Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Learning Cross-Domain Neural Networks for Sketch-Based 3D Shape Retrieval

Fan Zhu, Jin Xie and Yi Fang*
NYU Multimedia and Visual Computing Lab
Electrical and Computer Engineering, New York University Abu Dhabi
Abu Dhabi, UAE, PO Box 129188

Abstract

Sketch-based 3D shape retrieval, which returns a set of relevant 3D shapes based on users' input sketch queries, has been receiving increasing attentions in both graphics community and vision community. In this work, we address the sketch-based 3D shape retrieval problem with a novel Cross-Domain Neural Networks (CDNN) approach, which is further extended to Pyramid Cross-Domain Neural Networks (PCDNN) by cooperating with a hierarchical structure. In order to alleviate the discrepancies between sketch features and 3D shape features, a neural network pair that forces identical representations at the target layer for instances of the same class is trained for sketches and 3D shapes respectively. By constructing cross-domain neural networks at multiple pyramid levels, a many-to-one relationship is established between a 3D shape feature and sketch features extracted from different scales. We evaluate the effectiveness of both CDNN and PCDNN approach on the extended large-scale SHREC 2014 benchmark and compare with some other well established methods. Experimental results suggest that both CDNN and PCDNN can outperform state-of-the-art performance, where PCDNN can further improve CDNN when employing a hierarchical structure.

Introduction

The human freehand sketch is a succinct, convenient and efficient way to visually record and present humans' ideas. Along with the recent developments of consumer electronic devices (e.g., touch-pad mobile phones), freehand sketches are becoming one of the mainstream human-computer interaction methods, and are expected to become the base of many more applications. While sketches mainly convey abstract descriptions of objects, 3D shapes, on the other hand, contain the majority information that are required for industrial productions. In the traditional fashion of design operation processes, design ideas and intents are progressively and iteratively explored in the form of sketches at the initial stage, after which finalized drafts are transformed to digital 3D shapes, where this process is normally conducted manually and is very time consuming. As an improvement, a sketch-based 3D shape retrieval system can significantly simplify this process from the following two aspects: 1) statistical investigation (Gunn 1982) suggests that 80% of design work can benefit from directly reusing or modifying an existing design; 2) instead of manually transforming the draft design into digital forms, a final product prototype can be obtained by directly reusing or modifying a desired 3D shape selected from a pool of retrieved 3D shapes based on the input draft.

In this work, we address a novel and challenging designer-computer interaction task, sketch-based 3D shape retrieval. In order to alleviate the domain discrepancy between sketches and 3D shapes, we construct pyramid cross-domain neural networks (PCDNN), which map the mismatched sketch and 3D shape low-level representations to a unified feature space at multiple pyramid levels. The pyramid structure is defined by a fixed hierarchy of rectangular windows, and computes local histogram features within each subdivided image regions. Within each pyramid level, a cross-domain neural network (CDNN) pair with identical target layers for objects of the same category is learned. By extending a CDNN to the pyramid structure, multi-resolution sketch histograms are mapped to corresponding 3D shapes, so that the homogeneous information (i.e., shared commonness) between sketches and 3D shapes can be captured at finer levels when compared with a single-level structure that performs on the concatenation of all pyramid levels of sketch representations. When sketch queries pass through the learned neural networks, multi-resolution histogram features are computed and fed into corresponding neural networks, followed by which hidden layers are extracted from networks of all levels and concatenated as final representations. The pipeline of the proposed framework is shown in Figure 1. We evaluate the performance of CDNN and its pyramid extension separately on the large-scale extended SHREC 2014 sketch-based 3D shape retrieval benchmark. We conclude the main contributions of this work are as follows:

- ⋆ We address the challenging sketch-based 3D shape retrieval problem with neural network-based approaches, which can learn discriminative and domain-invariant representations for sketches and 3D shapes while does not require strong restrictions (*e.g.*, the correspondence infor-

*: Corresponding Author (Email: yfang@nyu.edu)

