Eurographics Workshop on 3D Object Retrieval (2016)
Alfredo Ferreira, Andrea Giachetti, Daniela Giorgi, Remco C. Veltkamp (Editors)

SHREC'16 Track: 3D Sketch-Based 3D Shape Retrieval

Bo Li[†‡1], Yijuan Lu[†‡2], Fuqing Duan[‡3], Shuilong Dong[‡4], Yachun Fan[‡3], Lu Qian[‡3], Hamid Laga[‡5], Haisheng Li[‡4], Yuxiang Li[‡6], Peng Liu[‡4], Maks Ovsjanikov[‡6], Hedi Tabia[‡7], Yuxiang Ye[‡4], Huanpu Yin[‡4], Ziyu Xue[‡4]

[1] Department of Mathematics and Computer Science, University of Central Missouri, Warrensburg, USA
[2] Department of Computer Science, Texas State University, San Marcos, USA
[3] Department of Computer Information and Technology, Beijing Normal University, Beijing, China
[4] School of Computer and Information Engineering, Beijing Technology and Business University, Beijing, China
[5] School of Engineering and IT, Murdoch University, Australia [6] LIX, École Polytechnique, France
[7] ENSEA, ETIS/ENSEA, University of Cergy-Pontoise, CNRS, UMR 8051, France

Abstract

Sketch-based 3D shape retrieval has unique representation availability of the queries and vast applications. Therefore, it has received more and more attentions in the research community of content-based 3D object retrieval. However, sketch-based 3D shape retrieval is a challenging research topic due to the semantic gap existing between the inaccurate representation of sketches and accurate representation of 3D models. In order to enrich and advance the study of sketch-based 3D shape retrieval, we initialize the research on 3D sketch-based 3D model retrieval and collect a 3D sketch dataset based on a developed 3D sketching interface which facilitates us to draw 3D sketches in the air while standing in front of a Microsoft Kinect.
The objective of this track is to evaluate the performance of different 3D sketch-based 3D model retrieval algorithms using the hand-drawn 3D sketch query dataset and a generic 3D model target dataset. The benchmark contains 300 sketches that are evenly divided into 30 classes, as well as 1258 3D models that are classified into 90 classes. In this track, nine runs have been submitted by five groups and their retrieval performance has been evaluated using seven commonly used retrieval performance metrics. We wish this benchmark, the comparative evaluation results and the corresponding evaluation code will further promote sketch-based 3D shape retrieval and its applications.

Categories and Subject Descriptors (according to ACM CCS): H.3.3 [Computer Graphics]: Information Systems—Information Search and Retrieval

1. Introduction

Sketch-based 3D model retrieval is to retrieve relevant 3D models using sketch(es) as input. This scheme is intuitive and convenient for users to learn and search for 3D models. It is also popular and important for related applications such as sketch-based 3D modeling and recognition.

However, existing sketch-based 3D model retrieval systems are mainly based on 2D sketch queries which contain limited 3D information of the 3D shapes they are supposed to represent. What's more, there is a semantic gap between the iconic representation of 2D sketches and the accurate 3D coordinate representation of 3D models. This makes the task of retrieval using sketch queries much more challenging than those using 3D model queries.

Motivated by the above obstacles, an interesting question has been raised: "why not 3D sketches?: A 3D sketch may provide a better description for an object than a 2D sketch, which not only encodes 3D information (such as depth and features of more facets) of objects, but also contains the salient 3D feature lines of its counterpart of 3D models.

The popularity of low-cost depth cameras like Microsoft's

† Track organizers. For any questions related to the track, please contact li.bo.ntu0@gmail.com.
‡ Track participants.

Kinect makes 3D sketching in a virtual 3D space no longer a dream. Kinect facilitates us to track the 3D locations of 20 joints of a human body. Therefore, a Kinect sensor can be used to track the 3D locations of a user's hand to create a 3D sketch.

In 2015, a Kinect-based 3D sketching system [LLG*15a, LLG*15b] was developed to allow a user to use his/her hand as a drawing tool to draw a 3D sketch. A voice-activated Graphical User Interface (GUI) is designed to facilitate 3D sketching. Based on the Kinect-based 3D sketching system, we have collected a **Kinect300** 3D sketch dataset, which comprises 300 sketches of 30 classes, each with 10 models, from 17 users (4 females and 13 males) in computer science or mathematics related majors. The average age of all the 17 users is 21, and only two males have art experiences.

