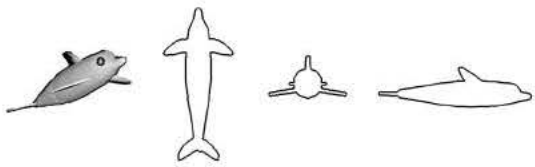


Fig. 15. A 3D model and three silhouettes taken from top, front and left viewpoints respectively.

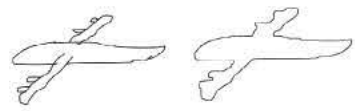


Fig. 16. A sketch and its silhouette representation.

a representative orientation by computing squared gradients inside a cell.

Let $[G_x, G_y]^T$ be the corresponding gradient vector for a pixel (x, y) in a sketch image. The squared gradient is computed for each pixel doubling its gradient angle and squaring its gradient length. To this end, the gradient vector is represented in its corresponding polar coordinates. The squared gradient $[G_{sx}, G_{sy}]^T$ corresponding to the gradient $[G_x, G_y]^T$ is computed as follows,

$$\begin{bmatrix} G_{sx} \\ G_{sy} \end{bmatrix} \quad \begin{bmatrix} G_x^2 \quad G_y^2 \\ 2G_x \cdot G_y \end{bmatrix}. \tag{11}$$

The estimated orientation for a cell is the orientation of the average squared gradient $[\overline{G_{sx}}, \overline{G_{sy}}]$ computed for the cell. Finally, a K bin histogram is computed using the angles of all average squared gradients for each cell.

In order to exploit local information of sketches, they divide the sketch image into 6×6 blocks, and compute a 36 bin HELO descriptor for each block. Then, they concatenate all the local HELO descriptors to get the final feature vector. In addition, they set $W \quad 25$.

5.5.2.2. Histogram of Oriented Gradients based on Silhouettes (HOG SIL). The HOG [54] approach allows us to compute a histogram of gradient orientations by concatenating local orientation histograms computed in small regions. This approach divides the sketch image into $C \times C$ size cells. For each cell a K bin histogram of orientation is computed. In order to increase the robustness of the descriptor, each local histogram is then normalized using information of neighbor histograms. To this end, blocks of $B \times B$ cells are formed. In this point, it is possible to form blocks with or without overlapping.

In their experimental evaluation, they set $C \quad 16$, $B \quad 3$, $K \quad 9$, and they use the non overlapping block approach. In addition, since the size of the descriptor depends on the size of the input image, they resize all the silhouette images, obtained from the 2D projections as well as from input sketches, to 100×100 pixels.

5.5.2.3. Fourier Descriptors on 3D models Silhouettes (FDC). Let X be the set of x coordinates and Y be the set of y coordinates of a boundary. The Fourier descriptor is computed over the centroid distance C which is obtained as follows,

$$C \quad dist_E(X \quad x_c, Y \quad y_c), \tag{12}$$

where (x_c, y_c) is the centroid of the shape and $dist_E$ is the Euclidean distance. To deal with different boundary sizes, they sample 128

points using the *equal arc length sampling* as suggested by Zhang et al. [42].

Next, they apply a Fourier Transform on the sample set. Let F be the computed Fourier descriptor with 256 entries. To deal with the rotation invariance issue, they use the magnitude of the Fourier descriptor. In addition, considering that the sample set is composed of real values, they take only half of the Fourier entries. Finally, to deal with different scales, the descriptor is normalized with respect to F_0, so the final descriptor is given by,

$$FD \quad \left[\frac{|F_1|}{|F_0|}, \frac{|F_2|}{|F_0|}, \ldots, \frac{|F_{128}|}{|F_0|}\right]. \tag{13}$$

5.5.3. Similarity search

To improve the efficiency of the searching, they use a KD Tree index together with the Manhattan distance to search all the database of 3D models. For indexing and retrieving, they use the FLANN (Fast Library for Approximate Nearest Neighbors) implementation provided by Muja et al. [78].

6. Results

6.1. Small scale benchmark: SHREC'12 Sketch Track Benchmark

In this section, we perform a comparative evaluation of the results of the 14 runs submitted by 3 of the 4 groups on SHREC'12 Sketch Track Benchmark (Pascoal's results are not available on this benchmark; Li's SBR VC and SBR 2D 3D select NUM = 100 only). We measure retrieval performance based on the 7 metrics mentioned in Section 3.3: PR, NN, FT, ST, E, DCG and AP. In addition, we also compare their scalability and efficiency.

As described in Section 3.1, there are two versions of target dataset (Basic and Extended) as well as two types of sketch datasets (hand drawn sketches and standard line drawings). This results in four combinations: (1) Hand drawn sketch queries and Basic version of target dataset; (2) Standard line drawing queries and Basic version of target dataset; (3) Hand drawn sketch queries and Extended version of target dataset; and (4) Standard line drawing queries and Extended version of target dataset. Comparisons of the 14 contributed methods for the above four cases are shown in Fig. 17 and Tables 3 6.

First, we start with the overall performance evaluation. As shown in the aforementioned figures and tables, Furuya's CDMR BF fGALIF + CDMR BF fDSIFT performs best, closely followed by CDMR BF fGALIF. Then, CDMR BF fDSIFT, UMR BF fGALIF + UMR BF fDSIFT, UMR BF fGALIF, SBR VC_NUM_100 and SBR 2D 3D_NUM_100 succeed. Performance of the remaining three methods is comparable and the disparity among them is relatively small. In a word, these runs can be ranked in several groups according to the overall performance and we also provide this ranking information ("R") in the following Tables 3 12. If we consider non machine learning approaches, Li's SBR VC and SBR 2D 3D perform best and they outperform either the GALIF or the DSIFT feature based methods and we can expect better performance if we apply "CDMR" on them.

Second, we look into different types of queries. Compared with hand drawn sketch queries, standard line drawing queries usually achieve superior performance. One possible explanation for this is that this dataset only contains a single line drawing per class, which has been carefully created to convey shape as well as salient features of that class.

Third, we asked contributors to also provide timing information, together with their hardware and software configurations, to compare runtime requirements of their methods, based on the second case (hand drawn sketch queries and extended target dataset). Table 8 lists the timing information in seconds and comparison results. In this distributed evaluation, it was not possible to control for the hardware platform (roughly comparable for all groups, though) or implementation efficiency of the setups. However, we believe that the timing information is useful for an approximate comparison of the runtime requirements of the algorithms.

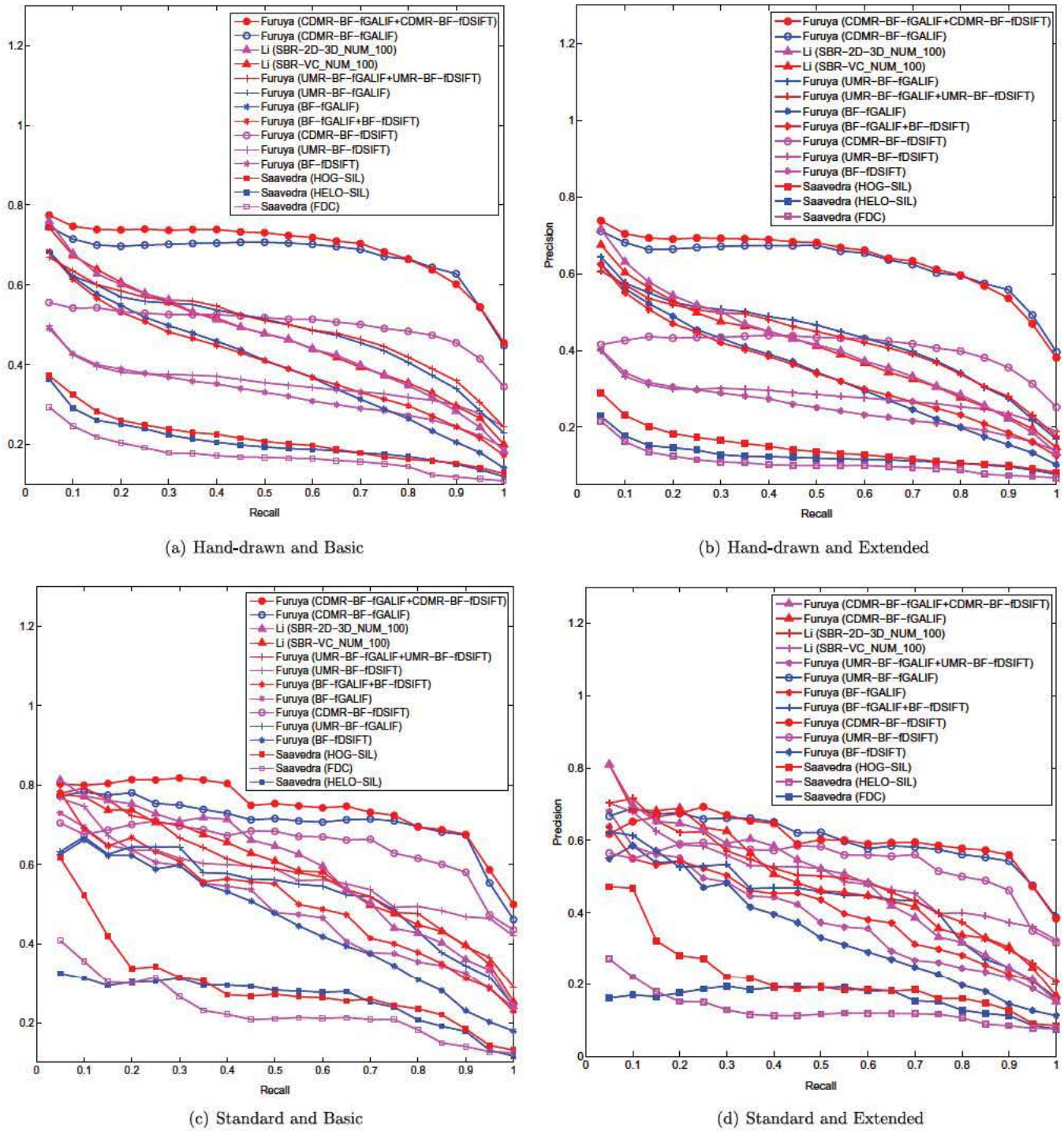
Last but not least, we evaluate the scalability of the different methods to irrelevant models (Section 3.1.1) in the dataset. Table 7 lists the percentage of performance decreases when using the extended target dataset instead of the basic one (using hand drawn sketch queries). Similarly, we also list the ranking information in terms of scalability. Furuya's CDMR BF fGALIF has best scalability, closely followed by CDMR BF fGALIF + CDMR BF fDSIFT. In the Rank 2, UMR BF fGALIF, SBR 2D 3D_NUM_100, UMR BF fGALIF + UMR BF fDSIFT and SBR VC_NUM_100 share similar scalability. Compared with the above six approaches, the other three methods exhibit a stronger decrease in retrieval performance when adding 140 irrelevant models to the target dataset. Please also note that compared with fGALIF, fDSIFT often has more decreases in retrieval performance.

