Yinjie Lei, Ziqin Zhou, Pingping Zhang, Yulan Guo, Zijun Ma, Lingqiao Liu

Deep point-to-subspace metric learning for sketch-based 3D shape retrieval

Pattern Recognition, 2019; 96:106981-1-106981-13

Published at: http://dx.doi.org/10.1016/j.patcog.2019.106981

Deep point-to-subspace metric learning for sketch-based 3D shape retrieval

Yinjie Lei[a], Ziqin Zhou[a,1], Pingping Zhang[b], Yulan Guo[c,d], Zijun Ma[a], Lingqiao Liu[e,*]

[a] *College of Electronics and Information Engineering, Sichuan University, Chengdu, China*
[b] *School of Information and Communication Engineering, Dalian University of Technology, Dalian, China*
[c] *School of Electronics and Communication Engineering, Sun Yat-sen University, Guangzhou, China*
[d] *College of Electronic Science and Engineering, National University of Defense Technology, Changsha, Hunan, China*
[e] *School of Computer Science, The University of Adelaide, Adelaide, Australia*

Abstract

One key issue in managing a large scale 3D shape dataset is to identify an effective way to retrieve a shape-of-interest. The sketch-based query, which enjoys the flexibility in representing the user's intention, has received growing interests in recent years due to the popularization of the touchscreen technology. Essentially, the sketch depicts an abstraction of a shape in a certain view while the shape contains the full 3D information. Matching between them is a cross-modality retrieval problem, and the state-of-the-art solution is to project the sketch and the 3D shape into a common space with which the cross-modality similarity can be calculated by the feature similarity/distance within. However, for a given query, only part of the viewpoints of the 3D shape is representative. Thus, blindly projecting a 3D shape into a feature vector without considering what is the query will inevitably bring query-unrepresentative information. To handle this issue, in this work we propose a Deep Point-to-Subspace Metric Learning (DPSML) framework to project a sketch into a feature vector and a 3D shape into a subspace spanned by a few selected basis feature vectors. The similarity between them is defined as the distance between the query feature vector and its closest point in the subspace by solving an optimization problem on the fly. Note that, the closest point is query-adaptive and can reflect the viewpoint information that is representative to the given query. To efficiently learn such a deep model, we formulate it as a classification problem with a

[*] Corresponding author

Email addresses: `yinjie@scu.edu.cn` (Yinjie Lei), `ziqinzhou@stu.scu.edu.cn` (Ziqin Zhou), `jssxzhpp@mail.dlut.edu.cn` (Pingping Zhang), `yulan.guo@nudt.edu.cn` (Yulan Guo), `mazijun@stu.scu.edu.cn` (Zijun Ma), `lingqiao.liu@adelaide.edu.au` (Lingqiao Liu)

[1] The second author has the equal contribution as the first author for this work.

special classifier design. To reduce the redundancy of 3D shapes, we also introduce a Representative-View Selection (RVS) module to select the most representative views of a 3D shape. By conducting extensive experiments on various datasets, we show that the proposed method can achieve superior performance over its competitive baseline methods and attain the state-of-the-art performance.

Keywords: sketch-based 3D shape retrieval, cross-modality discrepancy, representative-view selection, point-to-subspace distance

1. Introduction

With the rapid development of 3D sensing techniques, 3D shape data has received increasing research interests in the field of computer vision. Since the volume of 3D shape data grows significantly, shape retrieval has been becoming a crucial problem for 3D shape data management [1–6]. In its early year, a keyword is first labeled for each 3D shape, and is used as the query for retrieval [7, 8]. However, the keyword labeling is a time-consuming process, and is also impractical for the real-world applications, especially when dealing with large-scale datasets. Then, by using a 3D shape as query, considerable research has been devoted to the content-based 3D shape retrieval techniques. However, the acquisition of a query shape itself is difficult due to the nature of the 3D modality. Recently, the prevalence of touchscreen technologies (e.g., smart phones and tablet computers) enable the hand-drawing sketch a more convenient way for representing the user's intention. Compared with using a keyword or 3D shape as query, the sketch-based 3D shape retrieval is more straightforward and thus easier to be implemented in practical applications [9–12].

The hand-drawing sketches usually contain limited information and only reflect certain views of 3D shapes. As a result, obtaining a discriminative 3D shape features aiming to reduce the cross-modality discrepancy to sketch becomes a key issue. In order to extract 3D shape features, different 3D shape representations have been proposed. Recently, the point-cloud based [13–16] and the multi-view based [17–20] representations gradually become dominate choices. In particular, the multi-view based representations have achieved state-of-the-art performance so far [17–20]. For this type of representations, the 3D shape is initially rendered by a family of 2D views, as shown in Fig. 1. On top of that, one can then leverage the well-established 2D image deep models (e.g., AlexNet [21], VGG [22] and ResNet [23]), which are pre-trained on large-scale datasets

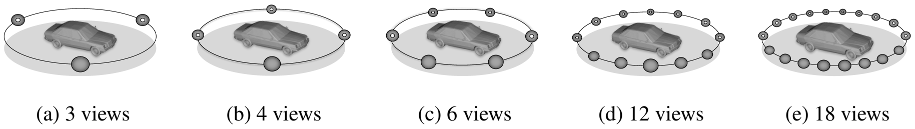
(a) 3 views (b) 4 views (c) 6 views (d) 12 views (e) 18 views

Figure 1: The camera array settings for the multi-view representation of a 3D shape.

(e.g., ImageNet [24]), for feature extraction.

Despite the promising prospect of the sketch-based 3D shape retrieval, there still exists three major challenges which have been hindering its development. First, the free-hand sketch drawing is a subjective activity, resulting in large variation among different individuals. Second, the sketch and 3D shape have a large cross-modality discrepancy, which makes it difficult to obtain modality-independent features. Third, the sketch usually reflects certain view of a 3D shape, and the visual appearance of different views may vary significantly. Aiming to handle these problems, the existing methods can be coarsely categorized into traditional descriptor based [2, 25] and deep-learned descriptor based [26, 27]. The first kind methods commonly apply the hand-crafted or shallow-learned features to describe both sketches and 3D shapes for similarity measurement. Nevertheless, it is difficult to design discriminative feature descriptors applied for both sketches and 3D shapes due to the large cross-modality discrepancy [11]. In contrast, the second kind methods, which are based on the deep-learned features are considered to be more robust and with more discriminative power. It can better accommodate the cross-modality discrepancy, and attain an improved retrieval accuracy.