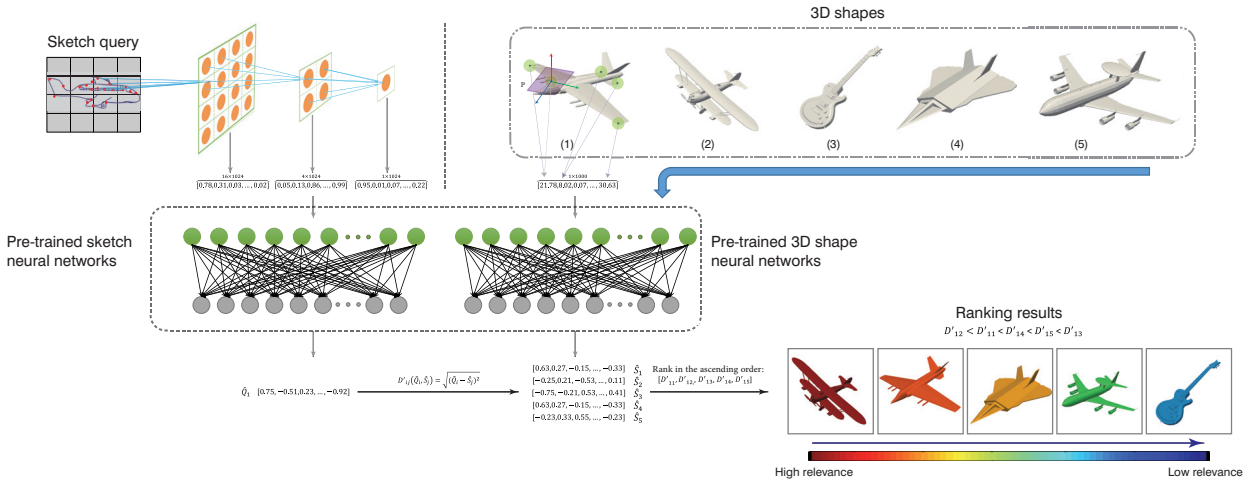


Figure 1: Illustration of the pipeline of our proposed sketch-based 3D shape retrieval framework.

mation[1]) for the training data.

- ⋆ We propose a CDNN approach, and extend CDNN to cooperate with a hierarchical structure (PCDNN), so that multi-resolution sketch histograms can be mapped to 3D shapes at increasingly fine resolutions.
- ⋆ Experimental results suggest both CDNN and PCDNN can achieve state-of-the-art performance on the large-scale extended SHREC 2014 benchmark, while PCDNN can further lead to a dramatic performance improvement over CDNN.

Related Works

A large number of works (Fang et al. 2015; Xie et al. 2015a; Novotni and Klein 2003; Xie et al. 2015b) have been proposed to address the 3D shape retrieval problem using 3D shape queries. Compared with these works, the sketch-based 3D shape retrieval (Li et al. 2012; 2013; 2014a) problem is closer to practical applications and more challenging. Li *et al.* (Li et al. 2014a) conduct comprehensive comparisons between six state-of-the-art sketch-based 3D shape retrieval methods from four participating groups on the SHREC 2014 benchmark report, where the overlapped pyramid of histograms of oriented gradients (OPHOG) and the similarity constrained manifold ranking (SCMR) methods proposed by Tatsuma's group lead the best performance on the scoreboard.

Existing sketch-based 3D shape retrieval techniques can be categorized in two groups: 1) approaches (Liu et al. 2008; Liu, Chen, and Tang 2011; Wang et al. 2009; Leclerc and Fischler 1992) that intend to inflate sketch drawings to 3-dimensional space according to heuristic rules (*e.g.,* line parallelism, polyhedron symmetry, corner orthogonality, *etc*); 2) approaches (Yoon et al. 2010; Pu and Ramani 2005; Pu, Lou, and Ramani 2005; Li et al. 2014b) that aim to directly alleviate the divergence between heterogeneous domains by learning domain invariant metrics.

Sketch-based 3D shape retrieval remains a challenging task in the community due to three reasons: 1) there does not exist a recognized representation for sketch data; 2) establishing the matches from the sketch domain to the 3D shape domain or vice versa is difficult; 3) many previous approaches that use 2D rendered projection images for representing 3D shapes suffer from the view variance and deformation problem. As solutions to these difficulties, 1) we, as the first attempt, apply the Sparse Coding Spatial Pyramid Matching (ScSPM) (Yang et al. 2009) feature, which has been a powerful handcrafted representation for regular images, for sketch representation; we propose a novel pyramid cross-domain neural networks architecture, which is motivated by the pyramid schemes (Lazebnik, Schmid, and Ponce 2006; Yang et al. 2009), for sketch-based retrieval; 3) we use the view-invariant Local Depth Scale-Invariant Feature Transform (LD-SIFT) (Darom and Keller 2012) feature for representing 3D shapes, and directly input the 3D shape features instead of 2D rendered images to the cross-domain neural networks. Wang *et al.* (Wang, Kang, and Li 2015) recently also propose a neural network-based method for sketch-based 3D shape retrieval. While they learn Convolutional Neural Networks (CNN) (Krizhevsky, Sutskever, and Hinton 2012) on the image level by projecting 3D shapes to 2D images, our approach directly learns the cross-domain mapping between 2D sketch features and 3D shape features. Though our network architecture is much simpler, our PCDNN approach can achieve state-of-the-art performance on the SHRECT14 dataset.

Pyramid Cross-Domain Neural Networks

Category-Specific Neural Networks

We consider $\boldsymbol{X} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^P\} \in \mathbb{R}^{K \times P}$ as the input K-dimensional feature, $+1$ as an intercept term and

[1]The correspondence information denotes the one-to one mapping between cross-domain instance pairs, and is required by many transfer learning methods.