6.2. Large scale benchmark: SHREC'13 Sketch Track Benchmark

Similarly, a comparative evaluation of 16 runs (Li's SBR 2D 3D selects NUM = 50 only) of all the 15 contributed methods has been performed on the latest large scale benchmark of SHREC'13 Sketch Track Benchmark. As described in Section 3.2, the complete query sketch dataset is divided into "Training" and "Testing" datasets, which is to accustom to machine learning based retrieval algorithms. To provide complete reference performance data for both non learning based and learning based approaches, such as the contributed "CDMR " and "UMR " methods, we evaluate the submitted results on both "Training" and "Testing" datasets, as well as the complete sketch dataset. Fig. 18 and Tables 9 11 compare the contributed methods in terms of the 7 performance metrics on the above three datasets, respectively.

Similar as the results on the small scale SHREC'12 Sketch Track Benchmark, the aforementioned figure and table show that Furuya's CDMR BF fGALIF + CDMR BF fDSIFT performs best, closely followed by their CDMR BF fGALIF and CDMR BF fDSIFT. Then, UMR BF fGALIF + UMR BF fDSIFT, BF fGALIF + BF fDSIFT, UMR BF fDSIFT and UMR BF fGALIF succeed, followed by the four methods of BF fDSIFT, BF fGALIF, SBR VC and SBR 2D 3D. HOG SIL, HELO SIL and FDC have comparatively inferior performance, but they outperform HTD. We also can find that "CDMR " based approaches often consistently (either in small scale or large scale retrieval scenarios) achieve better performance than their "UMR " based counterparts. Similarly, if we only compare non machine learning approaches, Li's SBR VC_NUM_100 is comparable to BF fGALIF and BF fDSIFT, which are outperformed by their combinational one. SBR 2D 3D_NUM_50 is also comparable to SBR VC_NUM_50.

However, when compared with the performance obtained on the SHREC'12 Sketch Track Benchmark which employed a much smaller benchmark, the performance of all the methods is much less successful. For example, even for the best performing approach of CDMR BF fGALIF + CDMR BF fDSIFT, the decreases (comparing the performance on the "Complete" benchmark of SHREC'13 Sketch Track Benchmark and the "Hand drawn and Basic" benchmark of SHREC'12 Sketch Track Benchmark) are 61.2%, 68.2%, 62.9%, 71.3%, 44.9% and 63.7% in NN, FT, ST, E, DCG and AP, respectively. This finding is worth noting because it evidently raises the issue of the scalability in the case of large scale sketch based model retrieval.



(a) Hand-drawn and Basic

(b) Hand-drawn and Extended

(c) Standard and Basic

(d) Standard and Extended

Fig. 17. Precision-Recall plot performance comparisons of the four cases of SHREC'12 Sketch Track Benchmark: (a) Hand-drawn sketch queries and Basic version of target dataset. (b) Hand-drawn sketch queries and Extended version of target dataset. (c) Standard line drawing queries and Basic version of target dataset. (d) Standard line drawings queries and Extended version of target dataset.

We noticed that all the retrieval performance metrics values are not high, which is mainly due to the challenges of the benchmark. Firstly, the 80 sketches in a query class represent many variations of an object, which adds the difficulty for accurate retrieval and de serves a higher standard on the scalability of retrieval algorithms. Secondly, as mentioned in Section 3.2.1, the query class bias has al ready been solved by making each query class contain the same number of sketches, while the bias in the target class still exists. There is a large variation in the number of models in different classes. For example, the "airplane" class contains 184 target mod els while the "ant" class only has 5 models. Thus, to accurately re trieve these classes of models in the First Tier and Second Tier is difficult. Therefore, their performance metrics values, especially on NN, FT and ST, are relatively much lower and this happens to all the 15 running methods. One demonstrating example is shown in Fig. 19. More details about the variations in the performance with respect to different classes for each contributed method can be found in the SHREC'13 sketch track homepage [2]. The

Table 3
Other Performance metrics for the performance comparison on the Hand-drawn sketch queries and Basic version of target dataset. "R" denotes the ranking group number.

Contributor	Method	NN	FT	ST	E	DCG	AP	R
Furuya	BF-fDSIFT	0.364	0.295	0.454	0.312	0.610	0.321	3
	BF-fGALIF	0.600	0.367	0.513	0.357	0.689	0.393	3
	BF-fGALIF + BF-fDSIFT	0.596	0.369	0.529	0.365	0.689	0.399	3
	CDMR-BF-fDSIFT	0.504	0.428	0.604	0.418	0.701	0.491	2
	CDMR-BF-fGALIF	0.688	0.617	0.767	0.556	0.813	0.661	1
	CDMR-BF-fGALIF + CDMR-BF-fDSIFT	**0.708**	**0.641**	**0.806**	**0.582**	**0.833**	**0.681**	1
	UMR-BF-fDSIFT	0.396	0.303	0.493	0.333	0.617	0.345	3
	UMR-BF-fGALIF	0.612	0.443	0.576	0.411	0.718	0.478	2
	UMR-BF-fGALIF + UMR-BF-fDSIFT	0.572	0.458	0.619	0.434	0.728	0.486	2
Li	SBR-VC_NUM_100	0.664	0.427	0.587	0.413	0.730	0.461	2
	SBR-2D-3D_NUM_100	0.688	0.415	0.581	0.411	0.731	0.459	2
Saavedra	HELO-SIL	0.228	0.168	0.307	0.196	0.522	0.201	4
	HOG-SIL	0.256	0.183	0.312	0.203	0.529	0.210	4
	FDC	0.164	0.147	0.250	0.164	0.490	0.168	4

Table 4
Other performance metrics for the performance comparison on the Hand-drawn sketch queries and Extended version of target dataset. "R" denotes the ranking group number.

Contributor	Method	NN	FT	ST	E	DCG	AP	R
Furuya	BF-fDSIFT	0.280	0.225	0.370	0.244	0.552	0.246	3
	BF-fGALIF	0.516	0.317	0.446	0.307	0.646	0.332	3
	BF-fGALIF + BF-fDSIFT	0.508	0.317	0.461	0.316	0.648	0.338	3
	CDMR-BF-fDSIFT	0.340	0.349	0.531	0.362	0.631	0.399	2
	CDMR-BF-fGALIF	0.648	0.566	0.741	0.535	0.787	0.616	1
	CDMR-BF-fGALIF + DMR-BF-fDSIFT	**0.668**	**0.585**	**0.770**	**0.554**	**0.801**	**0.625**	1
	UMR-BF-fDSIFT	0.296	0.235	0.406	0.269	0.560	0.274	3
	UMR-BF-fGALIF	0.572	0.387	0.530	0.375	0.684	0.426	2
	UMR-BF-fGALIF + UMR-BF-fDSIFT	0.492	0.392	0.554	0.387	0.684	0.418	2
Li	SBR-VC_NUM_100	0.576	0.372	0.519	0.360	0.682	0.392	2
	SBR-2D-3D_NUM_100	0.628	0.371	0.520	0.364	0.692	0.400	2
Saavedra	HELO-SIL	0.116	0.097	0.183	0.115	0.443	0.122	4
	HOG-SIL	0.188	0.123	0.223	0.139	0.466	0.143	4
	FDC	0.108	0.087	0.170	0.108	0.426	0.104	4

Table 5
Other performance metrics for the performance comparison on the Standard line drawing queries and Basic version of target dataset. "R" denotes the ranking group number.