As mentioned above, the query sketch is only representative to part views of a 3D shape, and the unrepresentative views offer minor contribution or even be harmful for retrieval. However, many existing methods [20, 28–30] treat all the views equally without considering the viewpoint information. In order to resolve this problem, we propose a Deep Point-to-Subspace Metric Learning (DPSML) framework. First, a Representative-View Selection (RVS) module is applied to obtain several most representative 3D shape views, and then a subspace spanned by feature vectors from the selected views is generated for describing a 3D shape. Later, the similarity between a sketch and a 3D shape is defined as the distance between the sketch feature vector and its closest point in the spanned subspace by solving an optimization problem on the fly. Note that, the closest point is query-adaptive and can reflect the viewpoint information captured by the query sketch. Moreover, in order to efficiently learn a deep model, we formulate the representation learning problem as a classification problem without the pairwise sample learning process used by many existing methods [29, 31]. In summary, the proposed DPSML is an end-to-end framework, and its effectiveness and robustness are extensively demonstrated by a set of experiments on three widely used benchmark datasets i.e., SHREC 2013, 2014 and 2016.

The rest of this paper is organized as follows. Section 2 describes the related works which are representative to the proposed method. Then, we give a method overview. Section 3 presents a detailed explanation of the proposed framework. Section 4 provides the details of the used benchmark datasets, evaluation metrics and the implementation details. The experimental results, comparisons to the state-of-the-arts along with a discussion are provided in Section 4. Finally, Section 6 concludes this work.

2. Related Works and Overviews

2.1. Related works

The work in [12, 32] provided a comprehensive survey and comparison of the sketch-based 3D shape retrieval methods. In the following, we restrain the review to the representative methods closely related to this work. More specifically, we cover the traditional sketch-based 3D shape retrieval methods e.g., hand-crafted or shallow-learned features and the deep-learned descriptors for the task of 3D shape retrieval in Subsection 2.1.1 and 2.1.2, respectively.

2.1.1. Traditional sketch-based 3D shape retrieval

In its early year, most existing sketch-based 3D shape retrieval methods rely on developing the modality-invariant features for matching between the sketch and the 3D shape. Eitz et al. [9] develops a Gabor local line based feature (GALIF) with a bag-of-features (BoF) framework for sketch-based 3D shape retrieval. In [33], a method based on view clustering (SBR-VC) and a parallel relative frame based shape context matching is proposed. Furuya and Ohbuchi [11] integrate the dense scale-invariant feature transform (SIFT) and BoF with a manifold ranking for matching similarity between sketch and 3D shape. In [33], the histogram of edge local orientations (HELO), histogram of oriented gradients (HOG) and Fourier descriptors are applied to describe sketches and 3D shapes. Then, the KD-tree with Manhattan distance is calculated as the cross-modality similarity measurement. An integrated descriptor ZFEC is designed in [12] for describing both sketches and 3D shapes. ZFEC combines Zernike moments, contour-based Fourier descriptor, eccentricity features and circularity features. Tatsuma et al. [32] propose a local improved pyramid of histogram of orientation gradients (iPHOG) and the similarity constrained manifold ranking (SCMR). Zhu et al. [34] apply the sparse coding spatial pyramid matching (ScSPM) for describing sketches and the view-invariant local depth scale-invariant feature transform (LD-SIFT) for 3D shapes. In [35], Yasseen et al. propose the chordal axis transform based shape descriptor and the dynamic time warping based matching framework for sketch-based 3D shape retrieval. In [36], the HOG-SIFT feature is applied to describe sketches and 3D shapes. Then, a sparse coding based matching method is used to perform retrieval. Li et al. [37] propose a semantic sketch-based 3D retrieval method using viewpoint entropy distribution for describing a 3D shape and an adaptive view clustering method.

Due to the limited discriminative power of the hand-crafted and shallow-learned features, the performance of the traditional sketch-based 3D shape retrieval methods is unsatisfactory.

2.1.2. Deep-learned 3D shape descriptors

In recent years, the deep neural networks have been successfully applied to many research fields, and achieved the state-of-the-art performance. The deep-learned descriptors for the 3D shapes are believed to be more complex, discriminative and with more generalization ability. For completeness, in this subsection we also include some works using a 3D shape rather than a sketch as query for 3D shape retrieval [18–20, 28–30, 38, 39]. The sketch-based 3D shape retrieval is a cross-modality matching task, which is considered to be more challenging than shape-based 3D shape retrieval. Nevertheless, we include both sketch-based and shape-based methods here since the two share some similarities in learning deep representations for 3D shapes.

In [31], the authors first select two representative views of a 3D shape. Then, a pair of Siamese convolutional neural networks are used, e.g., one for sketch and another for 3D shape. A loss function, composed of a within-modality term and a cross-modality term, is used to learn deep features for both sketches and 3D shapes. Su et al. [28] propose a multi-view CNN to learn discriminative features from the rendered views of a 3D shape. Then, a max-pooling operation is used to combine the obtained features to form a compact descriptor. Based on multi-view CNN, Bai et al. [38] propose a speeding-up mechanism to enable a real-time 3D shape retrieval. Xie et al. [17] introduce the Wasserstein barycenter learning to obtain a compact descriptor from the rendered views of 3D shapes. Their proposed barycenter is obtained by considering all the views of a 3D shape. In [40], an adversarial learning method is developed to train the transformation model between sketches and 3D shapes. The multi-views of a 3D shape is aggregated by an average view-pooling operation. Dai et al. [41, 42] propose a deep correlated metric learning model to mitigate the modality discrepancy between the sketches and 3D shapes. A discriminative loss and a correlation loss are defined to jointly train two deep nonlinear transformations to map the two modalities into a common feature space. Feng et al. [19] propose a group-view convolutional neural network (GVCNN) framework for hierarchical correlation modeling from the rendered views of a 3D shape to obtain a discriminative descriptor. Yu et al. [30] extract the effective 3D shape feature by aggregating local convolutional features from the rendered views of a 3D shape through bilinear pooling. They calculate the patches-to-patches similarity among different views rather than view-based pooling. He et al. [29] propose a triplet-center loss to learn the compact 3D shape descriptor from the rendered views. The resulted features are with more discriminative power than using traditional classification loss. Sarkar et al. [39] propose another perspective of view-generation for 3D shape, where it is represented by the multi-layered height-maps (MLH). Then, a novel view-merging method for combining view dependent information is proposed to obtain a compact descriptor. In [20], a combined features for 3D shapes are achieved based on both point-cloud and multi-view representations, and the resulted features are with more discriminative power. Based on the multi-view representation of 3D shapes, Kanezaki et al. [18] propose a CNN based model (RotationNet), which is learned in an unsupervised manner during the training phase. The resulted model can jointly estimate the pose and class label of a 3D shape.

