3D Sketch-based 3D Model Retrieval with Convolutional Neural Network

Yuxiang Ye
Department of Computer Science
Texas State University
San Marcos, Texas 78666
Email: y_y13@txstate.edu

Bo Li[1,2]
[1] School of Computing
University of Southern Mississippi
Long Beach, Mississippi 39560
Email: bo.li@usm.edu
[2] Department of Computer Science
Texas State University
San Marcos, Texas 78666

Yijuan Lu
Department of Computer Science
Texas State University
San Marcos, Texas 78666
Email: lu@txstate.edu

***Abstract*—3D sketch-based 3D model retrieval is to retrieve similar 3D models using users' hand-drawn 3D sketches as input. Compared with traditional 2D sketch-based retrieval, 3D sketch-based 3D model retrieval is a brand new and challenging research topic. In this paper, we employ advanced deep learning method and propose a novel 3D sketch based 3D model retrieval system. Our system has been comprehensively tested on two benchmark datasets and compared with other existing 3D model retrieval algorithms. The experimental results reveal our approach outperforms other competing state-of-the-arts and demonstrate promising potential of our approach on 3D sketch based applications.**

I. Introduction

Traditional sketch-based 3D model retrieval systems are built on 2D sketching technology, which require users to draw sketches on a 2D plane (such as paper, touch screen). However, constraining user's sketch to two dimensional space limits the 3D information that the shape can convey. Therefore, 3D sketching technology was introduced in 2015 [1][2][3]. 3D sketching allows users to sketch an object in a 3D space (for example in the air) by tracking the human hand's motion with Kinect. People think 3D sketches should provide a better description of the object than 2D sketches. However, there is a lack of a comprehensive study on 3D sketching challenging, 3D sketch understanding, and 3D sketch based applications. How to understand (translate) 3D sketches drawn by human hands and how to match 3D sketches with 3D models become new research problems.

In this paper, we perform an initial study on 3D sketch understanding and 3D sketch based 3D model retrieval. Although it seems that 3D sketches encode more shape information (depth, salient 3D features) than 2D sketches, our research show that 3D sketches understanding is even more challenging than 2D sketches understanding due to its complexity, variation, and uncertainty: (i) Complexity: 3D sketching is more complex than 2D sketching. For example, drawing a 3D horse is more difficult than drawing a 2D horse on a paper. Most of people just try to sketch rough 3D outlines of an object (Fig. 1). However, these rough outlines are far away from the real object contours, which introduce a lot of difficulty for computer to recognize the objects depicted and understand the semantic information implicated. (ii) Variation: one thousand people may draw the same object in one thousand different ways. This issue is particularly obvious in 3D sketching. Two people are not even able to draw exact the same 3D dogs. (iii) Uncertainty: 3D sketches only record 3D coordinates of all the individual points captured from human's hand movement during sketching. A lot of noisy and inaccurate points are captured due to hand shaking, object occlusion, and camera delay. These difficulties make 3D sketch understanding and 3D sketch based 3D model retrieval very challenging.

To match 3D sketches with 3D models, the most direct way is to extract shape features from both 3D sketches and 3D models and compare the distances between them. However, it turns out such kind of approach doesn't work well [2] due to the big gap between the abstract representation of a 3D sketch and the accurate 3D coordinate representation of a 3D model.