Contributor	Method	NN	FT	ST	E	DCG	AP	R
Furuya	BF-fDSIFT	0.667	0.442	0.658	0.436	0.747	0.477	3
	BF-fGALIF	0.500	0.421	0.612	0.410	0.722	0.443	3
	BF-fGALIF + BF-fDSIFT	0.667	0.471	0.717	0.510	0.765	0.498	3
	CDMR-BF-fDSIFT	0.667	0.583	0.833	0.580	0.800	0.633	1
	CDMR-BF-fGALIF	**0.750**	0.629	0.792	0.567	0.842	0.694	1
	CDMR-BF-fGALIF + CDMR-BF-fDSIFT	**0.750**	**0.679**	**0.871**	**0.619**	**0.866**	**0.727**	1
	UMR-BF-fDSIFT	**0.750**	0.525	0.733	0.529	0.779	0.563	2
	UMR-BF-fGALIF	0.500	0.487	0.617	0.442	0.752	0.517	2
	UMR-BF-fGALIF + UMR-BF-fDSIFT	**0.750**	0.546	0.692	0.494	0.796	0.568	2
Li	SBR-VC_NUM_100	0.667	0.521	0.713	0.503	0.799	0.570	2
	SBR-2D-3D_NUM_100	**0.750**	0.542	0.700	0.516	0.807	0.580	2
Saavedra	HELO-SIL	0.167	0.221	0.379	0.256	0.550	0.254	4
	HOG-SIL	0.583	0.233	0.413	0.263	0.606	0.289	4
	FDC	0.250	0.208	0.329	0.237	0.549	0.226	4

remaining bias deserves our further improvement, such as making each class contain the same number of 3D models by adding more models from other 3D model benchmarks.

Finally, we have a similar approximate efficiency performance comparison on this benchmark, as shown in Table 12. Obviously, Saavedra's three methods are still the most efficient, followed by Furuya's BF fGALIF and BF fDSIFT. Li's two methods still have the same ranks. However, Furuya's two "CDMR " approaches drop from rank 2 to rank 4 while their two "UMR " methods also drop 1 or 2 places. We compare the changes in timing information be tween the large scale SHREC'13 Sketch Track Benchmark (1258 models) and the small scale SHREC'12 Sketch Track Benchmark (400 models) in Fig. 20. From the numbers, we also can find that the response time of "CDMR " and "UMR " approaches increase a lot with the increase of number of target 3D models. With the scale of the target database is increased by 3.2 times from 400 to 1258 models, the response time of CDMR BF fDSIFT, CDMR BF fGALIF, UMR BF fDSIFT and UMR BF fGALIF is correspondingly increased by about 730.3, 1230.9, 173.2, and 247.0 times; while it is 1.0 and 0.8 for BF fDSIFT, BF fGALIF, and 3.5 times for HOG SIL, HELO SIL and FDC, respectively. While, for SBR VC and SBR 2D 3D, the response time has been reduced 10.7% and 75.4% respec tively by selecting fewer representative views or sample points. In a word, the above best performing approaches CDMR BF fGALIF and CDMR BF fDSIFT have inferior performance on efficiency, thus have much room for further improvement in this regard.

Table 6
Other performance metrics for the performance comparison on the Standard line drawing queries and Extended version of target dataset. "R" denotes the ranking group number.

Contributor	Method	NN	FT	ST	E	DCG	AP	R
Furuya	BF-fDSIFT	0.583	0.337	0.512	0.356	0.684	0.376	3
	BF-fGALIF	0.417	0.337	0.496	0.346	0.656	0.336	3
	BF-fGALIF + BF-fDSIFT	0.417	0.388	0.613	0.394	0.697	0.394	3
	CDMR-BF-fDSIFT	0.500	0.467	0.717	0.494	0.724	0.523	1
	CDMR-BF-fGALIF	0.583	**0.542**	0.717	0.519	**0.781**	**0.590**	1
	CDMR-BF-fGALIF + CDMR-BF-fDSIFT	0.500	0.533	**0.783**	**0.551**	0.778	0.589	1
	UMR-BF-fDSIFT	0.667	0.442	0.671	0.455	0.733	0.490	2
	UMR-BF-fGALIF	0.500	0.371	0.550	0.388	0.701	0.427	2
	UMR-BF-fGALIF + UMR-BF-fDSIFT	0.667	0.433	0.642	0.455	0.739	0.479	2
Li	SBR-VC_NUM_100	**0.750**	0.442	0.629	0.439	0.747	0.477	2
	SBR-2D-3D_NUM_100	**0.750**	0.454	0.625	0.442	0.750	0.476	2
Saavedra	HELO-SIL	0.083	0.121	0.246	0.151	0.454	0.157	4
	HOG-SIL	0.417	0.163	0.271	0.167	0.541	0.214	4
	FDC	0.167	0.100	0.221	0.138	0.455	0.127	4

Table 7
Scalability comparison in terms of performance decrease (%, in terms of percentage of the performance results on the Basic version of target dataset) on the Hand-drawn sketch queries. "R" denotes the ranking group number.

Contributor	Method	NN	FT	ST	E	DCG	AP	R
Furuya	BF-fDSIFT	−23.1	−23.7	−18.5	−21.8	−9.5	−23.4	4
	BF-fGALIF	−14.0	−13.6	−13.1	−14.0	−6.2	−15.5	3
	BF-fGALIF + BF-fDSIFT	−14.8	−14.1	−12.9	−13.4	−6.0	−15.3	3
	CDMR-BF-fDSIFT	−32.5	−18.5	−12.1	−13.4	−10.0	−18.7	3
	CDMR-BF-fGALIF	−5.8	**−8.3**	**−3.4**	**−3.8**	**−3.2**	**−6.8**	1
	CDMR-BF-fGALIF + CDMR-BF-fDSIFT	**−5.6**	−8.7	−4.5	−4.8	−3.8	−8.2	1
	UMR-BF-fDSIFT	−25.3	−22.4	−17.6	−19.2	−9.2	−20.6	4
	UMR-BF-fGALIF	−6.5	−12.6	−8.0	−8.8	−4.7	−10.9	2
	UMR-BF-fGALIF + UMR-BF-fDSIFT	−14.0	−14.4	−10.5	−10.8	−6.0	−14.0	2
Li	SBR-VC_NUM_100	−13.3	−12.9	−11.6	−12.8	−6.6	−15.0	2
	SBR-2D-3D_NUM_100	−8.7	−10.6	−10.5	−11.4	−5.3	−12.9	2
Saavedra	HELO-SIL	−49.1	−42.3	−40.4	−41.3	−15.1	−39.3	5
	HOG-SIL	−26.6	−32.8	−28.5	−31.5	−11.9	−31.9	5
	FDC	−34.2	−40.8	−32.0	−34.2	−13.1	−38.1	5

Table 8
Timing information comparison on the "hand-drawn and extended" dataset of SHREC'12 Sketch Track Benchmark: t is the average response time (in seconds) per query. "R" denotes the ranking group number.

Contributor (with computer configuration)	Method	Language	t	R
Furuya (CPU: Intel Core i7 3930 K @ 3.20 Hz; GPU: NVIDIA GeForce GTX 670; Memory: 64 GB; OS: Ubuntu 12.04)	BF-fDSIFT	C++, CUDA	0.62	2
	BF-fGALIF	C++	0.27	2
	CDMR-BF-fDSIFT	C++	0.83	2
	CDMR-BF-fGALIF	C++	0.50	2
	UMR-BF-fDSIFT	Matlab	314.82	4
	UMR-BF-fGALIF	Matlab	109.77	3
Li (CPU: Intel Core 2 Duo E7500 @ 2.93 GHz; Memory: 16 GB; OS: Windows 7 64-bit)	SBR-VC_NUM_100	C/C++	233.77	4
	SBR-2D-3D_NUM_100	C/C++	178.56	3
Saavedra (CPU: Intel Core i7-3770 CPU @ 3.40 GHz; Memory: 8 GB; OS: Ubuntu 11.10)	HOG-SIL	C++	**0.02**	1
	HELO-SIL	C++	**0.02**	1
	FDC	C++	**0.02**	1

A brief analysis on the computational efficiency of the "CDMR " and "UMR " approaches is as follows. Both approaches are used to improve accuracy of inter feature distance and employ so called "manifold ranking" [66] as its inter feature distance metric learning. Manifold ranking comprises two steps: manifold graph formation and diffusion, which respectively have a upper bound of computa tional complexity of O (N^2) and O (N^3), where N is the total number of feature vectors. For the UMR, the matrix representing the mani fold graph is sparse, thus its actual cost is much less than O (N^3), but still much more than O (N). The UMR forms a graph connecting all the features from all the views (42 views per 3D model) of all the 3D models. Therefore, if there are 1000 models in a database, $N = 42 \times 1000 = 42000$. It is true that this big size, e.g. $N = 42000$, pushes the limit of computational cost, both in terms of space and time. For the CDMR, the matrix representing the manifold graph is not sparse, but a much smaller number of feature vectors (thus smal ler graph) leads to less computational cost than the UMR. A CDMR graph is smaller because it uses only one feature vector for each 3D model. Both UMR and CDMR produce better accuracy than simple distance computation among a pair of features. However, they are computationally expensive, and thus are not scalable for large benchmarks and deserve further study.