Deep-learned 3D shape features have shown superior performance over the hand-crafted and shallow-learned features [17–20]. Nevertheless, most the multi-view based deep-learned 3D shape descriptors use a pooling scheme to equally fuse all rendered views into a compact descriptor. Only few works [19, 30] pay attention on the different discriminative power among the views.

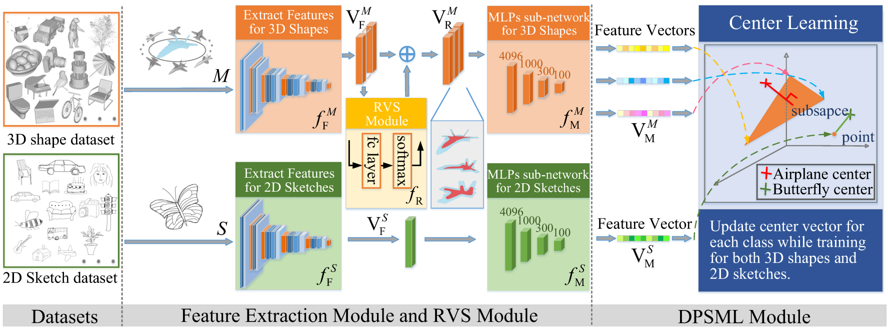
3D shape dataset
2D Sketch dataset
M
S
Extract Features for 3D Shapes
f_F^M
Extract Features for 2D Sketches
f_F^S
V_F^M
V_R^M
RVS Module
fc layer
softmax
f_R
V_F^S
MLPs sub-network for 3D Shapes
4096
1000
300
100
f_M^M
MLPs sub-network for 2D Sketches
f_M^S
Feature Vectors
V_M^M
Feature Vector
V_M^S
Center Learning
subsapce
point
Airplane center
Butterfly center
Update center vector for each class while training for both 3D shapes and 2D sketches.
Datasets
Feature Extraction Module and RVS Module
DPSML Module

Figure 2: The block diagram of the proposed DPSML framework. Our proposed model consists of two branches to learn the original features for sketches and 3D shapes separately via pre-trained deep models $f_{\mathbf{F}}^{M}$ and $f_{\mathbf{F}}^{S}$, which have the same structure without sharing weights. Then, the RVS module $f_{\mathbf{R}}$ is proposed to obtain fusion weights for rendered views of 3D shape and generate representative views. Next, two metric networks $f_{\mathbf{M}}^{M}$ and $f_{\mathbf{M}}^{S}$ are used to reduce the dimension of extracted features. Therefore, a sketch is described by a feature vector as a "point" in the representation space, while a 3D shape is spanned as a "subspace" by features vectors from the representative views. We randomly initialize a "virtual center" for each class in order to accelerate clustering in the training phase and develop the DPSML framework with a modified loss function. Note that, the distance from "point" to "subspace" is calculated by solving an optimization problem on the fly.

2.2. *Method overview*

We propose a novel framework, called Deep Point-to-Subspace Metric Learning (DPSML) for sketch-based 3D shape retrieval. Fig. 2 shows its block diagram, and the main steps are briefly described as follows. First, a 3D shape is represented by the a family of 2D views e.g., 12 views used in this work. The pre-trained deep models e.g., AlexNet, VGG and ResNet, are used to extract the original features for both sketches and 3D shapes. As a result, one feature vector is obtained for the sketch and 12 feature vectors for a 3D shape. Then, a Representative-View Selection (RVS) module is used to select the most representative views. Third, the DPSML framework is proposed to project a sketch to a point and the selected views of a 3D shape to a subspace. The similarity is defined by the distance between the sketch point and its closest point in the shape subspace. Note that, the resulted closest point is query-adaptive and can reflect the viewpoint information determined by the query sketch.

The main contributions of the proposed work can be summarized as follows:

- A RVS module is designed to identify the most representative views of a 3D shape for reducing the redundancy.

- The DPSML framework is proposed to calculate the query-adaptive similarity for sketch-based 3D shape retrieval.

- The representation learning problem is formulated as a classification problem, resulting in an efficient training process.

- A comprehensive experiments and comparisons are conducted on three large publicly available datasets, i.e. SHREC 2013, 2014 and 2016, to demonstrate the superior performance of the proposed method.

3. Methodology

As shown in Fig.2, the proposed framework mainly contains three modules. First, the feature extraction module is described in Subsection 3.1. Then, the details of the proposed RVS module are given in Subsection 3.2. Last, the detailed explanation of the DPSML framework is described in Subsection 3.3.

3.1. Feature Extraction

The proposed framework learns the sketch and 3D shape representations by solving a classification problem during the training phase. Specifically, we aim to build a shared classifier to identify a sketch or a 3D shape into its correct class, e.g., “Airplane”, “Chair” and etc. The classifier layer is shared to ensure the representation learned for both modalities are comparable and close within each class. More details about the shared classifier are introduced in Section 3.3. The sub-networks for obtaining the representations before the classification layer, however, i.e., not shared. Thus, we need to switch between these sub-networks according to its modality. Note that, the samples of the two modalities are randomly selected from the datasets, without any pairwise samples as input like some existing works [29, 31, 40–42].

Two branches $f_{\mathbf{F}}^{S}$ and $f_{\mathbf{F}}^{M}$ with the same pre-trained initialization weights (e.g., those obtained from the AlexNet, VGG or ResNet) are used to extract the original features for both sketches and the rendered views of 3D shapes. Note that, depending on the input modality, the two branches are fine-tuned separately. The dimension of the extracted feature is denoted as l (e.g., 4,096 for AlexNet and VGG11-16-19/25,088 for ResNet18-34/100,352 for ResNet50). Thus, the feature of a sketch is denoted as $\mathbf{v}_{\mathbf{F}}^{S} \in \mathbb{R}^{l}$, and $\mathbf{V}_{\mathbf{F}}^{M} \in \mathbb{R}^{l}$ for a 3D shape.

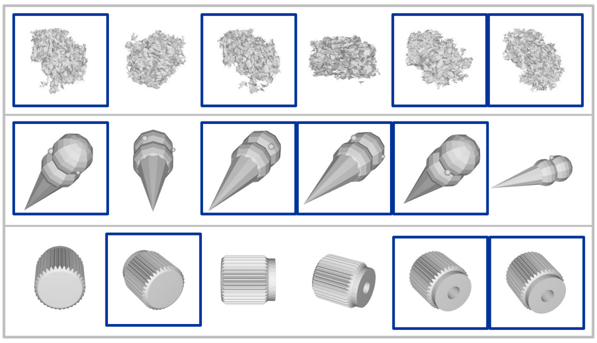

Figure 3: Some 2D views of 3D shapes e.g., bush, ice_cream_cone and wheel from 1st to 3rd rows respectively, where the framed images indicate views with similar appearance.