$\{w^1, w^2, \cdots, w^K\}$ and b as neuron parameters. The output of such a neuron $h_{W,b}(\boldsymbol{x}^p)$ can be computed as:

$$h_{W,b}(\boldsymbol{x}^p) = f(\sum_{k=1}^{K} w_k x_k^p + b), \tag{1}$$

where the function $f(\cdot)$ is chosen as the sigmoid function:

$$f(z) = \frac{1}{1 + exp(-z)}, \tag{2}$$

which scales the output $f(z)$ to in the range $[0, 1]$. An activation function $f(\cdot)$ takes a vector as the input and outputs a value. For convenience purposes, the common practice is to extend $f(\cdot)$ to apply to vectors in an element-wise fashion (*i.e.*, $f([\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3]) = [f(\boldsymbol{z}_1), f(\boldsymbol{z}_2), f(\boldsymbol{z}_3)]$).

A neural network can be constructed by assembling a number of neurons. We consider a 3-layer neural network, which has an input layer, a hidden layer and a target layer. Let $\hat{\boldsymbol{X}} = \{\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \cdots, \hat{\boldsymbol{x}}^P\} \in \mathbb{R}^{K \times P}$ be the K-dimensional target values at the target layer, where P is the number of instances. When the input data $\boldsymbol{X}$ are fed into the neural network, the weights on each neuron are optimizing towards the minimum discrepancy between $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$. Once the weights are optimized, the hidden layer values $\boldsymbol{Y} = \{\boldsymbol{y}^1, \boldsymbol{y}^2, \cdots, \boldsymbol{y}^P\} \in \mathbb{R}^{N \times P}$ are extracted as the N-dimensional feature of the input sample. We aim to obtain a network that can demonstrate strong extrapolation capability for new query instances. Thus, we enforce discriminative constraints to the target layers by setting identical target vectors[2] to instances that come from the same class. An illustration of the category-specific neural network for 3D shapes is given in Figure 2. At the target end, all 3D shapes in the "airplane" category are mapped to the same vector, while all 3D shapes in the "cars" category are mapped to a different vector. The objective function for learning a category-specific neural network can be formulated as the square-loss function on the weights:

$$\arg\min_{\boldsymbol{W},\boldsymbol{b}} \frac{1}{P} \sum_{i=1}^{P} \|\hat{\boldsymbol{x}}^i - h_{\boldsymbol{W}^l,\boldsymbol{b}^l}(\boldsymbol{x}^i)\|_2^2 + \lambda \sum_{l=1}^{L} \|\boldsymbol{W}^l\|_F^2, \tag{3}$$

where $\boldsymbol{W} = \{\boldsymbol{W}^1, \boldsymbol{W}^2, \cdots, \boldsymbol{W}^L\}\mathbb{R}^{K \times L}$ is the neuron parameters of the neural network and L is the number of layers, λ is the balancing parameter and W^l is the weight vector at layer l.

Cross-Domain Neural Networks

We use the ScSPM (Yang et al. 2009) feature for representing sketches and the LD-SIFT (Darom and Keller 2012) feature for representing 3D shapes, and let $\boldsymbol{X}_s = \{\boldsymbol{x}_s^1, \boldsymbol{x}_s^2, \cdots, \boldsymbol{x}_s^{P_s}\} \in \mathbb{R}^{K_s \times P_s}$ and $\boldsymbol{X}_m = \{\boldsymbol{x}_m^1, \boldsymbol{x}_m^2, \cdots, \boldsymbol{x}_m^{P_m}\} \in \mathbb{R}^{K_m \times P_m}$ be the sketch features and 3D shape features respectively. In order to retrieve 3D shapes based on sketch queries, we aim to obtain low discrepancy between sketch and 3D shape representations. On the top of intra-domain discriminativity, the domain adaptivity can be achieved by further assuming sketches and 3D shapes that come from the same category possess identical representations at the target layer (*i.e.,* a pair of category-specific neural networks for both sketches and 3D shapes can be connected from the target layer). Let $q(\boldsymbol{x}^i)$ be the class label of an input instance i, the objective function for jointly learning CDNN for both sketches and 3D shapes can be formulated as:

$$\begin{aligned}
&\arg\min_{\boldsymbol{W_s},\boldsymbol{b_s}} \frac{1}{P_s} \sum_{i=1}^{P_s} \|\hat{\boldsymbol{x}}_s^i - h_{\boldsymbol{W_s},\boldsymbol{b_s}}(\boldsymbol{x}_s^i)\|_2^2 + \lambda \sum_{l=1}^{L} \|\boldsymbol{W_s}^l\|_F^2, \\
&\arg\min_{\boldsymbol{W_m},\boldsymbol{b_m}} \frac{1}{P_m} \sum_{j=1}^{P_m} \|\hat{\boldsymbol{x}}_m^j - h_{\boldsymbol{W_m},\boldsymbol{b_m}}(\boldsymbol{x}_m^j)\|_2^2 + \lambda \sum_{l=1}^{L} \|\boldsymbol{W_m}^l\|_F^2, \\
&\quad \text{s.t.} \quad \hat{\boldsymbol{x}}_m^i = \hat{\boldsymbol{x}}_m^j = \hat{\boldsymbol{x}}_s^i = \hat{\boldsymbol{x}}_s^j \\
&\quad \text{if} \quad q(\boldsymbol{x}_m^i) = q(\boldsymbol{x}_m^j) = q(\boldsymbol{x}_s^i) = q(\boldsymbol{x}_s^j),
\end{aligned} \tag{4}$$