Based on this new benchmark, we organized this track to foster this challenging research area of sketch-based 3D model retrieval by soliciting retrieval results from current State-of-The-Art 3D model retrieval methods for comparison, especially in terms of scalability to 3D sketch queries. We also provided corresponding evaluation code for computing a set of performance metrics similar to those used in the Query-by-Model retrieval technique.

2. Data Collection

Our 3D sketch-based 3D model retrieval benchmark is motivated by a 3D sketch collection built by Li and Lu et al. [LLG*15a, LLG*15b] and SHREC'13 Sketch Track Benchmark (**SHREC13STB**) [LLG*13].

To explore how to draw 3D sketches in a 3D space and how to use a hand-drawn 3D sketch to search similar 3D models, Li and Lu et al. [LLG*15a, LLG*15b] collected 300 human-drawn 3D sketches of 30 classes, each with 10 sketches by utilizing a Kinect-based virtual 3D drawing system. It avoids the bias issue since they collected the same number of sketches for every class, while the sketch variation within one class is significant.

To facilitate learning-based retrieval, we randomly select 7 sketches from each class for training and use the remained 3 sketches per class for testing, while all the target models as a whole are remained as the target dataset. Participants need to submit results on the training and testing datasets, respectively, if they use learning in their approach(es). Otherwise, only the retrieval results based on the complete query dataset are needed. To provide a complete reference for the future users of our benchmark, we evaluate the participating algorithms on both the testing dataset (7 sketches per class, totally 210 3D sketches) and the complete benchmark (10 sketches per class, 300 sketches).

2.1. 3D Sketch Dataset

The 3D sketch query set comprises 300 3D sketches (30 classes, each with 10 sketches), while 21 classes have relevant models in the target 3D dataset of the SHREC'13 Sketch-Based Retrieval benchmark. Therefore, during the evaluation process, we only consider the performance of the 210 3D sketch queries that have relevant 3D models in the target dataset. One 3D sketch example for each of the 30classes is demonstrated in Fig. 1.

Figure 1: *Example 3D sketches (one example per class, shown in one view) of our **Kinect300** dataset [LLG*15a].*

2.2. 3D Model Dataset

The 3D benchmark dataset is built on the SHREC'13 Sketch Track Benchmark (**SHREC13STB**). Totally, 1258 models of 90 classes are selected to form the target 3D model dataset. We use this dataset as our target 3D model dataset. Some examples are shown in Fig. 2.

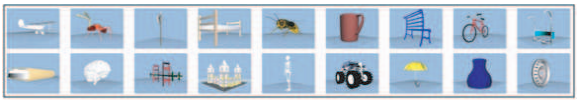


Figure 2: *Example 3D models in the **SHREC13STB** benchmark.*

2.3. Evaluation Method

To have a comprehensive evaluation of the retrieval algorithm, we employ seven commonly adopted performance metrics in the 3D model retrieval community. They are Precision-Recall (PR) diagram, Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), E-Measures (E), Discounted Cumulated Gain (DCG) and Average Precision (AP). We also have developed the code to compute them.

3. Participants

Five groups have participated in the SHREC'16 track on 3D Sketch-Based 3D Shape Retrieval. Nine (9) rank list results (runs) for five (5) different methods developed by five (5) groups have been submitted. The participants and their runs are listed as follows:

- *LSFMR* submitted by Yachun Fan, Fuqing Duan and Lu Qian from Beijing Normal University, Beijing, China (Section 4.1)
- *CNN-Point* and *CNN-Edge* submitted by Yuxiang Li and Maks Ovsjanikov from Ecole Polytechnique, France (Section 4.2)
- *HOD1-4*, *HOD64-1*, *HOD64-2*, and *HOD64-4* submitted by Hedi Tabia from ENSEA and the University of Cergy-Pontoise, France; and Hamid Laga from Murdoch University, Australia (Section 4.3)
- *CNN-SBR* submitted by Yuxiang Ye, Yijuan Lu and Bo Li from Texas State University, USA (Section 4.4)
- *CNN-Maxout-Siamese* submitted by Huanpu Yin, Shuilong Dong, Peng Liu, Ziyu Xue, and Haisheng Li from Beijing Technology and Business University (Section 4.5)

4. Methods

In this section, each participating approach is illustrated in detail.

4.1. Localized Statistical Feature and Manifold Ranking, by Y. Fan, F. Duan and L. Qian

This approach is based on the Bag of Feature (BoF) paradigm. Figure 3 illustrates the main steps of the approach. Three parts of preprocessing, online retrieval and manifold ranking are included in the approach.