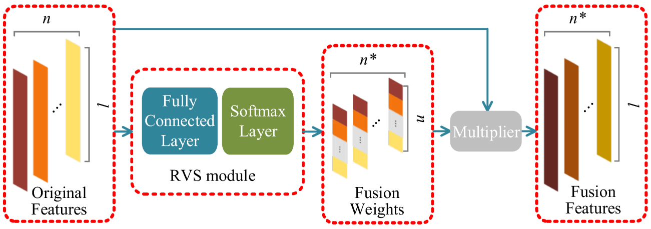

Figure 4: Structure of RVS module. n^* (less than n) vectors are obtained as fusion weights, which are applied on the original features to form n^* fusion feature vectors.

3.2. *Representation-view Selection Metric Learning*

As described above, n original feature vectors are obtained for a 3D shape. However, some views are redundant due to the their visual appearance similarity (see Fig. 3). In order to reduce the complexity of models, a Representative-View Selection (RVS) module $f_{\mathbf{R}}$ is introduced to eliminate such redundancy and results in an enhanced representation with n^* feature vectors, where n^* indicates the number of selected representative views and that is less than n. Specifically, $f_{\mathbf{R}}$ performs a weighted-sum-pooling operation (which works as a soft selection operator) by n^* times on the original view-based feature vectors as illustrated in Fig. 4. For each operation, the weights for sum-pooling are calculated by a dedicated attention function separately. The structure of $f_{\mathbf{R}}$ consists of a fully connection layer followed a soft-max layer.

For a given 3D shape, the input for RVS is its original features denoted as $\mathbf{V}_{\mathbf{F}}^{M} = [\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_n] \in \mathbb{R}^{l \times n}$. The output of the fully connected (linear) layer is a set of weights denoted by $\mathbf{A} \in \mathbb{R}^{n \times n^*}$ and can be calculated as:

$$\mathbf{A}_j = (\mathbf{w_R})_j^\top \mathbf{V}_\mathbf{F}^M + (b_\mathbf{R})_j \;\; j \in \{1, 2, \cdots n^*\} \tag{1}$$

where $\{\mathbf{w_R}, b_\mathbf{R}\}$ are the parameters of the fully connected layer. $\mathbf{A}_j \in \mathbb{R}^n$ is one column of $\mathbf{A}$. Each dimension of $\mathbf{A}_j$ indicates the importance of the corresponding view in the view of the j-th selector. $\mathbf{A}_j$ is then normalized by using the soft-max operator, formally:

$$\overline{\mathbf{A}}_j = softmax(\mathbf{A}_j) = \frac{\exp(\mathbf{A}_j)}{\sum_{t=1}^{n} \exp(\mathbf{A}(t, j))}, \tag{2}$$

where $\mathbf{A}(t, j)$ indicates the t-th dimension of $\mathbf{A}_j$. $\overline{\mathbf{A}}_j$ is the normalized selection weight which is used to perform weighted sum-pooling of $\mathbf{V}_\mathbf{F}^M$:

$$\mathbf{V}_\mathbf{R}^M(t, :) = \sum_{t=1}^{n} \mathbf{V}_\mathbf{F}^M(t, :) * \overline{\mathbf{A}}(t, j), \; j = 1, 2, \ldots n^* \tag{3}$$

where $\mathbf{V}_\mathbf{R}^M \in \mathbb{R}^{l \times n^*}$ is n^* feature vectors from the selected views by the RVS module.

In the above design, $\overline{\mathbf{A}}_j$ essentially acts as an anchor, where the original view features close to $\overline{\mathbf{A}}_j$ tend to have large inner product values and thus will be "selected" after the soft-max normalization. Note that, similar view features tend to have similar attention weights, and consequently they tend to be selected or de-selected by the same pooling operator which effectively merges the redundant features. Later, two MLPs sub-networks (i.e., $f_\mathbf{M}^S$ and $f_\mathbf{M}^M$), called **metric sub-networks**, are designed to extract high-level features for sketches and 3D shapes. Those two metric networks consist of a set of fully connection layers, the output dimension of the last layer is fixed to l^*, and $l^* = 100$ for fair comparison (see implementation details in Subsection 4.3). We denote the $\mathbf{v}_\mathbf{M}^S \in \mathbb{R}^{l^*}$ the final feature vector of a sketch and $\mathbf{V}_\mathbf{M}^M \in \mathbb{R}^{l^* \times n^*}$ is a set of feature vectors for a 3D shape. More specifically, a sketch is projected into a "point" and a shape is projected into a "subspace" spanned by a set of basis vectors.

3.3. Deep Point-to-Subspace Metric Learning

3.3.1 Distance as Similarity Score

When perform retrieving, we need to calculate the distance between a sketch and a 3D shape to rank the retrieval results. Since a sketch is described by a point and a 3D shape by a subspace, the distance between $\mathbf{v}_\mathbf{M}^S$ and $\mathbf{V}_\mathbf{M}^M$ can be defined as the closest distance between $\mathbf{v}_\mathbf{M}^S$ and a point in the subspace spanned by $\mathbf{V}_\mathbf{M}^M$, which can be formally calculated by the following equation:

$$\mathcal{D} = \min_{\mathbf{a}} \|\mathbf{v}^S_{\mathbf{M}} - \mathbf{V}^M_{\mathbf{M}}\mathbf{a}\|_2 \tag{4}$$

where $\mathbf{a} \in \mathbb{R}^{n^*}$ is the combination coefficients for basis feature vectors in $\mathbf{V}^M_{\mathbf{M}}$, and therefore$\mathbf{V}^M_{\mathbf{M}}\mathbf{a}$ represents the closest point to $\mathbf{v}^S_{\mathbf{M}}$ in the subspace. The RHS of the above equation is essentially an quadratic programming optimization problem and can be solved by:

$$\mathbf{a} = (\mathbf{V}^{M\top}_{\mathbf{M}}\mathbf{V}^M_{\mathbf{M}})^{-1}\mathbf{V}^{M\top}_{\mathbf{M}}\mathbf{v}^S_{\mathbf{M}}. \tag{5}$$

To avoid the possible numerical problem, we add an identity matrix $\alpha\mathbf{I}$ before taking the inverse operation as follows:

$$\mathbf{a} = (\mathbf{V}^{M\top}_{\mathbf{M}}\mathbf{V}^M_{\mathbf{M}} + \alpha\mathbf{I})^{-1}\mathbf{V}^{M\top}_{\mathbf{M}}\mathbf{v}^S_{\mathbf{M}}, \tag{6}$$

where $\mathbf{I}$ is an identity matrix and α is a constant.