Table 9
Performance metrics for the performance comparison on the training dataset of the SHREC'13 Sketch Track Benchmark. "R" denotes the ranking group number.

Contributor	Method	NN	FT	ST	E	DCG	AP	R
Furuya	BF-fDSIFT	0.153	0.099	0.158	0.095	0.354	0.119	3
	BF-fGALIF	0.183	0.101	0.154	0.089	0.355	0.118	3
	BF-fGALIF + BF-fDSIFT	0.210	0.123	0.185	0.106	0.381	0.142	2
	CDMR-BF-fDSIFT	0.217	0.158	0.231	0.136	0.413	0.196	1
	CDMR-BF-fGALIF	0.238	0.175	0.267	0.149	0.428	0.215	1
	CDMR-BF-fGALIF + CDMR-BF-fDSIFT	**0.273**	**0.205**	**0.301**	**0.167**	**0.459**	**0.247**	1
	UMR-BF-fDSIFT	0.160	0.115	0.182	0.107	0.369	0.136	2
	UMR-BF-fGALIF	0.170	0.120	0.185	0.104	0.371	0.135	2
	UMR-BF-fGALIF + UMR-BF-fDSIFT	0.208	0.131	0.201	0.115	0.387	0.152	2
Li	SBR-VC_NUM_100	0.160	0.097	0.149	0.085	0.349	0.113	3
	SBR-VC_NUM_50	0.131	0.082	0.130	0.076	0.333	0.098	3
	SBR-2D-3D_NUM_50	0.133	0.080	0.126	0.075	0.330	0.097	3
Pascoal	HTD	0.019	0.017	0.033	0.018	0.241	0.025	5
Saavedra	HOG-SIL	0.104	0.066	0.104	0.059	0.304	0.082	4
	HELO-SIL	0.104	0.061	0.092	0.052	0.300	0.076	4
	FDC	0.062	0.049	0.085	0.051	0.295	0.062	4

Table 10
Performance metrics for the performance comparison on the testing dataset of the SHREC'13 Sketch Track Benchmark. "R" denotes the ranking group number.

Contributor	Method	NN	FT	ST	E	DCG	AP	R
Furuya	BF-fDSIFT	0.145	0.099	0.154	0.093	0.351	0.115	3
	BF-fGALIF	0.176	0.101	0.156	0.091	0.354	0.119	3
	BF-fGALIF + BF-fDSIFT	0.213	0.123	0.186	0.107	0.379	0.143	2
	CDMR-BF-fDSIFT	0.217	0.156	0.231	0.135	0.411	0.193	1
	CDMR-BF-fGALIF	0.242	0.174	0.263	0.146	0.427	0.215	1
	CDMR-BF-fGALIF + CDMR-BF-fDSIFT	**0.279**	**0.203**	**0.296**	**0.166**	**0.458**	**0.246**	1
	UMR-BF-fDSIFT	0.154	0.113	0.178	0.104	0.366	0.133	2
	UMR-BF-fGALIF	0.159	0.119	0.179	0.102	0.367	0.131	2
	UMR-BF-fGALIF + UMR-BF-fDSIFT	0.209	0.131	0.195	0.113	0.386	0.152	2
Li	SBR-VC_NUM_100	0.164	0.097	0.149	0.085	0.348	0.114	3
	SBR-VC_NUM_50	0.132	0.082	0.131	0.075	0.331	0.098	3
	SBR-2D-3D_NUM_50	0.132	0.077	0.124	0.074	0.327	0.095	3
Pascoal	HTD	0.017	0.016	0.031	0.018	0.240	0.024	5
Saavedra	HOG-SIL	0.110	0.069	0.107	0.061	0.307	0.084	4
	HELO-SIL	0.110	0.064	0.096	0.054	0.302	0.079	4
	FDC	0.069	0.048	0.085	0.051	0.296	0.051	4

Table 11
Performance metrics for the performance comparison on the complete SHREC'13 Sketch Track Benchmark. "R" denotes the ranking group number.

Contributor	Method	NN	FT	ST	E	DCG	AP	R
Furuya	BF-fDSIFT	0.150	0.099	0.157	0.094	0.353	0.118	3
	BF-fGALIF	0.180	0.101	0.154	0.090	0.354	0.119	3
	BF-fGALIF + BF-fDSIFT	0.211	0.123	0.186	0.106	0.380	0.142	2
	CDMR-BF-fDSIFT	0.217	0.157	0.231	0.135	0.412	0.195	1
	CDMR-BF-fGALIF	0.239	0.175	0.265	0.148	0.428	0.215	1
	CDMR-BF-fGALIF + CDMR-BF-fDSIFT	**0.275**	**0.204**	**0.299**	**0.167**	**0.459**	**0.247**	1
	UMR-BF-fDSIFT	0.158	0.114	0.180	0.106	0.368	0.135	2
	UMR-BF-fGALIF	0.166	0.119	0.182	0.103	0.370	0.134	2
	UMR-BF-fGALIF + UMR-BF-fDSIFT	0.209	0.131	0.199	0.114	0.387	0.152	2
Li	SBR-VC_NUM_100	0.161	0.097	0.149	0.085	0.349	0.113	3
	SBR-VC_NUM_50	0.131	0.082	0.130	0.076	0.332	0.098	3
	SBR-2D-3D_NUM_50	0.133	0.079	0.125	0.074	0.329	0.096	3
Pascoal	HTD	0.018	0.017	0.032	0.018	0.241	0.024	5
Saavedra	HOG-SIL	0.106	0.067	0.105	0.060	0.305	0.083	4
	HELO-SIL	0.106	0.062	0.094	0.053	0.301	0.077	4
	FDC	0.065	0.048	0.085	0.051	0.296	0.062	4

7. Discussions

7.1. Methods

We classify all contributed 15 methods with respect to the different classification methods mentioned in the first paragraph of Section 2.1. Most methods employ local features, except that SBR VC, SBR 2D 3D, FDC and HTD perform global feature matching. Only SBR VC and SBR 2D 3D perform view selection while all the other methods adopt the approach of fixed view sampling. All the 9 methods of the Furuya's group adopt a Bag of Words framework and among them 6 "CDMR" and "UMR" based

Table 12
Timing information comparison on the "Testing" dataset of SHREC'13 Sketch Track Benchmark: t is the average response time (in seconds) per query. "R" denotes the ranking group number. *Note* that for SBR-VC, the number of representative views $N_c = \frac{C}{6} \cdot N_0$.

Contributor (with computer configuration)	Method	Language	t	R
Furuya (CPU: Intel Core i7 3930 K @ 3.20 Hz; GPU: NVIDIA GeForce GTX 670; Memory: 64 GB; OS: Ubuntu 12.04)	BF-fDSIFT	C++, CUDA	1.26	2
	BF-fGALIF	C++	0.49	2
	CDMR-BF-fDSIFT	C++	606.96	4
	CDMR-BF-fGALIF	C++	615.95	4
	UMR-BF-fDSIFT	Matlab	54853.77	5
	UMR-BF-fGALIF	Matlab	27219.49	5
Li (CPU: Intel Core 2 Duo E7500 @ 2.93 GHz; Memory: 16 GB; OS: Windows 7 64-bit)	SBR-VC_NUM_100	C/C++	208.85	4
	SBR-VC_NUM_50	C/C++	68.92	3
	SBR-2D-3D_NUM_50	C/C++	43.93	3
Pascoal (CPU: Intel Core i7-2760QM CPU @ 2.40 GHz (8CPU); Memory: 16 GB; OS: Windows 8)	HTD	Java	0.32	2
Saavedra (CPU: Intel Core i7-3770 CPU @ 3.40 GHz; Memory: 8 GB; OS: Ubuntu 11.10)	HOG-SIL	C++	**0.09**	1
	HELO-SIL	C++	**0.09**	1
	FDC	C++	**0.09**	1

methods utilize a distance metric learning approach [66] (one type of machine learning techniques) to significantly improve the performance of their methods; while the remaining 6 methods contributed by the other 3 groups directly compare global features and do not adopt any machine learning approach.

Based on the fact that all the top algorithms on the two benchmarks are machine learning technique based ones and it contributes much to the best performance obtained, we regard it as a promising approach to advance the performance of existing retrieval algorithms. This can be also found in our performance evaluation results: though SBR VC and SBR 2D 3D are at least comparable to BF fGALIF and BF fDSIFT, CDMR BF fGALIF evidently outperforms either of them after employing the "CDMR " distance learning approach. However, though utilizing a machine learning approach can obviously improve the retrieval performance of a sketch based retrieval algorithm, it may also significantly increase its retrieval time. Therefore, more research and improvements are deserved to solve this contradictory issue in order to apply machine learning techniques into sketch based 3D model retrieval in a scalable manner.