Before visual vocabulary training, SVM is applied to remove the noise points in 3D sketches and PCA-based alignment is applied to normalize 3D models and 3D sketches. The local features of the 3D sketch training data are clustered by k-means method. A visual dictionary is built after clustering these feature descriptors. In this approach, the number of the visual vocabulary is 1024.

In this approach, a 3D sketch is modeled as a collection of surface points. Similarly, the 3D models are sampled by a collection of points. The point sets of a 3D model are generated by referencing to the NPR method in [Her10]. The occluding contour points and boundary points of the 3D model are calculated. The occluding contour points are the points at which the normals are perpendicular to the viewing direction. The boundary points are the points at which no two faces share one edge.

Feature quantification is used to calculate the distribution of occurrence of code words based on different visual vocabularies for 3D models or 3D sketches. The indexing of

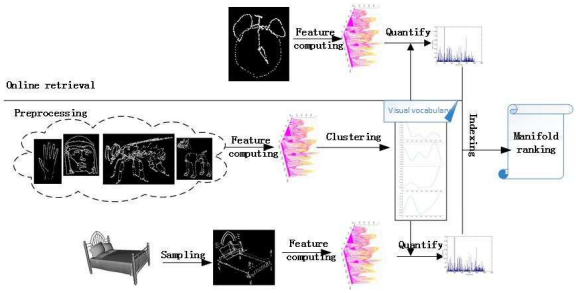


Figure 3: *Main steps of the approach.*

all the 3D features applies visual vocabulary as the prime index, while the weight of the visual vocabulary in the 3D model is used as the secondary index. All the visual vocabulary weights of the 3D models are ranked and stored.

4.1.1. Localized statistical feature

In this approach, a new local feature vector named Localized Statistical Feature (LSF) is proposed. This feature vector describes the local region shape as a point statistical result. The local region comes with a dense grid division. The method of dense subdivision increases the retrieval performance for either local features or global features.

In order to statistically describe the 3D point distribution in a local region, each local region is divided into some smaller sub-regions. Suppose that the local region is a box, the sub-region is obtained by using a bisection method applied on each axis. The sub-region is called cell. The local region is divided into $L \times L \times L$ cells.

The combined each cell feature value forms the local feature vector LSF. For each cell, the feature value is the number of points in the cell. All cells accumulate a local 1-D vector as the local region feature representation.

Because the number of points in a 3D model or a 3D sketch is not a fixed value, for two shapes of the same objects their point number distributions may be totally different. Different from a global normalization method, a local normalization method is utilized. For a global method, every cell feature value is divided by the total number of points in an 3D object. While, for a local method, each cell feature value is divided by the total sum of the feature values of the local region that the cell is belongs to.

For a comparison of two LSF vectors, χ^2 distance is employed rather than Euclidean distance.

$$\chi^2(F_1,F_2) = \sqrt{\sum_{c=1}^{L^3}\left(\frac{F_1(c)-E\chi(F_1)}{E\chi(F_1)}\right)+\sum_{c=1}^{L^3}\left(\frac{F_2(c)-E\chi(F_2)}{E\chi(F_2)}\right)} \quad (1)$$

In this function, F is the LSF vector, E function represents the expectation of the F.

4.1.2. Manifold ranking

Manifold can be embedded into a high-dimensional Euclidean space which recovers its intrinsic structure. This approach is to rank the 3D objects with respect to their intrinsic structures. Two manifolds of features are created. One manifold is for a 3D sketch which is compared with each 3D model. The other is for two different 3D models. In the first manifold, 3D sketches are used to train the visual vocabulary. In the second manifold, 3D models are used as the training data. In this way, the high-precision retrieval results can be obtained not only in the first manifold but also in the second manifold. The higher retrieval precision achieved for the retrieval between 3D sketches and 3D models, the better effect it is for the 3D sketch query.

Given the feature vectors $\chi = \{x_1, \cdots, x_q, x_{q+1}, \cdots, x_n\} \subset R^D$ of 3D sketches and 3D models, let $r : x \to R$ be a ranking function that assigns each point x_i a ranking score r_i. An initial vector is defined as $p = [p_1, \cdots, p_n]^T$, where p_i is the similarity between the query 3D sketch and the ith 3D model. The cost function $C(r)$ is defined as follows [LLL*15],

$$C(r) = \frac{1}{2}\sum_{i,j=0}^{n} W_{ij} \| \frac{r_i}{\sqrt{D_{ii}}} - \frac{r_j}{\sqrt{D_{jj}}} \|^2 + \mu \sum_{i=0}^{n} \|r_i - p_i\|^2 \quad (2)$$

where W is the affinity matrix, and D is the diagonal matrix $D_{ii} = \sum_j \sigma W_{ij}$. $\mu > 0$ is a regularization parameter.