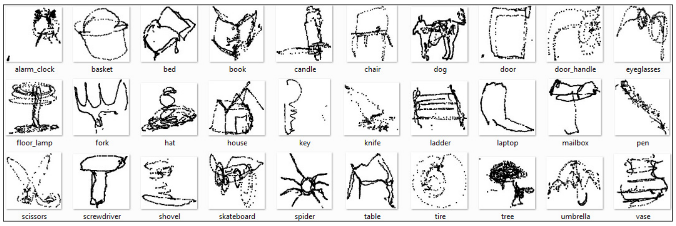

Fig. 1: Example 3D sketches of Kinect300 dataset

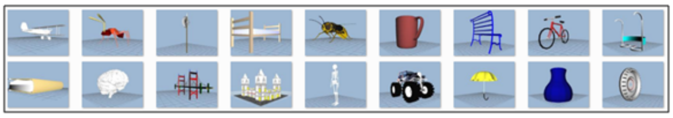

Fig. 2: Example 3D models of the SHREC13STB benchmark

Inspired by the above challenges, in this work, we propose a novel 3D sketch-based 3D model retrieval system CNN-SBR using multiple advanced deep learning and 3D model processing techniques. We evaluate our CNN-SBR system with other state-of-the arts on SHREC'16 3D Sketch Track Benchmark which consists of two parts: (i) 3D sketch dataset which consists of 300 3D sketches (30 categories, 10 sketches per category, also called Kinect300 [1]) and 21 categories have the corresponding models in the target SHREC13STB dataset. Fig. 1 shows some example 3D sketches. (ii) 3D model dataset. This 3D benchmark dataset is built on the SHREC13STB, which consists of 1258 models unevenly distributed in 90 categories. Fig. 2 shows some example 3D models.

To our knowledge, CNN-SBR system performs best among all the existing retrieval systems that enable users to search 3D models based on hand-drawn 3D sketches. The main contributions introduced in this work are highlighted as follows:

- A novel 3D sketch-based 3D model retrieval system is introduced to solve the matching problem between 3D sketches and 3D models.
- Our CNN-SBR system combines multiple machine learning and 3D vision processing techniques, which will explicitly guide the research in 3D sketch understanding
- Comprehensive experiments have been conducted to evaluate the state-of-the-art sketch based retrieval approaches on 3D sketch-based 3D model retrieval.
- The experimental results not only show our approach outperforms other state-of-the-arts, but also demonstrates promising application potential of our approach on 3D sketch understanding, on-line 3D model shopping, and large scale 3D model search, *etc*.

II. Related Work

Sketch-based 3D model retrieval targets on retrieving 3D models given a hand-drawn query sketch. Recently, sketch-based 3D model retrieval has attracted much attention since it can be widely used in sketch-based rapid prototyping, recognition, mobile 3D search, 3D printing, 3D animation production and etc. In this section, we review three related areas for our work: sketch recognition, deep CNNs for visual recognition and sketch-based 3D model retrieval.

A. Sketch Recognition

Most early sketch datasets are the small scale collections include: artistic drawings [4], professional CAD figures [5], and specific domain structure sketches [6], [7].

Recently, a large-scale collection of free-hand sketches (TU Berlin dataset [8]) is open to the public. It contains 20,000 single-object sketches in 250 daily object categories.

In earlier sketch recognition works (*i.e.,* [9] [10]), sketching is introduced as a human-computer interaction technology in which mouse or pen is used to draw lines and curves. Recently, researchers have explored more hand-crafted features, such as stroke length, stroke order and even stroke orientation to understand human sketch input at a higher level. Eitz *et al.* [8] demonstrated sHOG feature coupled with Bag-of-Words (BoW) methods on sketch understanding and achieved a promising result of 56% accuracy in identifying unknown sketches on TU Berlin dataset.

Very recently, [11] explores that Fisher Vectors, an image feature representation obtained by pooling local image features, can be applied to single-object sketch recognition and achieves human-like sketch recognition accuracy (68.9% vs. 73.1% for human on the TU Berlin dataset). Despite these great efforts, hand-crafted features still require researchers to have solid knowledge on drawings or specific domains. More general sketch understanding approaches need to be explored.

B. Deep CNNs for Visual Recognition

Recently, deep CNNs have showed promising results on many vision recognition tasks in different domains. CNN was introduced in early 1980s, and was applied to solve simple and small vision recognition tasks like handwritten digit recognition [12].

In early time, the biggest bottleneck of deep CNN was the high computational cost when the number of classes and input data volume are large, which significantly increase the number of neurons in CNN. However, with the proliferation of modern GPUs, this bottleneck has been relieved. With the introduction of rectifier linear (ReLU) [13], max-pooling, local response normalization (LRN) [14], and dropout regularization units [15], CNNs become more effective, robust, and applicable. In particular, some benchmarks' top results in visual recognition challenges, *e.g.,* ILSVRC [16], have been dominated by deep CNNs-based approaches. Yu *et al.* [17] designed a sketch-oriented deep CNN model "Sketch-A-Net" for sketch recognition task and achieved the accuracy of 74.9% on TU Berlin dataset. Nevertheless, most previous works focus on 2D sketch understanding tasks.