In terms of 2D shape descriptors, shape context, GALIF, DSIFT and HOG (refer to [4,14 16,5,17]) features are relatively more promising in achieving top performance if compared with Fourier descriptors, and structure or topology based ones. We also have found that compared with global shape descriptors based on direct feature matching, it is much easier for local shape descriptors to achieve real time efficiency while combined with a Bag of Words framework.

7.2. Benchmarks

The SHREC'12 Sketch Track Benchmark could be extended by sketch data as currently being compiled by other researchers, making it more representative. Also, controlling the level of standardization with respect to sketch parameters such as sketching quality, style, and level of detail is deemed interesting. The standard query sketches [56] included in this benchmark are a starting point to this direction. While, the SHREC'13 Sketch Track Benchmark could be extended by adding more models to the target 3D model dataset to make each class contain the same number of models, which will remove the remaining bias and make the benchmark more representative.

8. Conclusions and future work

8.1. Overall performance evaluation

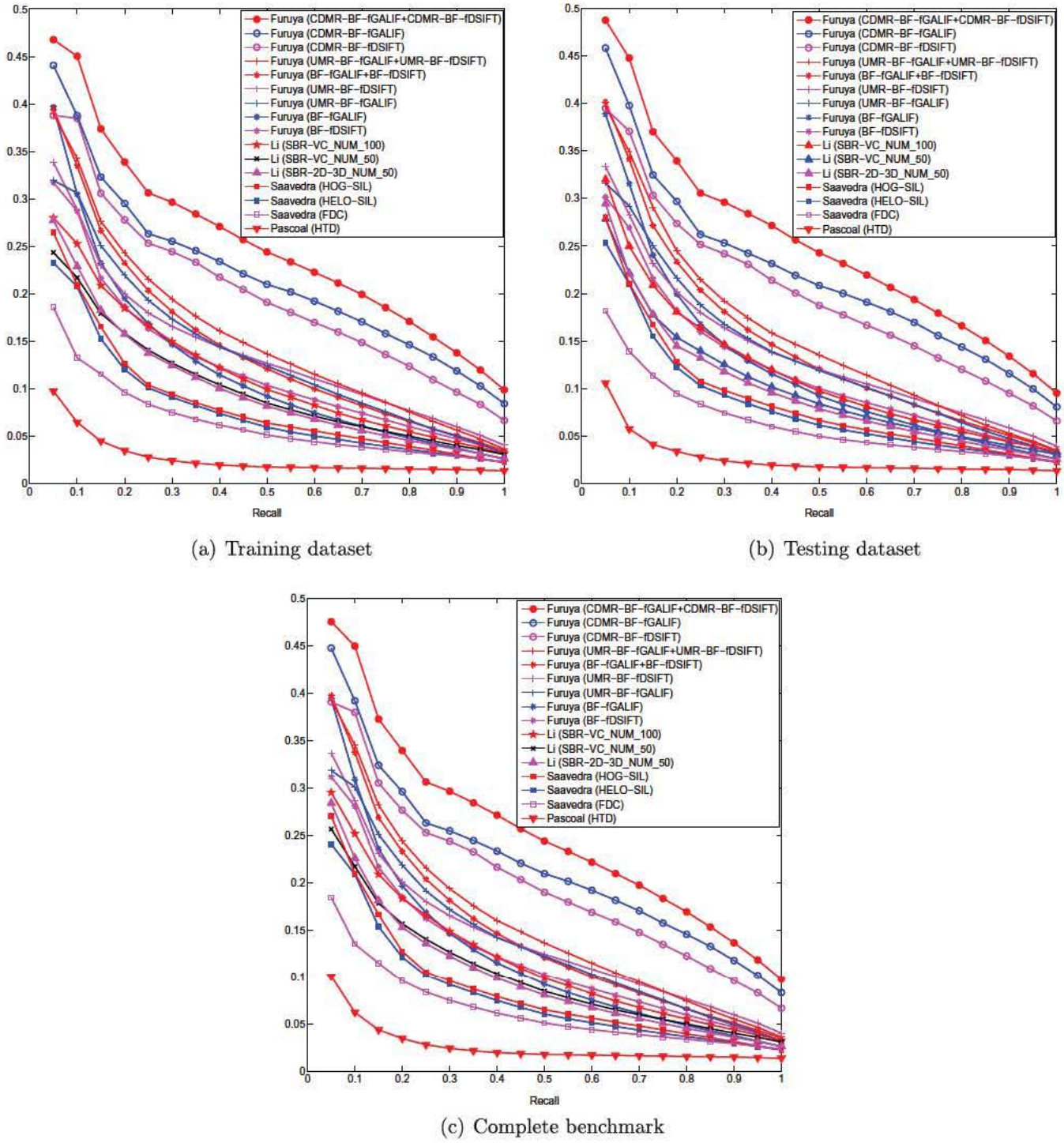
(1) On the small scale benchmark, we performed a comprehensively comparative evaluation of 14 state of the art sketch based retrieval methods in terms of accuracy, scalability and efficiency. Overall, Furuya's CDMR BF fGALIF + CDMR BF fDSIFT and CDMR BF fGALIF methods perform best, followed by the five comparable methods of Furuya's CDMR BF fDSIFT, UMR BF fGALIF + UMR BF fDSIFT and UMR BF fGALIF, as well as Li's SBR VC_NUM_100 and SBR 2D 3D_NUM_100; (2) On the large scale benchmark, we can draw a similar conclusion: Furuya's three CDMR based algorithms (CDMR BF fGALIF + CDMR BF fDSIFT, CDMR BF fGALIF and CDMR BF fGALIF) have the best performance, followed by their three UMR based algorithms (UMR BF fGALIF + UMR BF fDSIFT, UMR BF fDSIFT and UMR BF fGALIF) and their hybrid approach BF fGALIF + BF fDSIFT. Furuya's BF fDSIFT, BF fGALIF and Li's SBR VC and SBR 2D 3D succeed them and they outperform Saavedra's three approaches (HOG SIL, HELO SIL and FDC) which are again better than HTD. However, compared with the case of small scale benchmark, all the performance drops drastically, which indicates both the challenge of the large scale benchmark and the new issues of scalability of existing sketch based retrieval algorithms. This will also be helpful to guide our research on developing new sketch based 3D retrieval algorithms or extending current methods to accustom to the large scale retrieval scenarios.

8.2. Efficiency comparison

(1) On the small scale benchmark, in terms of retrieval speed, we observe large differences between all methods: from 0.02 s (Saavedra's three methods) to 314.82 s (Furuya's UMR BF fDSIFT). In this case, the best performing "CDMR " based methods, together with BF fDSIFT and BF fGALIF are interactive, while other remaining approaches need improvement in this regard. (2) On the large scale benchmark, the difference in the retrieval speed is even larger: from 0.09 s (Saavedra's three methods) to 54853.77 s (Furuya's UMR BF fDSIFT). What is more, we observe that the speed of the top "CDMR " based methods drops drastically (730∼1230 times slower) when scaled to a less than 2.5 times bigger benchmark, thus they can be further improved in this aspect. Saavedra's three methods are still the fastest, followed by Pascoal's HTD; while Li's SBR VC and SBR 2D 3D methods extend their retrieval time proportionally according to the scale of the benchmark.

8.3. Concluding remark

In conclusion, the small scale sketch based retrieval track is the first attempt to include this topic in SHREC in order to foster this challenging and interesting research direction. Even though it is the first time, we already have 5 groups who have successfully



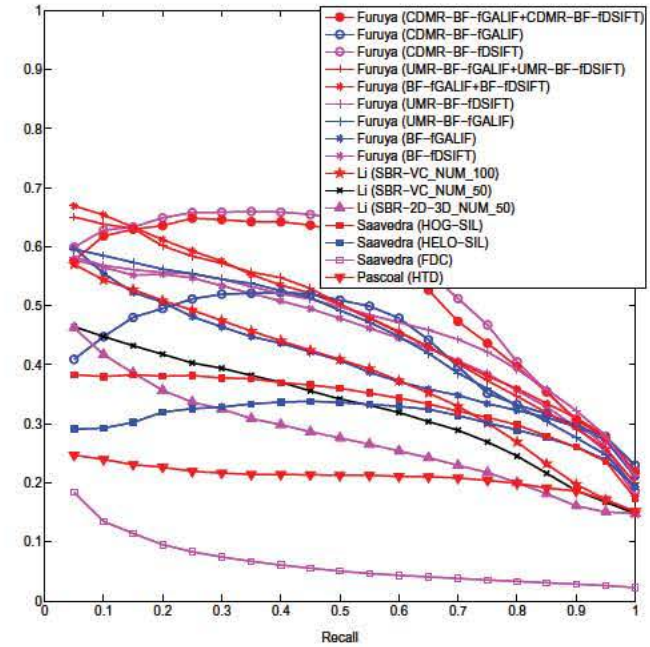
(a) Training dataset (b) Testing dataset