The smaller the cost function is, the accurate the ranking is. Thus, $C(r)$ derivation operator is conducted by r, and the convergent function of the sequence: $r_i(t)$ is generated as below,

$$r^* = (I - \alpha S)^{-1} p \quad (3)$$

where, the matrix S is obtained through the symmetrical normalization of the matrix W, $S = D^{-1/2} W D^{-1/2}$, and E is the unit matrix. α is a parameter within (0,1), which defines the origin of the obtained ranking score of a point during the process of propagating the ranking score. $r = [r_1, \cdots, r_n]^T$ is the final ranking score.

4.2. Shape retrieval based on CNN (CNN-Point and CNN-Edge), by Y. Li, and M. Ovsjanikov

4.2.1. Pipeline

This 3D sketch-based shape retrieval method is also an image classification task. Both a target model and a query sketch are represented with a set of images, which are used to train and test a convolutional neural network. The diagram below shows the pipeline of this method combining training and testing phases.

The offline training phase transforms target models to sets of images, which has three steps:

- Step 1: Each model (all 1258 of them) is transformed to point cloud with some noise;
- Step 2: Each model is rendered from 120 uniformly distributed points of view to obtain 150960 images in total;
- Step 3: A neural network is trained on this dataset which takes as input an image and outputs a vector of dimension 1258 representing the probability of becoming each model.

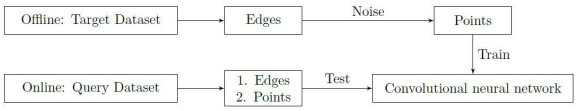


Figure 4: *Shape retrieval pipeline.*

Using this trained neural network, the online testing phase consists of five steps:

- Step 1: Each 3D sketch is preprocessed using *Edge representation* or *Point representation*;
- Step 2: Each preprocessed sketch is rendered from 90 uniformly distributed points of view;
- Step 3: For each sketch, its 90 derived images are tested with the neural network and the output vector was summed up;
- Step 4: The model is retrieved which has the same index as the maximum in the overall prediction vector;
- Step 5: The inverse function is used to get a distance-liked result.

4.2.2. Target preprocessing

The preprocessing of the target dataset transforms 3D models to point clouds with some noise. For each model, noise is randomly added alongside all edges according to their lengths. The number of points in each model is about 1000. Then each resulting point cloud model is rendered with a 128×128 grayscale image.

4.2.3. Query preprocessing

A specific noise-removal technique is not applied, but some points are arbitrarily chosen to be removed if they are too distant from their neighbors (in the sense of creation time). Do notice that some sketches are completed ruined by this method due to a large number of outliers.

4.2.3.1. Point representation: the point cloud is directly used after denoising.

4.2.3.2. Edge representation: in addition to the denoised point cloud, consecutive points are connected because they are recorded one by one in order. This will give a sketch-liked image.

Figure 5 shows one example for each of the above two representations.

Figure 5: *From left to right are 3D model, noised 3D model, point-based 3D sketch and edge-based 3D sketch.*

4.3. Histogram of Oriented Distances, by H. Tabia and H. Laga

This approach represents both 3D sketches and 3D models using the joint distribution of two parameters accumulated in a 2D histogram following an approach similar to spin images [JH99]. The descriptor, dubbed Histogram of Oriented Distances, or *HOD*, is constructed as follows.

First, randomly sample n points $P = \{p_i, i = 1...n\}$ from the shape. Then, compute for each pair of points $\{p_i, p_j\}$ the Euclidean distance $d_{ij} = \|p_i - p_j\|$ and measure the angle θ_{ij} between the two vectors $\overrightarrow{cp_i}$ and $\overrightarrow{cp_j}$, where c is the shape's center of mass. Finally, compute the probability distribution of the distance $d \in \mathbb{R}^+$ and the orientation $\theta \in [0, \pi]$ of the sampled pairs of points as a 2D histogram $h(d, \theta)$. Note that the slice of the 2D histogram corresponding to a fixed orientation θ is simply the D_2 shape distribution [OFCD02] of pairs of similarly oriented points from the center. By this representation, the global structure of the 3D sketches will be captured. The dissimilarity between the sketch and the target object can be easily computed using the L_2 distance between the two distributions. In this implementation, four different histogram sizes 64×4, 64×2, 64×1 and 1×4 are tested, where $k \times l$ corresponds to k bins for the distance and l bins for the orientation. The approach does not require any preprocessing of both the target 3D shapes and the 3D sketches than the normalization for scale and the sampling of random points. Figure 6 shows six different 64×4 histograms computed from six 3D sketches; (a) and (b) represent the histograms of two chairs, (c) and (d) are for two scissors, while (e) and (f) are for two houses.