C. Sketch-Based 3D Model Retrieval

In computer vision community, the effort of sketch-based retrieval research work has been introduced for many years [18]. Early works on this task focus on global shape descriptors, such as distance functions [19] and shape statistics [20]. Recently, researchers employ local features for partial matching [21] for 3D model retrieval.

Most of the above 3D model representation methods are borrowed from traditional 2D feature, such as BF-DSIFT [22] which is an extended SIFT feature with Bag-of-Features. Therefore, it is important to choose a good representation of line drawing images for sketch-based 3D model retrieval. Recently, Eitz *et al.* [23] proposed the GALIF and built on a collection of Gabor filters followed by a BoW method.

Instead of relying on the traditional 2D image features, some methods also explore graph-based feature [24] and semantic labeling [25] to facilitate the 3D model retrieval. In this work, we employ view-based methods and only use 2D sketch features in 3D model retrieval.

III. Our Approach

A. CNN-SBR Architecture Overview

Our CNN-SBR system, which is demonstrated in Fig. 3, is inspired by early sketch-based image retrieval work. We employ state-of-the-art deep CNN in sketch object recognition

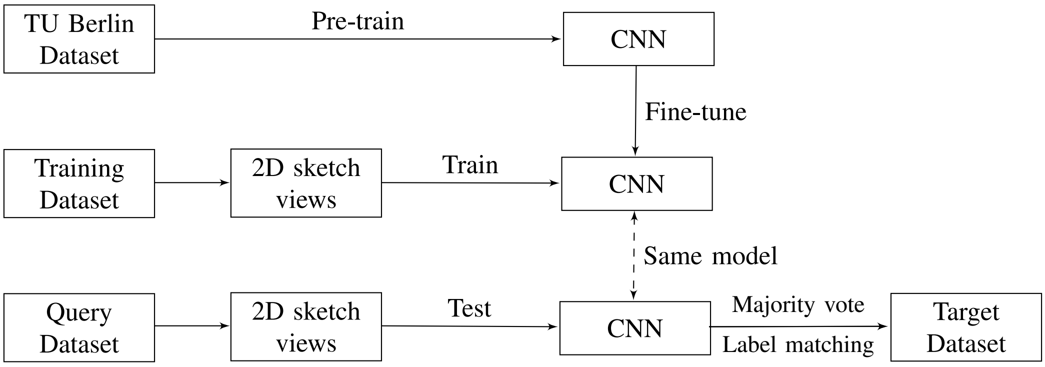

Fig. 3: Illustration of CNN-SBR architecture

and combine multiple 3D model processing techniques in this work. We first pre-train our deep CNN model on TU Berlin dataset and obtain well-learned weights for our CNN model. Then, we convert all the 3D sketches to multiple 2D sketch views for both training and query datasets, and perform data augmentation for these 2D sketch views, then fine-tune the CNN model using previously well-learned weights. After that, we have the classification result for each query 3D sketch based on its 2D sketch views and fine-tuned CNN model. Finally, we use majority vote and simple label matching to generate the output result.

B. Data Processing

To adapt the framework for 2D sketch-based CNN model, we need to convert the 3D sketches to 2D sketch views. We project all the coordinates in each 3D sketch to its six square faces, if we regard a 3D sketch as regular hexahedron, and map the 3D coordinates to 2D depth image where the pixel value represents the distance to its view point (0 is the nearest while 255 is the furthest).

We also employ the data augmentation technique to prevent the over-fitting issue in deep CNN method. In our experiments, we replicate both TU Berlin dataset and 2D sketch views by 500 times using random rotation, shift and flip. More specifically, we describe our algorithm as **Algorithm** 1.

Based on this algorithm, an image duplicate would be generated with one or more image transformations of shift, rotation and flip. We introduce this random data augmentation algorithm to increase the variety of the experiment dataset, which significantly reduces the sketching noise from different user hand-drawing.