where W_s, b_s, W_m and b_m are parameters of the sketch network and the 3D shape network respectively. As explained in Section , the target vectors $\hat{\boldsymbol{X}}_s = \{\hat{\boldsymbol{x}}_s^1, \hat{\boldsymbol{x}}_s^2, \cdots, \hat{\boldsymbol{x}}_s^{P_s}\} \in \mathbb{R}^{K_s \times P_s}$ and $\hat{\boldsymbol{X}}_m = \{\hat{\boldsymbol{x}}_m^1, \hat{\boldsymbol{x}}_m^2, \cdots, \hat{\boldsymbol{x}}_m^{P_m}\} \in \mathbb{R}^{K_m \times P_m}$ are predefined, and do not change over the optimization of Equation (4). Thus, optimizing CDNN can be seen as separately optimizing two independent neural networks. Once we obtain the optimum $\hat{\boldsymbol{W}}_s$, $\hat{\boldsymbol{b}}_s$, $\hat{\boldsymbol{W}}_m$ and $\hat{\boldsymbol{b}}_m$, neuron values in L_2 layers are extracted as the representations when sketch and 3D shape features pass through the networks.

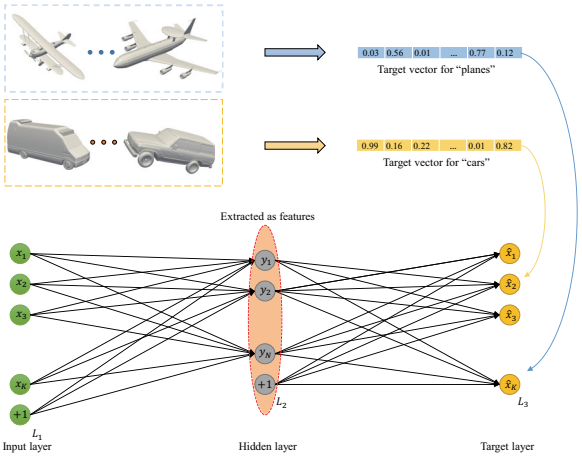


Figure 2: Illustration of the category-specific neural network for 3D shapes. Instances that come from the same class (*e.g.*, "airplanes" and "cars") are allocated with identical vectors at the target layer.

[2]In our implementation, these vectors are simply defined as random vectors. Experimental results suggest that setting the targets $\hat{\boldsymbol{X}}$ as random vectors can result in good performance. Our observations are also consistent with the results reported in (Zhang et al. 2013) and (Bingham and Mannila 2001).

Pyramid Cross-Domain Neural Networks

Due to the fact that sketches often present in noisy, incomplete and discontinuous natures, and that sketches and 3D shapes normally present in highly variant forms, we further propose a pyramid structure for learning multiple CDNN in a parallel fashion, so that the homogeneous information across both sketch and 3D shape domains can be mined more aggressively. We follow the ScSPM (Yang et al. 2009) framework to construct hierarchical representations of sketches. Regular SIFT (Lowe 1999) features are extracted from 16×16 patches of sketch images, and projected to a 1024-dimensional dictionary while constraining on the sparsity of projection coefficients. Max-pooling is applied to the sparse codes at 3 layers, which divide a sketch image into 1×1, 2×2 and 4×4 patches. Instead of directly concatenating the pooling results of all patches and feeding a global $(1 \times 1 + 2 \times 2 + 4 \times 4) \times 1024 = 21504$-dimensional feature to the neural networks (*i.e.,* the CDNN approach), we extract pooling results at each pyramid level and particularly train a CDNN that describes the connections between 3D shape features and sketch features at this level. When sketch and 3D shape features pass through PCDNN, the values at the hidden layer of each pyramid network are extracted and concatenated as the final representations for sketches and 3D shapes respectively (as shown within the red dashed rectangulars).

Optimization

Obtaining optimum parameters a the neural network is a regression problem. Since above neural networks can be optimized in the same manner, we use the notations in Section without loss of generality. Let R denote the number of nodes in a layer, we define

$$J(\boldsymbol{W}, \boldsymbol{b}) = \frac{1}{P} \sum_{j=1}^{P} \|\hat{\boldsymbol{x}}^j - h_{\boldsymbol{W},\boldsymbol{b}}(\boldsymbol{x}^j)\|_2^2 + \lambda \sum_{l=1}^{L} \sum_{j=1}^{R} \sum_{k=1}^{R+1} \|W_{jk}^l\|^2. \tag{5}$$