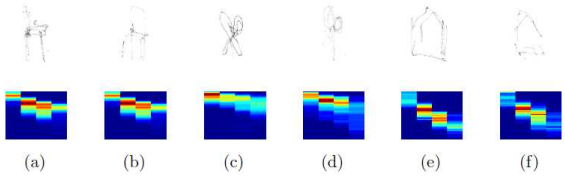


Figure 6: *Example of HOD descriptors of some 3D sketch shapes.*

4.4. CNN-SBR, by Y. Ye, Y. Lu and B. Li

This Convolutional Neural Network (CNN)-based 3D sketch-based shape retrieval architecture (CNN-SBR) is inspired by early sketch-based image retrieval work. The state-of-the-art deep Convolutional Neural Network (CNN) is employed in sketch object recognition and multiple 3D model processing techniques are combined in this work. First, pre-train the deep CNN model on the TU Berlin dataset [EHA12], which contains 20,000 free-hand sketches across 250 categories of daily objects, and obtain well-learned weights for the CNN model. Then, convert all the 3D sketches to multiple 2D sketch views for both the training and the testing datasets. Next, perform data augmentation for these 2D sketch views, and fine-tune the CNN model using previous well-learned weights. After that, the classification results for each query 3D sketch based on its 2D sketch views and a fine-tuned CNN model are obtained. Finally, apply majority vote and simple label matching to generate the output result for each testing query 3D sketch. The proposed CNN-SBR architecture is listed in Fig. 7.

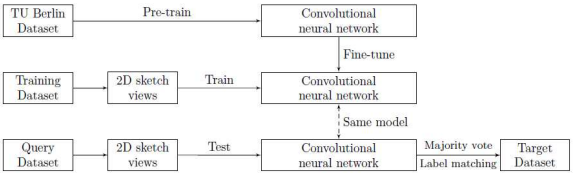


Figure 7: *Illustration of CNN-SBR framework.*

4.4.1. 2D sketch view generation

To adapt the CNN model for 3D sketch queries, the 3D sketches need to be converted to 2D sketch views. All the coordinates in each 3D sketch are projected into its six standard views (after aligned with PCA), and the coordinates are converted to 2D depth images where the pixel value represents the distance to its view point (0 is the nearest while 255 is the furthest).

4.4.2. Data augmentation

Data augmentation is a commonly-used technique in machine learning techniques to prevent over-fitting. In this algorithm, the 2D sketch views are replicated by 500 times using random vertical and horizontal shift, rotation, and flip operations.

4.4.3. Core Deep CNN model

On most popular image retrieval benchmarks, CNNs dominate the top performance. As shown in Fig. 8, Sketch-a-Net, which is a sketch-based CNN model designed for single sketch recognition problem, is applied as the core CNN model in the 3D model retrieval system.

4.4.4. Result generation

For each 3D sketch, use majority vote algorithm to choose the final classification label based on its six 2D sketch views.

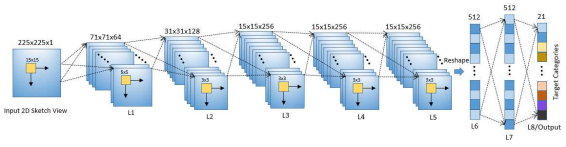


Figure 8: *Core deep CNN model: Sketch-a-Net.*

More specifically, for each 2D sketch view, a similarity vector (range: $[0,1)$) is obtained to predict categories. Thus, totally six similarity vectors and six most similar labels for six sketch views are obtained. Finally, use the formula "count of most similar label + average similarity" to rank all the target labels.

4.5. CNN-Maxout-Siamese, by H. Yin, S. Dong, P. Liu, Z. Xue, and H. Li

There are mainly three steps in this method, which are as follows.

4.5.1. Obtain 2D view and sketch pre-processing

In this approach, suggestive contours are used as the 2D line drawing rendering method for 3D models. For each model, two random sample views are chosen if their in-between angles are larger than 45° to characterize a 3D model. Each 3D sketch is randomly projected to three 2D images. Because the original sketch image dataset contains only a limited number of training images, data augmentation is performed to boost the performance.