Input: Original 3D sketch dataset S
Output: Enlarged 3D sketch dataset T with random shifts, rotations, and flips

initialization;
$w = width_{original} - width_{target}$;
foreach $I \in S$ **do**
  for $i \leftarrow 1$ **to** 500 **do**
    $C \leftarrow copy(I)$;
    $x_{shift} \leftarrow random(0, w)$;
    $y_{shift} \leftarrow random(0, w)$;
    $C \leftarrow shift(C, x_{shift}, y_{shift})$;
    $roll \leftarrow random(0, 1)$;
    if $roll < 0.5$ **then**
      $rd \leftarrow random(-5, 5)$;
      $C \leftarrow rotate(C, rd)$;
    end
    $roll \leftarrow random(0, 1)$;
    if $roll < 0.5$ **then**
      $C \leftarrow flip(C)$
    end
    $append(T, C)$;
  end
end

Algorithm 1: Data augmentation algorithms

C. Applying Sketch-A-Net to CNN-SBR System

In this work, we employ Sketch-A-Net as our core CNN model for our CNN-SBR 3D model retrieval system. Sketch-A-Net is designed for 2D single object sketch recognition problem. Based on our best knowledge, Sketch-A-Net has the best performance on 2D single object sketch recognition. We feed it with the TU Berlin dataset as the ancillary training dataset for pre-training. Although the TU Berlin dataset is obtained from human 2D sketching, it provides Sketch-A-Net a good representation of drawing features which could help Sketch-A-Net to build an initial learning weight for future fine-tuning on 3D sketch dataset.

In fine-tuning step, instead of employing mature learned pre-trained model (training epoch over 500), we choose the pre-trained model at epoch 50, or a semi-mature model, whose local optima has not been convergent yet and the learning weights are still flexible. We find using this method would significantly alleviate the over-fitting issue and improve the retrieval performance.

D. Majority Vote and Label Matching

For each 3D sketch, we use majority vote algorithm to choose the final classification label based on its six 2D sketch views. More specifically, for each 2D sketch view, we have a similarity vector for predicting categories. Thus, we have

TABLE I: Performance metrics comparison on the SHREC'16 [26] 3D Sketch Track Benchmark.

Participant	Method	NN	FT	ST	E	DCG	AP
Complete benchmark (Non-learning based methods)							
LL	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 (Learning-based methods)							
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

totally six similarity vectors and six top-1 labels for six sketch views. In order to rank the categories correctly, we use the following method to evaluate the similarity vectors:

1) We re-scale the similarities between a 3D sketch and target 3D model categories to range $[0, 1]$. A higher value means bigger similarity.
2) For each target 3D model category, we first count the number of top-1 labels among six similarity vectors. The top-1 label count must be an integer in the range of $[0, 6]$.
3) We then compute the average similarity between this sketch and target 3D model categories based on six similarity vectors. Average similarity would also fall into the range of $[0, 1]$.
4) Finally, we rank all the target 3D model categories using the summation of the top-1 label count and the average similarity, and then rank all the related models accordingly.

Based on the above algorithm, we first consider those target 3D model categories that have the most top-1 label count and rank them at the top places. It should be noted that in the experiment there are only six top-1 labels but 21 target 3D model categories. In this case, some target categories may have the same top-1 label count, then we need to compute the average similarities between the input 3D sketch and all the target 3D model categories to rank those categories. After that, for each query 3D sketch, we can simply rank the target 3D models based on the rank of their categories which are obtained in the above method.

IV. Experiments and Comparisons

To comprehensively evaluate the performance of our CNN-SBR system, we participated in a 2016 Shape Retrieval Contest (SHREC'16) track [26] which targets on 3D sketch-based 3D model retrieval. There are six teams who participated in this SHREC'16 challenge and one baseline method (3DSH [27]). All the participating algorithms are evaluated either on the test dataset (3 sketches per class, totally 90 3D sketches) of the SHREC'16 3D sketch track benchmark for learning based algorithms or on the complete dataset (10 sketches per class, 300 sketches) for non-learning based algorithms. Our CNN-SBR system achieved the best performance on all the evaluation metrics in the SHREC'16 challenge. In this section, we compare our CNN-SBR system with several other participating methods and discuss the possible reasons why our method outperforms others.

A. Running Cost

We implemented CNN-SBR system using Matlab and the MatConvNet toolbox. All the experiments were excuted on a server with an 8-core 3.50GHz CPU and a GeForce GTX Titan X GPU. The pre-training time on the TU Berlin dataset is approximately 1 hour on GPU, while fine-tuning on Kinect300 dataset is about 30 minutes on GPU. Label matching and majority voting only take several minutes.