Since $J(W_s, b_s)$ is a non-convex function, computing a minimum $J(W_s, b_s)$ using the gradient descent solution suffers the local optima problem. However, practical experience suggests that gradient descent is still an applicable approach for solving this problem (Ng 2011). In each iteration of gradient descent, the parameters W_s and b_s are updated by the partial derivatives of $J(W_s, b_s)$, which can be described as:

$$\begin{aligned} \frac{\partial}{\partial W_{jk}^l} J(\boldsymbol{W}, \boldsymbol{b}) &= \left[\frac{1}{P} \sum_{j=1}^{P} \frac{\partial}{\partial W_{jk}^l} J(\boldsymbol{W}, \boldsymbol{b}, x_k^j, \hat{x}_k^j) \right] + \lambda W_{jk}^l \\ \frac{\partial}{\partial b_j^l} J(\boldsymbol{W}, \boldsymbol{b}) &= \frac{1}{P} \sum_{j=1}^{P} \frac{\partial}{\partial b_j^l} J(\boldsymbol{W}, \boldsymbol{b}, x_k^j, \hat{x}_k^j) \end{aligned} \tag{6}$$

The backpropagation algorithm (Werbos 1990), (Bengio 2009) can be applied for computing the partial derivatives efficiently. We first compute the "forward" activations through the neural network based on the descriptions in Section , and store the output of the sigmoid function as $a^l = h_{W^l,b^l}(X)$ at a neuron in layer l. Then, the error of each node j in layer l can be computed based on the node error in layer $l+1$:

$$\delta_j^l = \left(\sum_{k=1}^{R+1} W_{jk}^l \delta_k^{l+1} \right) f'(\sum_{k=1}^{R} w_k^l x_k + b_k^l), \tag{7}$$

such that the partial derivatives can be computed as:

$$\begin{aligned} \frac{\partial}{\partial W_{jk}^l} J(\boldsymbol{W}, \boldsymbol{b}) &= a_k^l \delta_j^l \\ \frac{\partial}{\partial b_j^l} J(\boldsymbol{W}, \boldsymbol{b}) &= \delta_j^l. \end{aligned} \tag{8}$$

Since the sigmoid function is chosen, the derivative function $f'(z) = f(z)(1 - f(z))$. At the target layer, the error δ_j^L can be directly measured from the value at the output neuron j and its corresponding regression value using:

$$\begin{aligned} \delta_j^L &= \frac{\partial}{\partial z_j^L} \frac{1}{2} \|\hat{\boldsymbol{x}}^j - h_{\boldsymbol{W},\boldsymbol{b}}(\boldsymbol{x})\|^2 \\ &= -(\hat{x}^j - a_j^L) f'(\sum_{k=1}^{R} w_k^L x_k + b_k^L). \end{aligned} \tag{9}$$

When the gradient decent algorithm converges or a maximum iteration number M is reached, the approximately optimal parameters W^* and b^* can be obtained.

Retrieval

We denote $\hat{\boldsymbol{Y}}_q$ as encoded query features and $\hat{\boldsymbol{Y}}_s$ as encoded 3D shape features. In order to rank 3D shapes according to query sketches, we compute the dissimilarity matrix D' based on $\hat{\boldsymbol{Y}}_q$ and $\hat{\boldsymbol{Y}}_s$ based using the Euclidean distance.

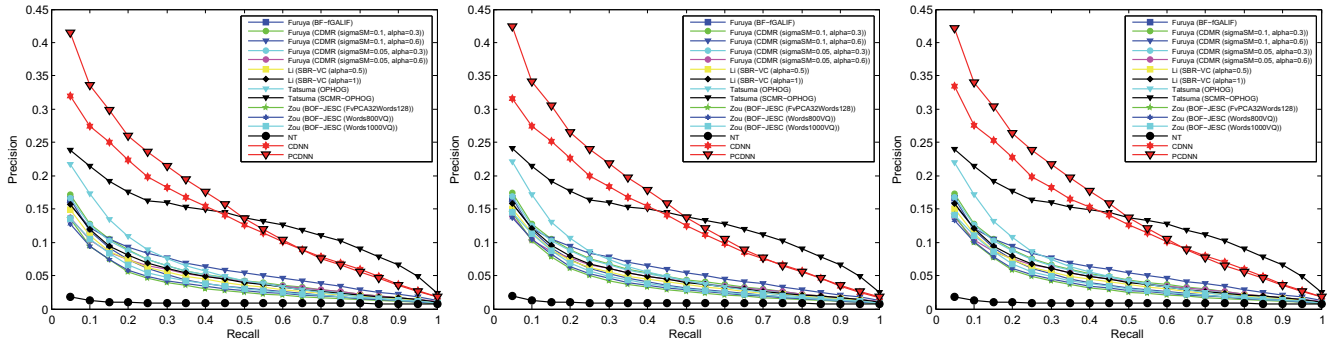
$$D'_{ij}(\hat{\boldsymbol{Y}}_q, \hat{\boldsymbol{Y}}_s) = \sqrt{(\hat{\boldsymbol{Y}}_q^i - \hat{\boldsymbol{Y}}_s^j)^2}, \tag{10}$$

Then, ranking can be carried out based on the ascend order of each row of the dissimilarity matrix, *i.e.*, the lower the entry value D'_{ij} is, the more relevant the 3D shape $\hat{\boldsymbol{Y}}_s^j$ and the query sketch $\hat{\boldsymbol{Y}}_q^i$ are.