3.3.2 Training Loss Function

With the distance defined in Eq. 4, one can use triplet loss [43] as the training loss function to encourage similar sketch-shape pairs to produce smaller distances than those are not paired. However, training with the triplet loss usually needs to carefully design a sample strategy to sample from a huge space of possible triplets and often results in a slow training process. Recent study shows that classification based loss [44] can achieve competitive results with much a simpler training step. The idea of this kind of methods is to convert the feature representation learning problem into a classification problem. For general feature learning/metric learning, we expect that the samples within the same class are similar to each other while being different from the samples in the other classes. The work in [44] shows that we can first train a deep network by a classification task and the learned representations before the classification layer can roughly satisfy the above desired property.

Our method is inspired by the center loss [44] but is different in **two aspects**: (1) instead of using a linear classifier which is inner-product-based, we adopt a distance-based classifier. (2) for the 3D shape part, the distance to the class center is calculated by solving a problem similar to Eq. 4. More specifically, the parameters of our classifier are a set of "virtual centers" for each class, denoted as $\mathbf{C} = [\mathbf{c}_1, ...\mathbf{c}_t, \ldots \mathbf{c}_k] \in \mathbb{R}^{l^* \times k}$, where k indicates the total number of classes in the training set and $\mathbf{C}$ is learned with the network parameters in an end-to-end fashion. Inside the classifier, the distances between a sketch representation and class centers or the distance between a 3D model representation and class centers are calculated as follows:

1) Each sketch is described by one feature vector $\mathbf{v}_{\mathbf{M}}^{S}$, and the distances between a sketch and all class centers $\mathbf{d}^{S}$ can be calculated as:

$$\mathbf{d}^{S} = [d_1^S, d_2^S, \dots d_k^S] \in \mathbb{R}^k, \;\; with \;\; d_t^S = ||\mathbf{v}^S - \mathbf{c}_t||_2 \tag{7}$$

2) Each 3D shape is described by a subspace spanned by $\mathbf{V}_{\mathbf{M}}^{M} = \{\mathbf{v}_1^M, \mathbf{v}_2^M, \dots \mathbf{v}_{n^*}^M\}$, the distances between shape and all class-centers $\mathbf{d}^{M}$ can be calculated as:

$$\mathbf{d}^{M} = [d_1^M, d_2^M, \dots d_k^M] \in \mathbb{R}^k \tag{8}$$

$$with \;\; d_t^M = \min_{\mathbf{a}} ||\mathbf{V}_{\mathbf{M}}^{M}\mathbf{a} - \mathbf{c}_t||_2 \tag{9}$$

Note that, the 3D shape is described by a subspace, its distance to each class center can be obtained similarly by solving an optimization problem as demonstrated in Subsection 3.3.

The loss function is supposed to minimize the mutual distance of samples falling into the same class and maximize the mutual distance for samples not belonging to the same class. In our method and other classification based representation methods, this requirement is approximated by minimizing the distance between a sample to its corresponding center and maximizing the distance between a sample to centers of other classes.

Specifically, we design a loss to encourage this property with two loss terms, that is, a relative distance loss and a absolute distance loss:

$$\mathcal{L} = \mathcal{H}(-\mathbf{d}, y) + \lambda \cdot \mathcal{G}(\mathbf{d}, y), \tag{10}$$

where $\mathbf{d} = \mathbf{d}^S$ or $\mathbf{d}^M$ and y represents the ground-truth class label. $\mathcal{H}$ is called relative distance loss and $\mathcal{G}$ is called absolute distance loss. Their definitions and roles are as follows:

- relative distance loss tends to maximize relative distance ratio between the distance to the true class center and the distance to other centers. It also works as a

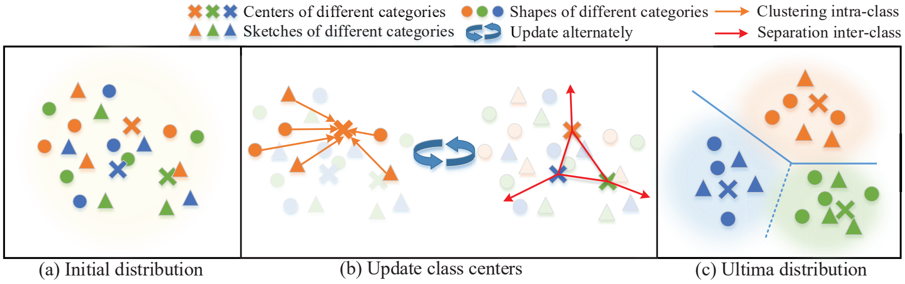

Figure 5: A toy illustration of the process of our overall loss function.

standard classification loss. We design it in a similar fashion as the cross entropy loss:

$$\mathcal{L}_1 = \mathcal{H}(-\mathbf{d}, y) = -log(\frac{\exp(-d_y)}{\sum_j \exp(-d_j)}), \tag{11}$$

where d_j indicates the distance to the j-th center.

- absolute distance loss aims to minimize the within-class distance and it works as a regularization term. It is defined as

$$\mathcal{L}_2 = \mathcal{G}(\mathbf{d}, y) = d_y, \tag{12}$$

in other words, we design to minimize the distance between a sample and its 285 corresponding center.

The different roles of the above two loss functions can be explained by Fig. 5. From which one can see that the relative distance loss makes the samples from different classes far from each other and the absolute distance loss makes the samples within the same class close to each other.

290 *3.3.3 Gradient Calculation*

To train the model with the proposed loss function. One needs to perform back propagation to the distance in Eq. 8, which involves calculating a gradient for a function expressed as an optimization problem. In other words, we need to calculate the following two terms:

$$\begin{gathered}\frac{\partial d(\mathbf{c},\theta)}{\partial \theta} = \frac{\partial \min_{\mathbf{a}} \|\mathbf{V}_{\mathbf{M}}^{M}(\theta)\mathbf{a} - \mathbf{c}\|_2^2}{\partial \theta} \\ \frac{\partial d(\mathbf{c},\theta)}{\partial \mathbf{c}} = \frac{\partial \min_{\mathbf{a}} \|\mathbf{V}_{\mathbf{M}}^{M}(\theta)\mathbf{a} - \mathbf{c}\|_2^2}{\partial \mathbf{c}},\end{gathered} \tag{13}$$

where θ represents the model parameters for generating the basis $\mathbf{V}^{M}_{\mathbf{M}}$ and $\mathbf{c}$ is a “virtual center” in the classifier layer. To calculate these gradients, we can first expand the distance as:

$$\begin{aligned} d(\mathbf{c},\theta) &= \min_{\mathbf{a}} \|\mathbf{V}^{M}_{\mathbf{M}}(\theta)\mathbf{a}-\mathbf{c}\|_2^2 \\ &= \min_{\mathbf{a}} \left(\mathbf{a}^\top \mathbf{V}^{M}_{\mathbf{M}}(\theta)^\top \mathbf{V}^{M}_{\mathbf{M}}(\theta)\mathbf{a} - 2\mathbf{a}^\top \mathbf{V}^{M}_{\mathbf{M}}(\theta)^\top \mathbf{c} + \mathbf{c}^\top \mathbf{c}\right) \\ &= \min_{\mathbf{a}} \left(\mathbf{a}^\top \mathbf{v}(\mathbf{c},\theta) - \frac{1}{2}\mathbf{a}^\top \mathbf{P}(\theta)\mathbf{a}\right) + \mathbf{c}^\top \mathbf{c}, \end{aligned} \tag{14}$$

where $\mathbf{v}(\mathbf{c},\theta) = -2\mathbf{V}^{M}_{\mathbf{M}}(\theta)\mathbf{c}$ and $\mathbf{P}(\theta) = -2\{\mathbf{V}^{M}_{\mathbf{M}}(\theta)\}^\top \mathbf{V}^{M}_{\mathbf{M}}(\theta)$. The derivative of Eq. 14 has been studied in [45]. According to the Lemma 2 in [45], the gradient can be calculated by first finding the optimal $\mathbf{a}$ and substituting the optimal solution to the objective function to drop out the “$\min$” operation, that is,

$$\begin{aligned} \frac{\partial d(\mathbf{c},\theta)}{\partial \theta} = \frac{\partial \min_{\mathbf{a}} \|\mathbf{V}^{M}_{\mathbf{M}}(\theta)\mathbf{a}-\mathbf{c}\|_2^2}{\partial \theta} = \frac{\partial \|\mathbf{V}^{M}_{\mathbf{M}}(\theta)\bar{\mathbf{a}}-\mathbf{c}\|_2^2}{\partial \theta} \\ \frac{\partial d(\mathbf{c},\theta)}{\partial \mathbf{c}} = \frac{\partial \min_{\mathbf{a}} \|\mathbf{V}^{M}_{\mathbf{M}}(\theta)\mathbf{a}-\mathbf{c}\|_2^2}{\partial \mathbf{c}} = \frac{\partial \|\mathbf{V}^{M}_{\mathbf{M}}(\theta)\bar{\mathbf{a}}-\mathbf{c}\|_2^2}{\partial \mathbf{c}}, \end{aligned} \tag{15}$$

where $\bar{a}$ is the solution of $\min_{\mathbf{a}} \|\mathbf{V}^{M}_{\mathbf{M}}(\theta)\mathbf{a}-\mathbf{c}\|_2^2$. In other words, we can obtain the solution $\bar{a}$ by solving the optimal function in forward calculation and then calculate the derivation of class center $\mathbf{c}$ and parameter θ as Eq. 15 in backward calculation.

3.3.4 Training Phase and Testing Phase

Note that the classifier layer can be discarded after the training phase. It is only used in the training phase to help learning the representation. Later, a sketch or a 3D shape will go through their respective feature extraction and subsequent modules to obtain the representations during the testing phase. The distance between a sketch query and a 3D shape will then be calculated by using Eq.4 .

4. Experimental Setups

In order to demonstrate the effectiveness of the proposed method, we evaluate it on three public benchmark datasets, i.e., the SHREC 2013 [12, 33], SHREC 2014 [32, 46] and SHREC 2016 [47]. We first introduce the experimental setups, including the details of benchmark datasets and the used evaluation metrics. Next, we present the implementation details of our framework. Then, we calculate all the metrics to investigate the performance and compare our results against the state-of-the-arts. Finally, we conduct more experiments to evaluate the effects of different modules as ablation study.

4.1. Sketch-based 3D Shape Retrieval Datasets

SHREC 2013 dataset is a large-scale benchmark to evaluate the sketch-based 3D shape retrieval methods. It contains 7,200 2D sketches and 1,258 3D shapes belonging to 90 classes, created by collecting from both the hand-drawing 2D sketch dataset [9] and the Princeton Shape Benchmark (PSB) dataset [1]. There are 80 sketches per class, where 50 sketches are used for training and the rest 30 sketches for testing. However, the number of 3D shapes per class is not equal (about 14 in average).

SHREC 2014 dataset is much larger than SHREC 2013. This dataset contains 13,680 sketches and 8,987 3D shapes from 171 classes, created by collecting from various datasets, e.g., SHREC 2012 [48], Toyohashi Shape Benchmark (TSB) [49]. SHREC 2014 dataset is very challenging due to the diversity of its classes, the unequal number of samples from different classes and large variations within class. For each class, there are 80 sketches, where 50 sketches are used for training and the rest for testing. While the number of 3D shapes for each class is not equal, ranging from 2 to 384.

SHREC 2016 dataset is a new benchmark and different from both SHREC 2013 and 2014 datasets due to the use of hand-drawing 3D sketches (i.e., from the Kinect300 dataset) as queries for 3D shape retrieval. The 3D sketches are collected by a Microsoft Kinect device, which contain 300 samples and are divided into 30 classes. Each class has 10 sketches, while 7 sketches are used for training and the rest for testing. Specifically, the 3D shapes come from SHREC 2013 dataset, which have 90 classes and 1,258 samples in total. Note that, only 21 classes of 3D sketches (i.e., 210 in total) have corresponding 3D shapes, while the remaining 9 classes are without corresponding 3D shapes. Therefore, 147 sketches from the above mentioned 21 classes are used for deep model training while the remaining 63 sketches used as testing set.

4.2. Evaluation Metrics

We follow the state-of-the-art to conduct experiments and with six widely-used metrics, e.g., Nearest Neighbor (NN), First Tier (FT), Second Tire (ST), E-Measure (E), Discounted Cumulative Gain (DCG) and mean Average Precision (mAP). We also report the Precision-Recall Curve (PR curve) to visually demonstrate the retrieval performance[2].

4.3. Implementation Details

The proposed method is implemented based on the open-source Pytorch 0.3.1 toolbox with the python 3.6 platform. The proposed deep model is trained and tested on a workstation with 4 NVIDIA Tesla M40 GPUs (each with 24G memory) and two E5-2650 CPU.

Data Prepossessing. The sketch images and the rendered views of a 3D shape from the SHREC 2013 and SHREC 2014 datasets are uniformly resized into a resolution of $224 \times 224 \times 3$ and subtracted the ImageNet mean [24]. Considering our method is developed based on the 2D sketch, we simply use the front view of the 3D sketch as input for evaluation.