4.5.2. Learn feature presentations

Siamese network, which typically takes a pair of images for input, is used to learn feature presentations. The two sub-nets of Siamese network have the same architecture–Convolutional Neural Networks (CNN). The sub-net architecture is shown in Fig. 9. To solve the over-fitting problem in CNN, maxout network is chosen.

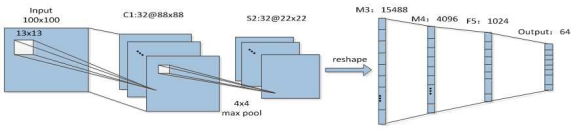


Figure 9: *The sub-net architecture of Siamese network.*

Given an input x, a maxout hidden layer is computed by the following function:

$$h_i(x) = max_{j\in[1,k]} z_{ij} \tag{4}$$

where $z_{ij} = x^T W_{ij} + b_{ij}$ and the dimension of W is $d*m*k$, d denotes the dimension of x, m denotes the number of hidden layer units, and k indicates the maximum number of "hidden layer" units.

Meanwhile, due to the gap existing between the domain of sketches and the domain of views, and the fact that the Siamese network is commonly used for the inputs from the same domain, two Siamese networks [WKL15] are defined: one for the view domain, and the other one for the sketch domain. The loss function computes the loss from both within-domain and cross-domain together:

$$\bigtriangledown(s_1,s_2,v_1,v_2,y) = L(s_1,s_2,y) + L(v_1,v_2,y) + L(s_1,v_1,y) \tag{5}$$

where s_1 and s_2 are two sketches, v_1 and v_2 are two views, s_1 and v_1 are from the same class, s_2 and v_2 are from the same class, and the loss function is of the following form:

$$L(x_1,x_2,y) = (1-y)\alpha D_w^2 + y\beta \exp^{-\frac{2.77}{\beta} D_w} \tag{6}$$

4.5.3. Similarity distance calculation

After getting the features, similarity distances between models and sketches are calculated by Euclidean distance.

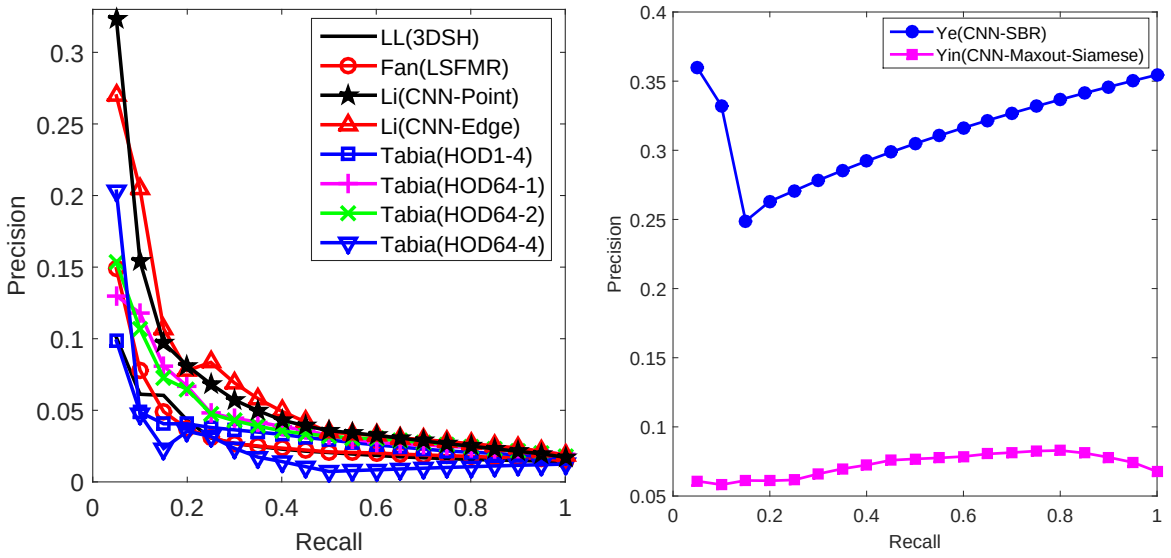
5. Results

In this section, we perform a comparative evaluation of the nine runs of the five methods submitted by the five groups. We measure retrieval performance based on the seven metrics mentioned in Section 2.3: PR, NN, FT, ST, E, DCG and AP.