Experiments

Dataset and Settings

The proposed methods are evaluated on the large-scale extended SHREC 2014 sketch-based 3D shape retrieval benchmark (Li et al. 2014a). The benchmark contains $13,680$ sketches and $8,987$ 3D shapes from 171 classes. The number of sketches in each class equals to 80, and the number of 3D shapes in each class varies from 1 to 632. The sketches are further split into the training part and the testing part, which contain 8550 and 5130 sketches respectively. We strictly follow the experimental settings in (Li et al. 2014a) and report the performance of our proposed methods by using the training dataset, the testing dataset and the complete benchmark as queries respectively. The sketch data used for training does not overlap with the data used for testing, and



(a) Testing dataset (b) Training dataset (c) Complete dataset

Figure 3: Precision-Recall plot performance comparisons on the extended large-scale SHREC'14 benchmark.

the 3D shape data are always the same in all runs. The results reported on the complete benchmark are obtained by 10-fold cross-validation, that the sketch data are split into 10 partitions (*i.e.*, 8 out of 80 sketches are selected from each category in each partition), and one partition is used as testing while the remaining are used as training in each of ten rounds.

LD-SIFT (Darom and Keller 2012) features are extracted from 3D shapes based on the Difference of Gaussians (DoG)-based detector (Darom and Keller 2012). In order to obtain global representations of the 3D shapes, the Bag-of-Words (BoW) paradigm is applied to the local LD-SIFT features by projecting these features onto a 1024-dimensional dictionary. Sketches are represented by the ScSPM (Yang et al. 2009) model, where the feature dimensions for the bottom level, the middle level and top level are 16384, 4096 and 1024, respectively. In order to fix the same feature dimension for inputs of all levels, Principal Component Analysis (Moore 1981) is applied to features at the bottom and middle level. We also fix the feature dimensions for the hidden layer and target layer as 1024. For the single-level CDNN approach, PCA is applied to the concatenated 21504-dimensional features to 1024 dimensions.

The restricted Boltzmann machine (RBM) (Hinton and Salakhutdinov 2006) is used for pre-training. The numbers of iterations for RBM and backpropagation are set as 50 and 500, respectively, and the balancing parameter λ is set as 0.001.

Experimental Results and Discussions

The 7 commonly used evaluation metrics, Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), E-Measure (E), Discounted Cumulated Gain (DCG) and Average Precision (AP) (Shilane et al. 2004), are used for evaluating the performance of the proposed methods. Considering the unbalanced number of 3D shapes within different classes, we follow (Li et al. 2014a) and adopt the reciprocally weighted evaluation metric for performance comparisons in our experiment, where lower weights are assigned to categories that contain less available 3D shapes. We compare with the-state-of-the-arts methods, Bag-of-Features of Dense SIFT

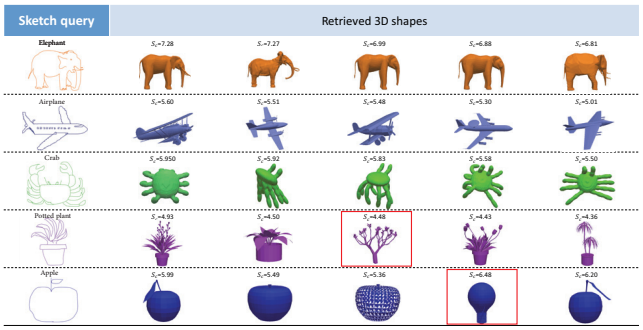


Figure 4: Illustration of the top-5 3D shape retrieval results for some sketch queries in different categories. The scores on the top of retrieved 3D shapes denote confidence levels of retrieval according to query sketches. Each red rectangular denotes a false positive retrieval.

(BF-DSIFT), Cross-Domain Manifold Ranking (CDMR), Shape Context Matching (SBR-VC), Overlapped Pyramid of Histograms of Oriented Gradients (OPHOG), Similarity Constrained Manifold Ranking-Overlapped Pyramid of Histograms of Oriented Gradients (SCMR-OPHOG) and Bag-of-Features Junction-based Extended Shape Context (BOF-JESC) (Li et al. 2014a). The performance comparisons of the training dataset, the testing dataset and the complete benchmark under the weighted evaluation metric are given in Table 1. Due to the space limit, we only show the best results of methods summarized in (Li et al. 2014a). Complete comparisons with different parameter settings of these methods are given by the Precision-Recall (PR)-curves in Figure 3.

In order to demonstrate the effectiveness of these transfer learning techniques, we also show the retrieval results when transfer learning techniques are not applied to low-level sketch and 3D shape features (NT). Experimental results suggest both CDNN and PCDNN can achieve the-state-of-the-arts performance, and PCDNN can further achieve significant improvements over CDNN when multiple neural networks are trained. Since the PR-curves for both the common and weighted evaluation metrics are the same, only

Table 1: Reciprocally weighted performance metrics comparison on different datasets of the extended large-scale SHREC'14 benchmark for the Query-by-Sketch retrieval.