Network Structures. For CNN sub-networks, we test different initialization from the pre-trained AlexNet [21], VGG19 [22] or ResNet34-50 [23]. Specifically, we use the layers of AlexNet before "fc7" layer (inclusive), the layers of VGG19 before "fc7" layer (inclusive) and the layers of ResNet34-50 before "pooling5" layer (inclusive). The MLPs sub-networks $f^S_{\mathbf{M}}$ and $f^M_{\mathbf{M}}$ are consisted of 3 fully connected layers (i.e., 4096-1000-300-100 for AlexNet/VGG19, 25088-1000-300-100 for ResNet34, and 100352-1000-300-100 for ResNet50), in which the weights are initialized using the "msra" method [50]. The "ReLU" activation function and batch normalization (BN) are adopted for all layers, and the standard Adam [51] is utilized as an optimizer during the training phase.

Parameter Settings. The maximum epoch number is set to 500. The initial learning rate is set to be 1×10^{-4} for the pre-trained CNN sub-networks, 1×10^{-3} for sub-networks $f^S_{\mathbf{M}}$, $f^M_{\mathbf{M}}$and RVS, while 1×10^{-2} for the DPSML sub-network. The learning rate decays by 10% after every 25 epochs. The balance hyper-parameter for loss function in Eq. 10 is set to $\lambda = 0.01$.

[2] An open-source code is used to calculate all these metrics is available at the website: https://userweb.cs.txstate.edu/~yl12/SBR2016/index.html

5. Experimental Results

5.1. Evaluation on the SHREC 2013 dataset

Our proposed method is based on an efficient point-to-subspace learning. In order to further improve the retrieval accuracy, a modified center learning method is used as part of the loss function. In order to demonstrate the effectiveness of RVS module, we compare the performance of the proposed method with different fusion operations, i.e., average pooling and FC-layer based feature. We also report the results with and without "center learning" method as described in the Subsection 3.3. Note that, "FC-layer based feature fusion" method concatenates the output vectors of the MLPs sub-networks and map to a final vectorized representation by using a fully connection layer aiming to reduce the dimension to 100. The combination of the above two factors creates more baseline methods in the revised manuscript, including:

- baseline 1: "average pooling" without "center learning"
- baseline 2: "FC-layer based feature fusion" without "center learning"
- baseline 3: "average pooling" with "center learning".
- baseline 4: "FC-layer based feature fusion" with "center learning".

A quantitative comparison is shown in Fig. 6 based on PR curves. It can be seen that the mAP of DPSML is higher than that of all baseline methods on the SHREC 2013 dataset. Note that, the same backbone of AlexNet is applied for both DPSML and baseline for original feature extraction. The results have verified the effectiveness of the proposed DPSML framework. Our DPSML method achieves a gain of 0.016 in terms of mAP than the best performance of baseline4 when AlexNet is applied as the CNN backbone.

Then, some examples of the retrieval results are shown in Fig. 7. The query sketches are listed on the left side (e.g., airplane, chair, bee, face, couch, potted_plant, guitar and car_sedan), and their retrieved top ten 3D shapes are listed on the right side according to their ranking order. The correct retrieved shapes are in gray color and the incorrect ones are in blue color. As shown in Fig. 7, the proposed method obtained promising retrieval results for the classes airplane, chair, bee, bicycle, couch, potted_plant, guitar and space_shuttle. However, the proposed method gives some incorrect retrieval results for the classes bee, space_shuttle due to only limited training samples are provided and bicycle, couch due to the appearance similarity with other classes.

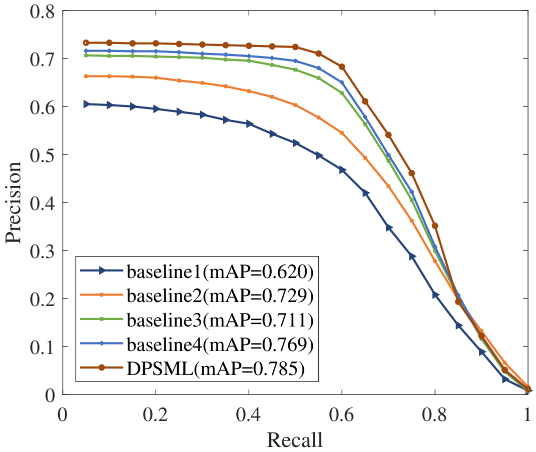

Figure 6: Comparison of baseline methods and DPSML of the sketch-based 3D shape retrieval performance on the SHREC 2013 dataset in terms of PR curves.

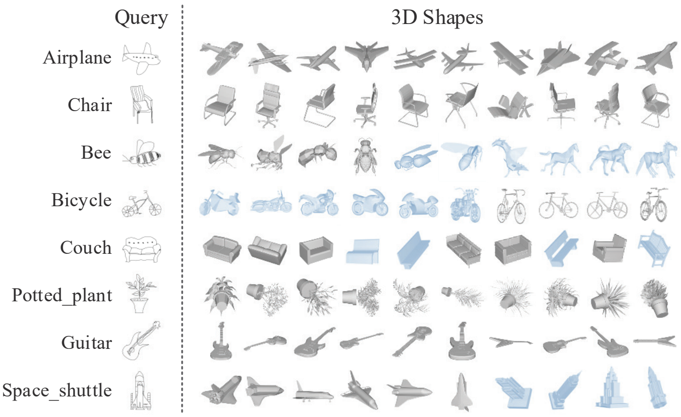

Figure 7: Some examples of retrieval results on the SHREC 2013 dataset. The query sketches are listed on the left and the retrieved 3D shapes are on the right. Note that, the corrected retrieved results are in gray color and the incorrect results are in blue color.

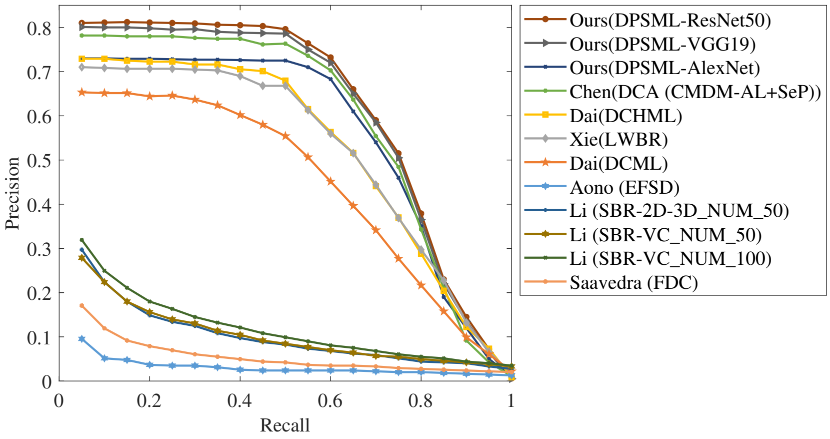

Figure 8: The PR curves of the proposed method as well as the state-of-the-arts on the SHREC 2013.