(c) Complete benchmark

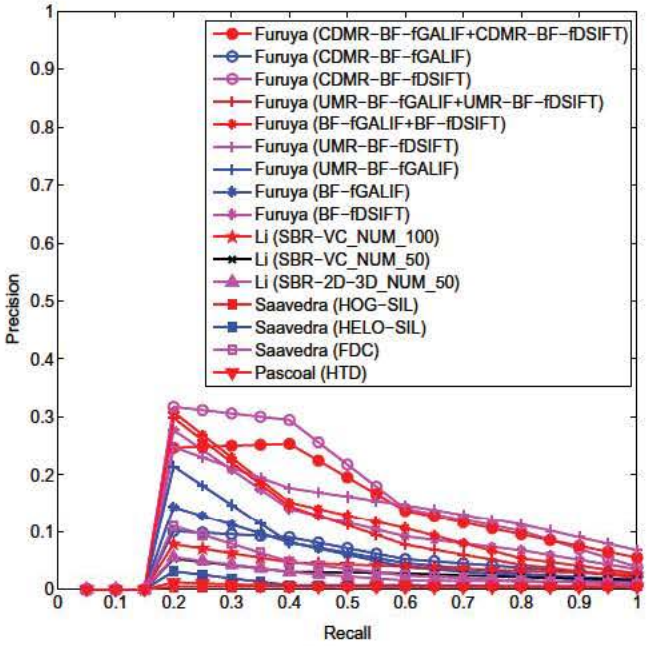
Fig. 18. Precision-Recall diagram performance comparisons on different datasets of the SHREC'13 Sketch Track Benchmark for the 16 runs of the 15 contributed methods.

participated. On the other hand, the large scale sketch based re trieval track is an attempt to further foster this challenging and interesting research direction encouraged by the success of SHREC'12 Sketch based 3D shape retrieval track. Though the benchmark is very challenging, we still have 3 groups who have successfully participated in the track and they have contributed 5 runs of 4 methods. Through these two tracks, we provided two common platforms (the two benchmarks) to solicit current sketch based 3D model retrieval approaches in terms of both small scale and large scale retrieval scenarios. This helps us iden tify state of the art methods as well as future research directions for this research area. We also hope that the small scale and large scale sketch retrieval benchmarks together with the evalua tion code will become useful references for researchers in this community.

To comprehensively evaluate the participating methods in the two tracks and other sketch based 3D model retrieval algorithms, solicit the state of the art approaches, and provide a more com plete reference for the researchers in this community, we invite re lated authors to run their methods on both benchmarks. Finally, a detailed comparative evaluation has been accomplished based on 17 contributed runs of 15 (4 top participating algorithms and 11



(a) airplane



(b) ant

Fig. 19. Precision-Recall diagram performance comparisons of the "airplane" and "ant" classes on the complete SHREC'13 Sketch Track Benchmark for the 15 contributed methods.

additional state of the art approaches) best retrieval methods on the two benchmarks. This evaluation work helps us to identify the state of the art approaches in terms of accuracy, scalability and efficiency, existing problems and current challenges, as well as promising research topics and techniques.

8.4. Future work

This evaluation work helps us to identify the state of the art approaches in terms of accuracy, scalability and efficiency, existing problems and current challenges, as well as promising research topics and techniques. Therefore, we identify the fu ture direction of this research area is developing efficient algo rithms which are scalable to different sizes and types of sketch queries and models. It can be achieved in the following eight as pects.

- **Scalable and interactive retrieval algorithms.** In our evalua tion, compared with the case of the small scale benchmark, the performance of all the methods drops drastically on the large scale benchmark, which evidently raises the issue of sca lability of existing sketch based retrieval algorithms. On the other hand, for some best performing approaches their retrieval time increases too fast on a larger dataset, which creates a hur dle for their applications. It is well known that highly interac tive response times are a key requirement for interactive sketch based retrieval applications such as shadow drawing for interactive retrieval [79], or modeling by sketched example. Scalability of query processing time with respect to large target database and descriptor sizes is an important issue yet to be addressed. One possibility is to take advantage of already devel oped high dimensional or metric index structures [80,81], which can accelerate the typically many nearest neighbor com putations required. However, this implies restricting the descriptors to vector spaces and metric distance functions. In case that more complex descriptors or distance functions are required for the retrieval task, developing ad hoc index struc tures may be required to efficiently fulfill this task. Also, we see bag of words approaches in conjunction with inverted indi ces as promising to provide scalable answer times.
- **Building large-scale benchmarks.** Since scalability is so impor tant an issue, we should create a large scale sketch based 3D retrieval benchmark, in terms of both 2D sketches and 3D mod els, to evaluate the scalability property of sketch based retrieval methods.
- **Interdisciplinary research directions.** We notice that generally the retrieval performance, either in terms of accuracy or effi ciency, is far from satisfactory and the performance of existing sketch based retrieval algorithms drops apparently when scaled to a large collection. To improve the retrieval accuracy, we recommend utilizing knowledge and techniques from other related disciplines, such as pattern recognition, machine learn ing, and computer vision. For example, to increase the accuracy and efficiency in sketch based retrieval, we can perform sketch recognition first and use the result to prioritize the comparison between a query sketch and 3D models in the database. During the evaluation, we have found that all the top algorithms on the two benchmarks utilize machine learning technique, which contributes much to the best performance obtained. Therefore, machine learning is another promising interdisciplinary approach to further improve the performance of existing retrie val algorithms. However, we need to pay more attention to the scalability properties of the machine learning techniques employed and make sure they or their variations can meet the real time requirements of the retrieval applications.
- **Query adaptive sketch-based retrieval algorithms.** On the effectiveness side, another promising direction for future work is to consider adaptive sketch based retrieval algorithms. Given that different types (or modes) of user sketches can occur (such as perspective, orthogonal, and abstract versus realistic), differ ent descriptor types may be best suited for the search, depend ing on the sketch type. Adaptive retrieval systems could improve the search by deciding an appropriate descriptor which best supports the type of sketch query. Sketch type dependent evaluation benchmarking is needed to understand the possible relationships between sketch types and method performance.

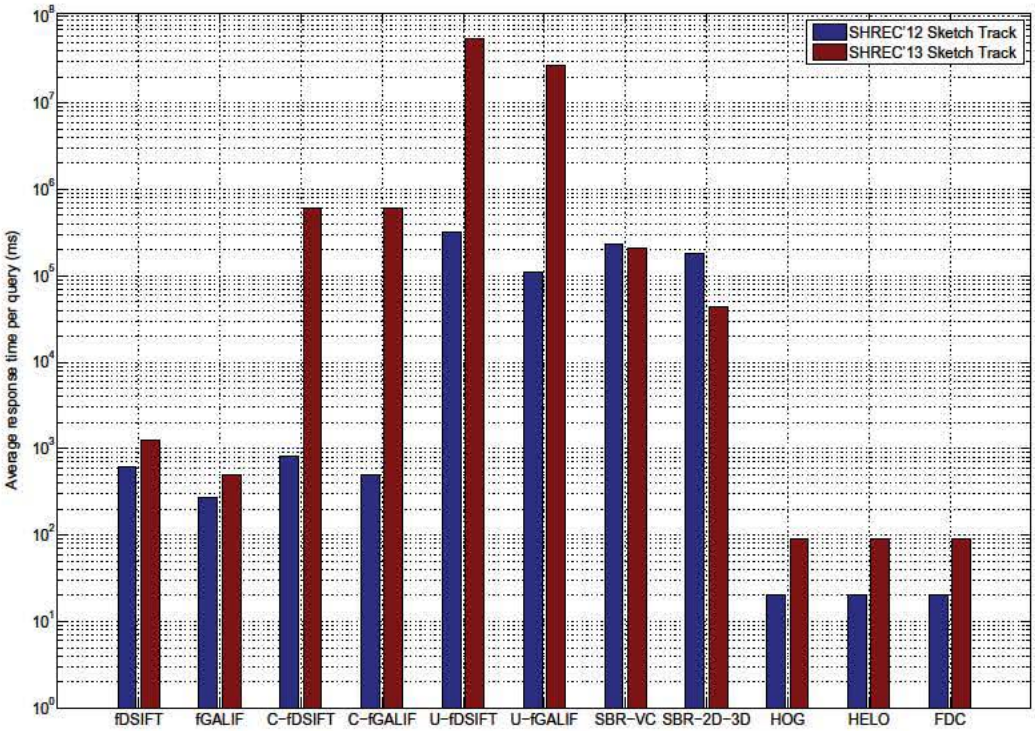


Fig. 20. Timing information comparison of the contributed algorithms based on the small-scale SHREC'12 Sketch Track Benchmark ("Hand-drawn and Extended" dataset, 400 models) and the large-scale SHREC'13 Sketch Track Benchmark ("Testing" dataset, 1258 models). Logarithmic scaling is applied on the Time axis. Methods fDSIFT, fGALIF, C-fDSIFT, C-fGALIF, U-fDSIFT, and U-fGALIF denote BF-fDSIFT, BF-fGALIF, CDMR-BF-fDSIFT, CDMR-BF-fGALIF, UMR-BF-fDSIFT, and UMR-BF-fGALIF, respectively. Note that for SBR-VC, NUM is set to 100 for both SHREC'12 and SHREC'13 Tracks, while different number of representative views have been adopted ($N_c = \frac{C}{2} \cdot N_0$ and $N_c = \frac{C}{6} \cdot N_0$ for SHREC'12 Sketch Track and SHREC'13 Sketch Track, respectively, see Section 5.3). Also note that different NUM values are used for SBR-2D-3D (NUM = 100 for SHREC'12 Sketch Track and NUM = 50 for SHREC'13 Sketch Track).