As described in Section 2, the complete query sketch dataset is divided into the "Training" and "Testing" datasets, which is to accustom to learning-based retrieval algorithms. To provide complete reference performance data for both learning-based methods and non-learning based approaches (like), we evaluate the submitted results on both the "Testing" dataset and the "Complete" sketch dataset. Figure 10 and Table 1 compare three non-learning participating methods and two learning-based participating methods in terms of the seven performance metrics on the above two datasets, respectively. As a baseline, we also provide the baseline method 3D shape histogram (3DSH) that we have implemented in [LLG*15a, LLG*15b].

As shown in the aforementioned figure and table, in the non-learning based category, Li's CNN-Edge and CNN-Point algorithms perform the best, followed by Tabia's HOD method, while the overall performance of all non-learning based methods are close to each other. In the learning based category, Ye's CNN-SBR algorithm has better permanence than Yin's CNN-Maxout-Siamese. More details about the retrieval performance with respect to different classes for each participating method can be found in the track homepage [SHR16].

In addition, compared to the baseline method 3D shape



(a) Non-learning approaches (complete benchmark) (b) Learning-based approaches (testing dataset)

Figure 10: *Precision-Recall diagram performance comparisons on different datasets of the SHREC'16 3D Sketch Track Benchmark for three non-learning based and two learning based participating methods.*

Table 1: *Performance metrics comparison on the SHREC'16 3D Sketch Track Benchmark.*

Participant	Method	NN	FT	ST	E	DCG	AP
Complete benchmark							
LL [LLG*15a, LLG*15b]	3DSH	0.029	0.021	0.038	0.021	0.254	0.029
Fan	LSFMR	0.033	0.020	0.033	0.018	0.248	0.032
Li	CNN-Point	0.124	0.044	0.075	0.046	0.294	0.060
	CNN-Edge	0.114	0.056	0.084	0.051	0.302	0.063
Tabia	HOD1-4	0.029	0.015	0.035	0.026	0.259	0.032
	HOD64-1	0.052	0.031	0.053	0.034	0.274	0.044
	HOD64-2	0.067	0.031	0.057	0.032	0.272	0.044
	HOD64-4	0.124	0.019	0.022	0.013	0.230	0.026
Testing dataset							
Ye	CNN-SBR	0.222	0.251	0.320	0.186	0.471	0.314
Yin	CNN-Maxout-Siamese	0.000	0.031	0.108	0.048	0.293	0.072

histogram (3DSH), all the three non-learning approaches have achieved better overall performance, which further advances this research direction of 3D sketch-based 3D model retrieval. However, as can be seen from Fig. 10 and Table 1, the obtained retrieval performance of all the four non-learning algorithms are relatively close to each other and also still far from satisfactory.

On the other hand, though we cannot directly compare non-learning approaches and learning approaches together, we have found much more promising results in learning-based approaches. Even in the top-performing non-learning approaches Li's CNN-Edge and CNN-Point, the deep learning approach CNN contributes a lot to its better accuracy among the non-learning based approaches, in terms of automatically learning the features.

Since most of existing sketch-based retrieval methods drop apparently when adapted to this challenging 3D benchmark. Therefore, one urgent future work is to have more investigation in both learned and handcrafted features to develop better algorithms that can be scalable to diverse types of sketch queries, including 2D sketches or images and 3D sketches. To achieve this, one approach is utilizing techniques from other related disciplines, such as machine learning, especially the currently most popular and promising machine learning technique –deep learning– to automatically learn the features, rather than selecting and fixing the features beforehand.

Finally, we classify all participating methods with respect to the techniques employed: four participating groups (Fan, Li, Ye, Yin) utilize local features while Tabia and the baseline method (3DSH) employ a global feature. Three groups (Li, Ye, Yin) employ deep learning framework to learn the features automatically, while both of the other two groups (Fan and Tabia) extract a statistical distribution of local features to represent a 3D model/sketch. But Fan further applies the Bag-of-Features framework and Manifold Ranking as well. On the other hand, Tabia directly computes the distance based on the distributions of sketches and models, similar to that in the baseline approach 3DSH.