B. Evaluation Results

We perform the comparison based on six widely-used evaluation metrics [28]: Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), E-Measure (E), Discounted Cumulative Gain (DCG), Average Precision (AP) on the test dataset of SHREC'16 3D Sketch Track Benchmark for learning based participating algorithms and on the complete dataset for non-learning based algorithms. The results are compared in Table I. We also perform the Precision-Recall comparison in Fig. 4.

From the above table and figure, we can see that our CNN-SBR retrieval system beats other learning-based participating algorithms on all the evaluation metrics. The performance is also significantly better than those achieved by non-learning based approaches. We can also learn from this table that 3D sketch-based 3D model retrieval is a very challenging task. Most participating methods have less than 10% accuracy on the nearest neighbor metric.

C. Competition Methods

We now review some representative competition methods in SHREC'16 challenge.

LSFMR is a non-learning based method and proposed by Fan *et al.* It consists of two components: LSF extraction and Manifold ranking. In preprocessing, SVM is applied to remove the noise points of the 3D sketches while PCA is applied to normalize their positions. Then, the local features of the training 3D sketch data are clustered using k-means method.

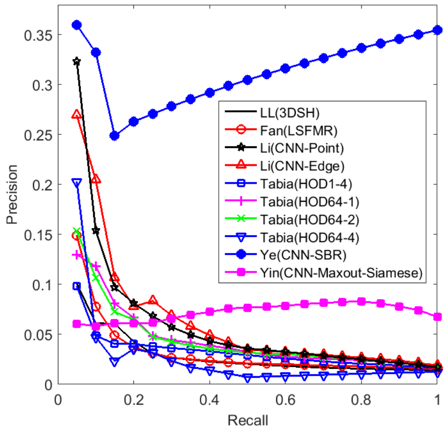

Fig. 4: Precision-Recall comparison on the complete dataset of SHREC13STB for non-learning based algorithms and on the testing dataset for learning based algorithms

LSF feature describes the local region shape, which comes with the dense grid division and serves as a local statistical feature. A dense network is built based on this feature and it captures both the local and global features. The local region is divided into $L \times L \times L$ cells to calculate the 3D points statistical distribution in the local region. For each cell, the feature value is the number of points in the cell. The final representation of a local region is 1-D vector constructed by accumulating all the cell feature values. To compare two LSF vectors, instead of employing Euclidean distance, χ^2 distribution is used.

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

where F denotes LSF vector, $E\chi$ denotes the expectation of F.

HOD is a non-learning based method and proposed by Tabia *et al.* The main idea of HOD is to build a dubbed Histogram of Oriented Distances descriptor based on the joint distribution of two parameters accumulated in a 2D histogram. The algorithm is constructed as follows:

1) Randomly sample n points $P = \{p_i, i = 1 \dots n\}$ from the 3D sketch.
2) Compute the Euclidean distance $d_i = \|p_i - p_j\|$ and measure the angle θ_{ij} between the two vectors $\overrightarrow{cp_i}$ and $\overrightarrow{cp_j}$, where c is the sketch's center of mass.
3) Compute the probability distribution of the distance $d \in \mathbb{R}^+$ and the orientation $\theta \in [0, \pi]$ of the sample pairs of points as a 2D histogram $h(d, \theta)$.

CNN-Point and **CNN-Edge** are non-learning based methods and proposed by Li and M. Ovsjanikov. The only difference between these two methods is the processing techniques that were applied on the query dataset. First, it transform a 3D model to a point cloud, while noise is randomly added during its transformation. Similarly, a query 3D sketch is also converted to a point-based or edge-based 3D sketch. Then, each point cloud is rendered from 120 uniformly distributed viewpoints to generate a gray-scale image of size 128×128. Finally, a classifier is trained on the resulting image dataset, which consists of 150960 images.

CNN-Maxout-Siamese is a learning-based method and proposed by Yin *et al.* CNN-Maxout-Siamese follows a similar design scheme like our CNN-SBR. In this system, it employs randomly sampled 2D views, together with their data augmentation, as the input and feed them into two Siamese CNN network phases (one for view domain and the other for sketch domain). Finally, the similarity distances between the 3D models and the query 3D sketches are calculated based on the Euclidean distance.