- **Semantics-driven sketch-based retrieval and search intention study.** To bridge the semantic gap between 2D sketches and 3D models, a promising research direction is to develop algorithms and benchmarks that deal more directly with semantics (e.g., semantic categories) and search intention (e.g., per session specification user intention, specifically, for example, by relevance feedback).
- **Developing new local shape descriptors.** According to the evaluation, we have found that most efficient retrieval algorithms adopt local shape descriptors. On the other hand, the retrieval algorithms based on HOG, GALIF, and DSIFT local shape descriptors, or the shape context global shape descriptor, outperform those based on Fourier descriptors, or structure or topology based descriptors, in terms of retrieval accuracy. Therefore, developing novel local shape descriptors is a relatively more promising research direction to meet the requirements of future applications, which require high efficiency and accuracy.
- **Domain specific sketch based retrieval algorithms and their evaluation methods.** We can extend sketch based retrieval for retrieving objects in images such as photographs and paintings; or special 3D objects like clothes and protein molecules. As such, it is necessary to build an appropriate image and 3D model database for benchmarking. New evaluation metrics may be also needed for the specific domains.
- **Scene sketch and partial sketch-based 3D retrieval.** Generally, existing sketch based 3D model retrieval algorithms assume there is only one object in the query sketch, and do not consider the case of a 2D scene sketch query containing several objects, which may overlap each other and also have their spatial context information. Therefore, sketch based 3D model retrieval in the context of a 2D scene sketch deserves our further exploration. Finally, we note that our benchmarks address, in spirit, the global retrieval model. Future benchmarking for sketch based retrieval may in particular address partial sketch based 3D retrieval algorithms which target interactively establishing correspondence between a partial sketch of an object and 3D model parts.

Acknowledgments

This work has been supported by the Army Research Office grant W911NF 12 1 0057, Texas State University Research Enhancement Program (REP), and NSF CRI 1305302 to Yijuan Lu, as well as the Shape Metrology IMS to Afzal Godil.

The work of Benjamin Bustos has been funded by Fondecyt (Chile) Project 1110111.

The work of Pedro B. Pascoal, Alfredo Ferreira, Manuel J. Fonseca reported in this paper has been supported by national funds through FCT under contract Pest OE/EEI/LA0021/2013.

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 Mathias Eitz (TU Berlin, Germany), James Hays (Brown University, USA) and Marc Alexa (TU Berlin, Germany) who built the large collection of sketches.

We would also like to thank Sang Min Yoon (Yonsei University, Korea), Maximilian Scherer (TU Darmstadt, Germany), Tobias Schreck (University of Konstanz) and Arjan Kuijper (Fraunhofer IGD) who collected the TU Darmstadt and Fraunhofer IGD sketch data.

We would also like to thank Daniela Giorgi, Silvia Biasotti, Laura Paraboschi (CNR IMATI, Italy) who built the Watertight Shape Benchmark for SHREC 2007 and Snograss and Vanderwart who built the standard line drawings dataset, as well as Philip Shilane, Patrick Min, Michael M. Kazhdan, Thomas A. Funkhouser (Prince ton University, USA) who built the Princeton Shape Benchmark.

References

[1] <http://www.itl.nist.gov/iad/vug/sharp/contest/2012/SBR/>, 2013.
[2] <http://www.itl.nist.gov/iad/vug/sharp/contest/2013/SBR/>, 2013.
[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. 189–194.
[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, 2012, pp. 109–118.
[5] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, M. Alexa, Sketch-based shape retrieval, ACM Trans. Graph. 31 (2012) 31:1–31:10.
[6] P. Shilane, P. Min, M.M. Kazhdan, T.A. Funkhouser, The Princeton shape benchmark, in: Shape Modeling International, IEEE Computer Society, 2004, pp. 167–178.
[7] 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, 2013, pp. 89–96.
[8] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, D. Jacobs, A search engine for 3D models, ACM Trans. Graph. 22 (2003) 83–105.
[9] M.M. Kazhdan, T.A. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3D shape descriptors, in: L. Kobbelt, P. Schröder, H. Hoppe (Eds.), Symposium on Geometry Processing, ACM International Conference Proceeding Series, vol. 43, Eurographics Association, 2003, pp. 156–164.
[10] S.M. Yoon, M. Scherer, T. Schreck, A. Kuijper, Sketch-based 3D model retrieval using diffusion tensor fields of suggestive contours, in: A.D. Bimbo, S.-F. Chang, A.W.M. Smeulders (Eds.), ACM Multimedia, ACM, 2010, pp. 193–200.
[11] J.M. Saavedra, B. Bustos, M. Scherer, T. Schreck, STELA: sketch-based 3D model retrieval using a structure-based local approach, in: F.G.B.D. Natale, A.D. Bimbo, A. Hanjalic, B.S. Manjunath, S. Satoh (Eds.), ICMR, ACM, 2011, pp. 26:1–26:8.
[12] D. Doug, F. Adam, R. Szymon, S. Anthony, Suggestive contours for conveying shape, ACM Trans. Graph. 22 (2003) 848–855.
[13] M. Aono, H. Iwabuchi, 3D shape retrieval from a 2D image as query, in: Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC) 2012, 2012, pp. 1–10.
[14] M. Eitz, K. Hildebrand, T. Boubekeur, M. Alexa, Sketch-based 3D shape retrieval, in: SIGGRAPH Talks, ACM, 2010, p. 1.
[15] M. Eitz, K. Hildebrand, T. Boubekeur, M. Alexa, An evaluation of descriptors for large-scale image retrieval from sketched feature lines, Comput. Graph. 34 (2010) 482–498.
[16] M. Eitz, K. Hildebrand, T. Boubekeur, M. Alexa, Sketch-based image retrieval: benchmark and bag-of-features descriptors, IEEE Trans. Visual. Comput. Graph. 17 (2011) 1624–1636.
[17] M. Eitz, J. Hays, M. Alexa, How do humans sketch objects?, ACM Trans Graph. 31 (2012) 44:1–44:10.
[18] T. Shao, W. Xu, K. Yin, J. Wang, K. Zhou, B. Guo, Discriminative sketch-based 3D model retrieval via robust shape matching, Comput. Graph. Forum 30 (2011) 2011–2020.
[19] B. Li, H. Johan, Sketch-based 3D model retrieval by incorporating 2D–3D alignment, Multimedia Tools Appl. 65 (2013) 363–385.
[20] B. Li, H. Johan, View context: a 3D model feature for retrieval, in: S. Boll et al. (Eds.), MMM 2010, LNCS, vol. 5916, Springer, Heidelberg, 2010, pp. 185–195.
[21] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 509–522.
[22] B. Li, Y. Lu, R. Fares, Semantic sketch-based 3D model retrieval, in: ICME, IEEE, 2013, pp. 1–4.
[23] T. Xue, J. Liu, X. Tang, Example-based 3D object reconstruction from line drawings, in: CVPR, IEEE, 2012, pp. 302–309.
[24] T. Xue, J. Liu, X. Tang, Object cut: complex 3D object reconstruction through line drawing separation, in: The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, 2010, pp. 1149–1156.
[25] J. Liu, Y. Chen, X. Tang, Decomposition of complex line drawings with hidden lines for 3D planar-faced manifold object reconstruction, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011) 3–15.
[26] X. Xie, K. Xu, N.J. Mitra, D. Cohen-Or, B. Chen, Sketch-to-design: context-based part assembly, Comput. Graph. Forum 32 (2013) 233–245.
[27] K. Xu, K. Chen, H. Fu, W.-L. Sun, S.-M. Hu, Sketch2scene: sketch-based co-retrieval and co-placement of 3D models, ACM Trans. Graph. 32 (2013) 123:1–123:15.
[28] S. Kanai, Content-based 3D mesh model retrieval from hand-written sketch, Int. J. Interact. Des. Manuf. 2 (2008) 87–98.
[29] A. Tatsuma, M. Aono, Multi-Fourier spectra descriptor and augmentation with spectral clustering for 3D shape retrieval, Vis. Comput. 25 (2009) 785–804.
[30] S.M. Yoon, A. Kuijper, View-based 3D model retrieval using compressive sensing based classification, in: The 7th International Symposium on Image and Signal Processing and Analysis, ISPA 2011, IEEE, 2011, pp. 437–442.
[31] J.M. Saavedra, B. Bustos, T. Schreck, S.M. Yoon, M. Scherer, Sketch-based 3D model retrieval using keyshapes for global and local representation, in: Eurographics Workshop on 3D Object Retrieval (3DOR), 2012, 2012, pp. 47–50.
[32] T. Judd, F. Durand, E.H. Adelson, Apparent ridges for line drawing, ACM Trans. Graph. 26 (2007) 19.
[33] Y. Ohtake, A.G. Belyaev, H.-P. Seidel, Ridge-valley lines on meshes via implicit surface fitting, ACM Trans. Graph. 23 (2004) 609–612.
[34] X. Xie, Y. He, F. Tian, H.S. Seah, X. Gu, H. Qin, An effective illustrative visualization framework based on photic extremum lines (PELs), IEEE Trans. Vis. Comput. Graph. 13 (2007) 1328–1335.
[35] L. Zhang, Y. He, H.S. Seah, Real-time computation of photic extremum lines (PELs), The Visual Comput. 26 (2010) 399–407.
[36] M. Kolomenkin, I. Shimshoni, A. Tal, Demarcating curves for shape illustration, ACM Trans. Graph. 27 (2008) 157.
[37] Y. Miao, J. Feng, Perceptual-saliency extremum lines for 3D shape illustration, The Visual Comput. 26 (2010) 433–443.
[38] L. Zhang, Y. He, J. Xia, X. Xie, W. Chen, Real-time shape illustration using Laplacian lines, IEEE Trans. Vis. Comput. Graph. 17 (2011) 993–1006.
[39] L. Zhang, J. Xia, X. Ying, Y. He, W. Mueller-Wittig, H.S. Seah, Efficient and robust 3D line drawings using difference-of-gaussian, Graph. Models 74 (2012) 87–98.
[40] M. Kolomenkin, I. Shimshoni, A. Tal, Multi-scale curve detection on surfaces, in: CVPR, IEEE, 2013, pp. 1–8.
[41] D. DeCarlo, Depicting 3D shape using lines, Proc. SPIE 8291 (2012). 829116–829116–16..
[42] D. Zhang, G. Lu, A comparative study of Fourier descriptors for shape representation and retrieval, in: Proceedings of the Fifth Asian Conference on Computer Vision (ACCV02), 2002, pp. 646–651.
[43] A. Sajjanhar, G. Lu, D. Zhang, Image retrieval using modified generic fourier descriptors, in: B. Gupta (Ed.), Computers and their Applications, ISCA, 2004, pp. 32–35.
[44] D. Zhang, G. Lu, Shape-based image retrieval using generic Fourier descriptor, Signal Process. Image Commun. (2002) 825–848.
[45] A. Khotanzad, Y. Hong, Invariant image recognition by Zernike moments, IEEE Trans. Pattern Anal. Mach. Intell. 12 (1990) 489–497.
[46] M. Novotni, R. Klein, 3D Zernike descriptors for content based shape retrieval, in: Proceedings of the 8th ACM Symposium on Solid Modeling and Applications, ACM Press, 2003, pp. 216–225.
[47] J. Revaud, G. Lavoué, A. Baskurt, Improving Zernike moments comparison for optimal similarity and rotation angle retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 627–636.
[48] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, Pattern Recog. 29 (1996) 51–59.
[49] M. Eitz, K. Hildebrand, T. Boubekeur, M. Alexa, A descriptor for large scale image retrieval based on sketched feature lines, in: C. Grimm, J.J. L Jr., (Eds.), SBM, Eurographics Association, 2009, pp. 29–36.
[50] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vision 60 (2004) 91–110.
[51] H. Jegou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, 2010, pp. 3304–3311.
[52] T. Furuya, R. Ohbuchi, Dense sampling and fast encoding for 3D model retrieval using bag-of-visual features, in: S. Marchand-Maillet, Y. Kompatsiaris (Eds.), CIVR, ACM, 2009, pp. 17:1–8.
[53] R. Ohbuchi, T. Furuya, Scale-weighted dense bag of visual features for 3D model retrieval from a partial view 3D model, in: S. Marchand-Maillet, Y. Kompatsiaris (Eds.), IEEE ICCV 2009 Workshop on Search in 3D and Video (S3DV), ACM, 2009, pp. 63–70.
[54] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), IEEE Computer Society, 2005, pp. 886–893.
[55] E. Candès, D. Donoho, Curvelets: A surprisingly effective nonadaptive representation of objects with edges, in: L.L. S et al. (Ed.), Curves and Surfaces, Vanderbilt University Press, Nashville, TN, 1999, pp. 1–16.
[56] J.G. Snodgrass, M. Vanderwart, A standardized set of 260 pictures: norms for name agreement, image agreement, familiarity, and visual complexity, J. Experimen. Pyschol.: Human Learn. Mem. 6 (1980) 174–215.
[57] F.-X. Alario, L. Ferrand, A set of 400 pictures standardized for French: Norms for name agreement, image agreement, familiarity, visual complexity, image variability, and age of acquisition, Behav. Res. Meth. Instrum., Comput. 31 (1999) 531–552.
[58] P. Bonin, R. Peereman, N. Malardier, A. Mot, M. Chalard, A new set of 299 pictures for psycholinguistic studies: French norms for name agreement, image agreement, conceptual familiarity, visual complexity, image variability, age of acquisition, and naming latencies, Behav. Res. Meth. Instrum., Comput. 35 (2003) 158–167.
[59] M. Sanfeliu, A. Fernandez, A set of 254 Snodgrass–vanderwart pictures standardized for spanish: norms for name agreement, image agreement, familiarity, and visual complexity, Behav. Res. Meth. Instrum., Comput. 28 (1996) 537–555.
[60] S. Pompia, M. Miranda, O. Bueno, A set of 400 pictures standardised for portuguese: norms for name agreement, familiarity and visual complexity for children and adults, Arquivos de Neuro-Psiquiatria 59 (2001) 330–337.