Contributor	Method	NN	FT	ST	E	DCG	AP
Training dataset				$1.0e-05*$			
Furuya	BF-fGALIF	0.435	0.274	0.414	0.175	2.038	0.344
	CDMR ($\sigma_{SM}=0.05, \alpha=0.3$)	0.442	0.301	0.454	0.201	2.055	0.369
Li	BSBR-VC ($\alpha=1$)	0.259	0.145	0.267	0.164	1.868	0.198
Tatsuma	SCMR-OPHOG	0.526	0.399	0.615	0.318	2.173	0.490
Zou	BOF-JESC (Words800VQ)	0.334	0.149	0.260	0.137	1.884	0.221
Ours	NT	0.003	0.003	0.005	3.754	0.003	0.006
	CDNN	0.876	0.582	0.905	0.529	**5.257**	0.762
	PCDNN	**2.026**	**1.261**	**1.704**	**0.783**	4.638	**1.493**
Testing dataset				$1.0e-05*$			
Furuya	BF-fGALIF	0.802	0.520	0.735	0.289	3.408	0.596
	CDMR ($\sigma_{SM}=0.05, \alpha=0.3$)	0.789	0.526	0.773	0.330	3.430	0.626
Li	BSBR-VC ($\alpha=1$)	0.449	0.264	0.425	0.264	3.051	0.291
Tatsuma	SCMR-OPHOG	0.993	0.743	1.035	0.541	3.676	0.886
Zou	BOF-JESC (Words800VQ)	0.462	0.271	0.467	0.236	3.149	0.370
Ours	NT	0.195	0.049	0.009	6.077	0.004	0.009
	CDNN	2.515	1.658	2.411	1.318	9.882	2.075
	PCDNN	**5.175**	**3.285**	**4.406**	**2.056**	**12.39**	**3.960**
Complete benchmark				$1.0e-05*$			
Furuya	BF-fGALIF	0.283	0.180	0.265	0.109	1.275	0.218
	CDMR ($\sigma_{SM}=0.05, \alpha=0.3$)	0.284	0.192	0.286	0.125	1.285	0.232
Li	BSBR-VC ($\alpha=1$)	0.164	0.094	0.164	0.101	1.159	0.118
Tatsuma	SCMR-OPHOG	0.345	0.260	0.386	0.200	1.366	0.316
Zou	BOF-JESC (Words800VQ)	0.196	0.097	0.167	0.087	1.179	0.138
Ours	NT	0.007	0.203	0.273	0.123	3.249	0.471
	CDNN	0.954	0.600	0.918	0.501	**3.706**	0.734
	PCDNN	**1.457**	**0.911**	**1.229**	**0.567**	3.375	**1.084**

one group of PR-curves for the training dataset, the testing dataset and the complete benchmark are plotted. We can observe that the PR-curve of PCDNN leads a large margin over the PR-curve of SCMR-OPHOG approach at lower recall rates (approximately less than 0.4), while the former drops below the latter at higher recall rates. This nature shows the superiority of CDNN and PCDNN for highly ranked documents. On the other hand, the larger performance improvements of CDNN and PCDNN in terms of NN, FT, ST, E and DCG scores, which measure the precisions of top ranked results instead of the whole ranking list, can also validate this. Compared to the unweighted performance metric, CDNN and PCDNN achieve more remarkable performance improvements under the reciprocally weighted metric. The higher precession values of SCMR-OPHOG at higher recall levels denote better performance when including some less highly ranked documents, however, users are more interested in highly ranked documents for retrieval tasks. Most importantly, our methods achieve higher overall scores under all criterion. The retrieval results of 5 categories in the complete benchmark are also given in Figure 4, where the score on the top of each retrieved 3D shape denotes the relative confidence according to the sketch query and each red rectangular denotes a false positive instance.

Conclusion and Future Work

In this work, we addressed the challenging sketched-based 3D shape retrieval problem with neural network-based approaches. By jointly learning a pair of category-specific neural networks while allocating identical target vectors at the target layers of both networks, sketches and 3D shapes can present in a unified feature space with reduced magnitudes of cross-domain discrepancies when their original low-level features pass through the neural network pair. We further developed a hierarchical learning paradigm, PCDNN, which maps sketch images to 3D shapes using multiple CDNN at different levels, so that the homogeneous information across both domain can be mined in a more aggressive fashion. Since both CDNN and PCDNN do not require any correspondence information during the learning process, they are superior to many other transfer learning methods, and can be easily generalized. We evaluated the effectiveness of both CDNN and PCDNN on the extended large-scale SHREC 2014 benchmark. Sufficient experimental results suggest both methods can achieve the state-of-the-art performance, and PCDNN can further improve CDNN when cooperating with the pyramid learning structure.

In the future work, we plan to investigate if the performance can be improved by cooperating with deep learning architectures (*e.g.*, increasing the number of neural network layers), or when increasing the number of pyramid levels. Also, we plan to investigate the possibility of employing an existing three-dimensional CNN that operates on 3D shapes while shares an identical softmax layer with a two-dimensional CNN that operates on sketches.

References

Bengio, Y. 2009. Learning deep architectures for ai. *Foundations and trends® in Machine Learning* 2(1):1–127.