It should be noted that, among all the participating methods only CNN-SBR and CNN-Maxout-Siamese are learning-based methods. Compared with other non-learning based methods, we can find they still have a big performance gap if compared with our CNN-SBR system. CNN-Maxout-Siamese employs a similar design scheme as CNN-SBR, but its nearest neighbor (NN) accuracy is close to 0.

If we review the details of implementation, our CNN-SBR system has the following advantages over other competitors:

- Instead of extracting conventional hand-crafted features, like **HOD**, **3DSH** and **LSFMR**, we employ deep CNN as a more general and adaptive feature learning strategy.
- CNN-SBR performs data augmentation on 2D sketch views of 3D sketch, which significantly enlarges the dataset size and alleviates over-fitting.
- CNN-SBR applies Sketch-A-Net as our core CNN model and pre-trains the model using TU Berlin dataset. Other CNN-based methods, including **CNN-Point**, **CNN-Edge**, and **CNN-Maxout-Siamese**, do not choose Sketch-A-Net as the core CNN model. They also do not employ this pre-training technique to boost the performance.
- Semi-mature CNN model after pre-training, which gives initial well-learned weights but still keeps the CNN model flexible for a following fine-tuning.

V. Conclusion and Future Work

3D sketching in 3D space and 3D sketch-based 3D model retrieval are brand new research topics. Very little preliminary work exists in this field, which allows us for exploring many exciting and interesting results. In this paper, a novel 3D sketch-based 3D model retrieval system CNN-SBR is proposed. CNN-SBR outperforms all other participating algorithms over all the evaluation metrics and achieves the top-1 performance in the SHREC'16 3D sketch-based 3D shape retrieval challenge. In the paper, we also discuss the advantages of our CNN-SBR approach, compared with other participating algorithms. Future goals include a further exploration of adaptively representing 3D sketches and 3D models using CNNs, and training our system for better 3D sketch and 3D model matching based on a larger collection of 3D sketches from more diverse users.

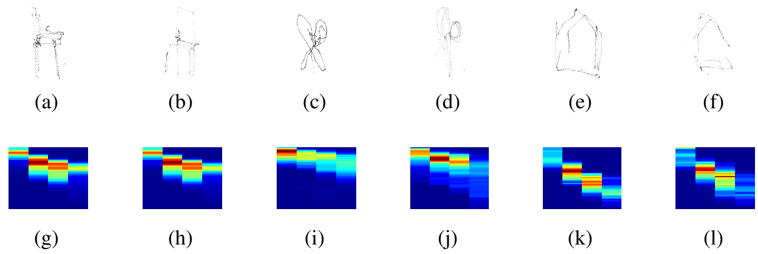

Fig. 5: Example of HOD descriptors of some 3D sketch shapes.

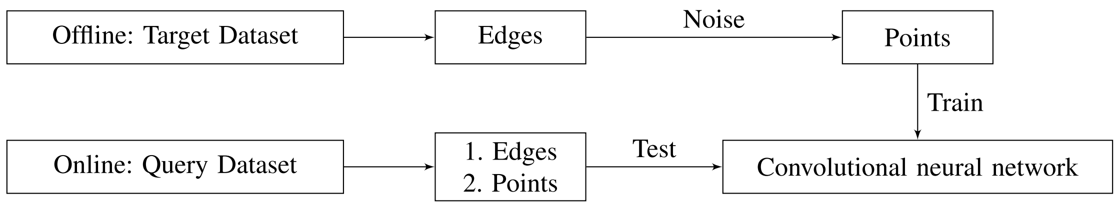

Fig. 6: Shape retrieval pipeline of CNN-Point/CNN-Edge

ACKNOWLEDGMENT

This work is supported by Army Research Office grant W911NF-12-1-0057, NSF CNS-1305302 and NSF CNS-1358939 to Dr. Lu.