[61] F. Cole, A. Golovinskiy, A. Limpaecher, H.S. Barros, A. Finkelstein, T.A. Funkhouser, S. Rusinkiewicz, Where do people draw lines?, ACM Trans Graph. 27 (2008) 1–11.
[62] J.M. Saavedra, B. Bustos, An improved histogram of edge local orientations for sketch-based image retrieval, in: M. Goesele, S. Roth, A. Kuijper, B. Schiele, K. Schindler (Eds.), DAGM-Symposium, Lecture Notes in Computer Science, vol. 6376, Springer, 2010, pp. 432–441.
[63] 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.
[64] B. Li, H. Johan, 3D model retrieval using hybrid features and class information, Multimedia Tools Appl. 62 (2013) 821–846.
[65] T. Furuya, R. Ohbuchi, Ranking on cross-domain manifold for sketch-based 3D model retrieval, in: Cyberworlds 2013, 2013, pp. 1–8.
[66] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, Advances in Neural Information Processing Systems, vol. 16, MIT Press, 2004, pp. 321–328.
[67] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Mach. Learn. 63 (2006) 3–42.
[68] B. Li, Y. Lu, H. Johan, Sketch-based 3D model retrieval by viewpoint entropy-based adaptive view clustering, in: U. Castellani, T. Schreck, S. Biasotti, I. Pratikakis, A. Godil, R.C. Veltkamp (Eds.), 3DOR, Eurographics Association, 2013, pp. 49–56.
[69] M. Fonseca, A. Ferreira, J. Jorge, Sketch-based retrieval of vector drawings, in: Sketch-based Interfaces and Modeling, 2011, 2011, pp. 181–201.
[70] P.M.A. Sousa, M.J. Fonseca, Sketch-based retrieval of drawings using spatial proximity, J. Vis. Lang. Comput. 21 (2010) 69–80.
[71] Y.-T. Shen, D.-Y. Chen, X.-P. Tian, M. Ouhyoung, 3D model search engine based on Lightfield descriptors, in: Eurographics, 2003, 2003, pp. 1–7.
[72] J.L. Bentley, T. Ottmann, Algorithms for reporting and counting geometric intersections, IEEE Trans. Comput. 28 (1979) 643–647.
[73] M.M. Syslo, An efficient cycle vector space algorithm for listing all cycles of a planar graph, SIAM J. Comput. 10 (1981) 797–808.
[74] J.D. Horton, A polynomial-time algorithm to find the shortest cycle basis of a graph, SIAM J. Comput. 16 (1987) 358–366.
[75] M.J. Egenhofer, K.K. Al-Taha, Reasoning about gradual changes of topological relationships, in: A.U. Frank, I. Campari, U. Formentini (Eds.), Spatio-Temporal Reasoning, Lecture Notes in Computer Science, vol. 639, Springer, 1992, pp. 196–219.
[76] M.J. Fonseca, C. Pimentel, J.A. Jorge, CALI: An online scribble recognizer for calligraphic interfaces, in: Proceedings of the 2002 AAAI Spring Symposium Sketch Understanding, 2002, pp. 51–58.
[77] D. Zhang, G. Luo, An integrated approach to shape based image retrieval, in: Proceedings of the 5th Asian Conference on Computer Vision (ACCV 2002), 2002, 2002, pp. 652–657.
[78] M. Muja, D.G. Lowe, Fast approximate nearest neighbors with automatic algorithm configuration, in: Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP), 2009, 2009, pp. 331–340.
[79] Y.J. Lee, C.L. Zitnick, M.F. Cohen, Shadowdraw: real-time user guidance for freehand drawing, ACM Trans. Graph. 30 (2011) 27:1–27:10.
[80] H. Samet, Foundations of Multidimensional and Metric Data Structures, Morgan Kaufman Publishers Inc., San Francisco, CA, USA, 2005.
[81] P. Zezula, G. Amato, V. Dohnal, M. Batko, Similarity search: the metric space approach, Advances in Database Systems, vol. 32, Springer, 2006.