Bingham, E., and Mannila, H. 2001. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 245–250.

Darom, T., and Keller, Y. 2012. Scale-invariant features for 3-d mesh models. *IEEE Transactions on Image Processing* 21(5):2758–2769.

Fang, Y.; Xie, J.; Dai, G.; Wang, M.; Zhu, F.; Xu, T.; and Wong, E. 2015. 3d deep shape descriptor. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2319–2328.

Gunn, T. G. 1982. The mechanization of design and manufacturing. *Scientific American* 247(3):114–30.

Hinton, G. E., and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *Science* 313(5786):504–507.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

Lazebnik, S.; Schmid, C.; and Ponce, J. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2169–2178.

Leclerc, Y. G., and Fischler, M. A. 1992. An optimization-based approach to the interpretation of single line drawings as 3d wire frames. *International Journal of Computer Vision* 9(2):113–136.

Li, B.; Schreck, T.; Godil, A.; Alexa, M.; Boubekeur, T.; Bustos, B.; Chen, J.; Eitz, M.; Furuya, T.; Hildebrand, K.; et al. 2012. Shrec'12 track: Sketch-based 3d shape retrieval. In *3DOR*, 109–118.

Li, B.; Lu, Y.; Godil, A.; Schreck, T.; Aono, M.; Johan, H.; Saavedra, J. M.; and Tashiro, S. 2013. Shrec'13 track: large scale sketch-based 3d shape retrieval. In *Proceedings of the Sixth Eurographics Workshop on 3D Object Retrieval*, 89–96. Eurographics Association.

Li, B.; Lu, Y.; Li, C.; Godil, A.; Schreck, T.; Aono, M.; Burtscher, M.; Fu, H.; Furuya, T.; Johan, H.; et al. 2014a. Shrec14 track: Extended large scale sketch-based 3d shape retrieval. In *Eurographics Workshop on 3D Object Retrieval 2014 (3DOR 2014)*, 121–130.

Li, B.; Lu, Y.; Godil, A.; Schreck, T.; Bustos, B.; Ferreira, A.; Furuya, T.; Fonseca, M. J.; Johan, H.; Matsuda, T.; et al. 2014b. A comparison of methods for sketch-based 3d shape retrieval. *Computer Vision and Image Understanding* 119:57–80.

Liu, J.; Cao, L.; Li, Z.; and Tang, X. 2008. Plane-based optimization for 3d object reconstruction from single line drawings. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 30(2):315–327.

Liu, J.; Chen, Y.; and Tang, X. 2011. Decomposition of complex line drawings with hidden lines for 3d planar-faced manifold object reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(1):3–15.

Lowe, D. G. 1999. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, volume 2, 1150–1157.

Moore, B. 1981. Principal component analysis in linear systems: Controllability, observability, and model reduction. *Automatic Control, IEEE Transactions on* 26(1):17–32.

Ng, A. 2011. Sparse autoencoder. *CS294A Lecture notes* 72.

Novotni, M., and Klein, R. 2003. 3d zernike descriptors for content based shape retrieval. In *ACM symposium on Solid modeling and applications*, 216–225.

Pu, J., and Ramani, K. 2005. A 3d model retrieval method using 2d freehand sketches. In *Computational Science–ICCS 2005*. Springer. 343–346.

Pu, J.; Lou, K.; and Ramani, K. 2005. A 2d sketch-based user interface for 3d cad model retrieval. *Computer-Aided Design and Applications* 2(6):717–725.

Shilane, P.; Min, P.; Kazhdan, M.; and Funkhouser, T. 2004. The princeton shape benchmark. In *Shape modeling applications, 2004. Proceedings*, 167–178.

Wang, Y.; Chen, Y.; Liu, J.; and Tang, X. 2009. 3d reconstruction of curved objects from single 2d line drawings. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1834–1841.

Wang, F.; Kang, L.; and Li, Y. 2015. Sketch-based 3d shape retrieval using convolutional neural networks. In *IEEE International Conference on Computer Vision*, 1876–1883.

Werbos, P. J. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE* 78(10):1550–1560.

Xie, J.; Fang, Y.; Zhu, F.; and Wong, E. 2015a. Deepshape: Deep learned shape descriptor for 3d shape matching and retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1275–1283.

Xie, J.; Zhu, F.; Dai, G.; and Fang, Y. 2015b. Progressive shape-distribution-encoder for 3d shape retrieval. In *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference*, 1167–1170. ACM.

Yang, J.; Yu, K.; Gong, Y.; and Huang, T. 2009. Linear spatial pyramid matching using sparse coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1794–1801.

Yoon, S. M.; Scherer, M.; Schreck, T.; and Kuijper, A. 2010. Sketch-based 3d model retrieval using diffusion tensor fields of suggestive contours. In *ACM International Conference on Multimedia*, 193–200.

Zhang, Y.; Shao, M.; Wong, E. K.; and Fu, Y. 2013. Random faces guided sparse many-to-one encoder for pose-invariant face recognition. In *IEEE International Conference on Computer Vision*, 2416–2423.