REFERENCES

[1] B. Li, Y. Lu, and et al., "Shrec16 track: 3d sketch-based 3d shape retrieval," in *Eurographics Workshop on 3D Object Retrieval (3DOR 2016)*, Lisbon, Portugal, 2016.
[2] B. Li, Y. Lu, A. Ghumman, B. Strylowski, M. Gutierrez, S. Sadiq, S. Forster, N. Feola, and T. bugerin, "3d sketch-based 3d model retrieval," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, New York, NY, USA, 2015, pp. 555–558.
[3] F. Wang, L. Kang, and Y. Li, "Sketch-based 3d shape retrieval using convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015, pp. 1875–1883.
[4] P. Sousa and M. J. Fonseca, "Geometric matching for clip-art drawing retrieval," *J. Vis. Comun. Image Represent.*, vol. 20, no. 2, pp. 71–83, Feb. 2009.
[5] M. Mohamad, M. Shafry, M. Rahim, N. Othman, and Z. Jupri, *A comparative study on extraction and recognition method of CAD data from CAD drawings*, 10 2009, pp. 709–713.
[6] T. Y. Ouyang and R. Davis, "Chemink: A natural real-time recognition system for chemical drawings," in *Proceedings of the 16th International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, 2011, pp. 267–276.
[7] T. Lu, C.-L. Tai, F. Su, and S. Cai, "A new recognition model for electronic architectural drawings," *Computer-Aided Design*, vol. 37, no. 10, pp. 1053 – 1069, 2005.
[8] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" *ACM Trans. Graph.*, vol. 31, no. 4, pp. 44:1–44:10, Jul. 2012.
[9] C. F. Herot, "Graphical input through machine recognition of sketches," *SIGGRAPH Comput. Graph.*, vol. 10, no. 2, pp. 97–102, Jul. 1976.
[10] I. E. Sutherland, "Sketch pad a man-machine graphical communication system," in *Proceedings of the SHARE Design Automation Workshop*, ser. DAC '64. New York, NY, USA: ACM, 1964, pp. 6.329–6.346.
[11] R. G. Schneider and T. Tuytelaars, "Sketch classification and classification-driven analysis using fisher vectors," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 174:1–174:9, Nov. 2014.
[12] Y. LeCun, B. E. Boser, and et al., "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed. Morgan-Kaufmann, 1990, pp. 396–404.
[13] Y. LeCun, L. Bottou, G. B. Orr, and K. R. Müller, "Neural networks: Tricks of the trade," in *Efficient BackProp*. Springer, 2002, pp. 9–50.
[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.
[15] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.
[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.
[17] Q. Yu, Y. Yang, Y.-Z. Song, T. Xiang, and T. Hospedales, "Sketch-a-net that beats humans," in *arXiv:1501.07873*, 2015.
[18] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs, "A search engine for 3d models," *ACM Trans. Graph.*, vol. 22, no. 1, pp. 83–105, Jan. 2003.
[19] M. Kazhdan, B. Chazelle, D. Dobkin, A. Finkelstein, and T. Funkhouser, "A reflective symmetry descriptor," in *7th European Conference on Computer Vision (ECCV 2002)*, May 2002, pp. 642–656.
[20] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Trans. Graph.*, vol. 21, no. 4, pp. 807–832, Oct. 2002.
[21] T. Funkhouser and P. Shilane, "Partial matching of 3d shapes with priority-driven search," in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, ser. SGP '06. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2006, pp. 131–142.
[22] T. Furuya and R. Ohbuchi, "Ranking on cross-domain manifold for sketch-based 3d model retrieval," in *Cyberworlds (CW), 2013 International Conference on*, Oct 2013, pp. 274–281.
[23] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, and M. Alexa, "Sketch-based shape retrieval," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 31, no. 4, pp. 31:1–31:10, 2012.
[24] B. Li, Y. Lu, and et al., "A comparison of 3d shape retrieval methods based on a large-scale benchmark supporting multimodal queries," *Computer Vision and Image Understanding*, vol. 131, pp. 1 – 27, 2015, special section: Large Scale Data-Driven Evaluation in Computer Vision.
[25] B. Gong, J. Liu, X. Wang, and X. Tang, "Learning semantic signatures for 3d object retrieval," *IEEE Transactions on Multimedia*, vol. 15, no. 2, pp. 369–377, Feb 2013.
[26] "Shrec'16," http://cs.txstate.edu/ỹl12/SBR2016/index.html/, 2016.
[27] B. Li, Y. Lu, and et al., "SHREC'13 Track: Large Scale Sketch-Based 3D Shape Retrieval," in *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2013.
[28] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton shape benchmark," in *Shape Modeling International*, Jun. 2004.