6. Conclusions

3D sketches are potential in bridging the semantic gap existing between the inaccurate 2D sketch queries and accurate 3D model representations for the same object we want to search in the scenario of 2D sketch-based 3D model retrieval. In conclusion, this 3D sketch-based 3D model retrieval track is to further foster the challenging and interesting research direction of sketch-based 3D model retrieval, encouraged by the success of SHREC'12 [LSG*12, LLG*14], SHREC'13 [LLG*13, LLG*14] and SHREC'14 [LLL*14, LLL*15] sketch-based 3D shape retrieval tracks. Though 3D sketch-based shape retrieval is even more challenging than 2D based, we still have five groups who have successfully participated in the track and contributed nine runs of five methods. This track provides a common platform to solicit current sketch-based 3D model retrieval approaches in terms of this 3D sketch-based retrieval scenario. It also helps us identify state-of-the-art methods as well as future research directions for this research area. We also hope that the 3D sketch retrieval benchmark, together with the retrieval results we have obtained in the track, will become a useful reference for researchers in this community.

Acknowledgments

This project and the work of Yijuan Lu is supported by Army Research Office grant W911NF-12-1-0057 and NSF CNS 1305302 to Dr. Yijuan Lu.

References

[EHA12] EITZ M., HAYS J., ALEXA M.: How do humans sketch objects? *ACM Trans. Graph. 31*, 4 (2012), 44:1–44:10. 5

[Her10] HERTZMANN A.: Non-photorealistic rendering and the science of art. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering 2010, Annecy, France, June 7-10, 2010* (2010), pp. 147–157. 3

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

[LLG*13] LI B., LU Y., GODIL A., SCHRECK T., AONO M., JOHAN H., SAAVEDRA J. M., TASHIRO S.: SHREC'13 track: Large scale sketch-based 3D shape retrieval. In *3DOR* (2013), pp. 89–96. 2, 8

[LLG*14] LI B., LU Y., GODIL A., SCHRECK T., BUSTOS B., FERREIRA A., FURUYA T., FONSECA M. J., JOHAN H., MATSUDA T., OHBUCHI R., PASCOAL P. B., SAAVEDRA J. M.: A comparison of methods for sketch-based 3D shape retrieval. *Computer Vision and Image Understanding 119* (2014), 57–80. 8

[LLG*15a] LI B., LU Y., GHUMMAN A., STRYLOWSKI B., GUTIERREZ M., SADIQ S., FORSTER S., FEOLA N., BUGERIN T.: 3D sketch-based 3D model retrieval. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, Shanghai, China, June 23-26, 2015* (2015), pp. 555–558. 2, 6, 7

[LLG*15b] LI B., LU Y., GHUMMAN A., STRYLOWSKI B., GUTIERREZ M., SADIQ S., FORSTER S., FEOLA N., BUGERIN T.: KinectSBR: A kinect-assisted 3D sketch-based 3D model retrieval system. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, Shanghai, China, June 23-26, 2015* (2015), pp. 655–656. 2, 6, 7

[LLL*14] LI B., LU Y., LI C., GODIL A., SCHRECK T., AONO M., BURTSCHER M., FU H., FURUYA T., JOHAN H., LIU J., OHBUCHI R., TATSUMA A., ZOU C.: SHREC'14 Track: extended large scale sketch-based 3D shape retrieval. In *Eurographics Workshop on 3D Object Retrieval, Strasbourg, France, 2014. Proceedings* (2014), pp. 121–130. 8

[LLL*15] LI B., LU Y., LI C., GODIL A., SCHRECK T., AONO M., BURTSCHER M., CHEN Q., CHOWDHURY N. K., FANG B., FU H., FURUYA T., LI H., LIU J., JOHAN H., KOSAKA R., KOYANAGI H., OHBUCHI R., TATSUMA A., WAN Y., ZHANG C., ZOU C.: A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries. *Computer Vision and Image Understanding 131* (2015), 1–27. 4, 8

[LSG*12] LI B., SCHRECK T., GODIL A., ALEXA M., BOUBEKEUR T., BUSTOS B., CHEN J., EITZ M., FURUYA T., HILDEBRAND K., HUANG S., JOHAN H., KUIJPER A., OHBUCHI R., RICHTER R., SAAVEDRA J. M., SCHERER M., YANAGIMACHI T., YOON G.-J., YOON S. M.: SHREC'12 track: Sketch-based 3D shape retrieval. In *Eurographics Workshop on 3D Object Retrieval (3DOR), 2012* (2012), pp. 109–118. 8

[OFCD02] OSADA R., FUNKHOUSER T. A., CHAZELLE B., DOBKIN D. P.: Shape distributions. *ACM Trans. Graph. 21*, 4 (2002), 807–832. 5

[SHR16] http://cs.txstate.edu/ yl12/SBR2016//, 2016. 6

[WKL15] WANG F., KANG L., LI Y.: Sketch-based 3D shape retrieval using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015* (2015), pp. 1875–1883. 6