An illustration of PR curves of the proposed method and the state-of-the-arts on the SHREC 2013 dataset is presented in Fig. 8. It can be observed that the proposed method outperforms all the existing methods. When compared with the most recently published methods (e.g., [17, 40–42]), the proposed method achieves superior retrieval performance based on the same CNN backbones for original feature extraction (e.g., AlexNet, VGG19 and ResNet50).

Table. 1 provides a quantitative comparison of the proposed method with the state-of-the-arts on the SHREC 2013 dataset using the standard evaluation metrics. It can be seen that the proposed method achieves superior performance than the state-of-the-arts for all the evaluation measures. More specifically, the proposed method outperforms the best reported state-of-the-art method [42] with a gain of 0.011 on the most important metric NN when AlexNet is adopted. Furthermore, the proposed method also outperforms another most recent work [40] in terms of the NN (i.e., 0.819 versus 0.783), when a deeper neural network is applied (e.g., ResNet50). Experimental results on the SHREC 2013 dataset have clearly demonstrated the effectiveness of the proposed method.

Table 1: Quantitative comparison of the sketch-based 3D shape retrieval methods on the SHREC 2013 dataset. The best results are in bold font.

Methods	Backbones	NN	FT	ST	E	DCG	mAP
CDMR[11]	-	0.279	0.203	0.296	0.166	0.458	0.250
SBR-VC[33]	-	0.164	0.097	0.149	0.085	0.348	0.114
SP[52]	-	0.017	0.016	0.031	0.018	0.240	0.026
FDC[33]	-	0.110	0.069	0.107	0.061	0.307	0.086
Siamese[31]	-	0.405	0.403	0.548	0.287	0.607	0.469
CAT-DTW[35]	-	0.235	0.135	0.198	0.109	0.392	0.141
KECNN[53]	AlexNet	0.320	0.319	0.397	0.236	0.489	-
DCML[41]	AlexNet	0.650	0.634	0.719	0.348	0.766	0.674
LWBR[17]	AlexNet	0.712	0.725	0.785	0.369	0.814	0.752
DCHML[42]	AlexNet	0.730	0.715	0.773	0.368	0.816	0.744
DCA [40]	ResNet50	0.783	0.796	0.829	0.376	0.856	0.813
baseline1	AlexNet	0.604	0.582	0.692	0.341	0.735	0.620
baseline2	AlexNet	0.663	0.681	0.743	0.351	0.767	0.729
baseline3	AlexNet	0.689	0.680	0.762	0.369	0.795	0.711
baseline4	AlexNet	0.725	0.749	0.805	0.376	0.814	0.769
Ours (DPSML)	AlexNet	0.741	0.761	0.821	0.385	0.836	0.785
Ours (DPSML)	VGG19	0.801	0.816	0.852	0.398	0.870	0.831
Ours (DPSML)	ResNet34	0.813	0.826	0.864	0.406	0.883	0.846
Ours (DPSML)	**ResNet50**	**0.819**	**0.834**	**0.875**	**0.415**	**0.892**	**0.857**

5.2. Evaluation on the SHREC 2014 dataset

Compared with the SHREC 2013 dataset, the SHREC 2014 is more challenging since it contains more classes and larger variations within each class. The experimental results on the SHREC 2014 dataset can further demonstrate the performance of the proposed method. Following the same setups as on the SHREC 2013 dataset, we first evaluate the performance of our proposed method compared with four baseline methods, as shown in Fig. 9 in terms of PR curves. The DPSML significantly outperforms all the four baseline methods, and the mAP value has increased to 0.751 based on the AlexNet backbone. Our DPSML method achieves a gain of 0.085 in terms of mAP than the best performance of baseline4 when AlexNet is applied as the CNN backbone.

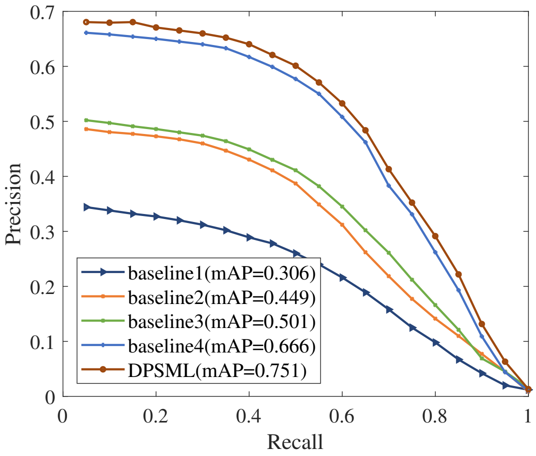

Figure 9: Comparison of baseline methods and DPSML of the sketch-based 3D shape retrieval performance on the SHREC 2014 dataset in terms of PR curves.

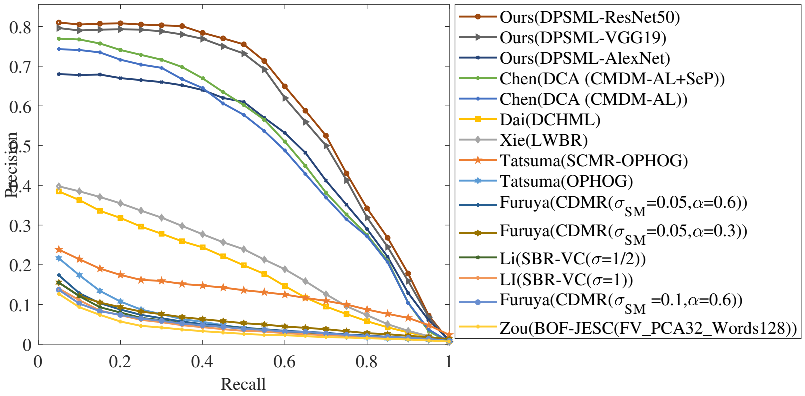

Figure 10: The PR curves of the proposed method as well as the state-of-the-arts on the SHREC 2014.

Fig. 10 demonstrates the PR curves of the proposed method and the comparison with the state-of-the-arts. It can be seen that, the precision value of the proposed method steadily exceeds the state-of-the-arts while the recall value increasing from 0 to 1. The methods with the closest performance to the proposed method are published in the work [40, 42], and the proposed method still exceeds the them with gains of 0.010 and 0.477 in terms of mAP respectively. The PR curves have verified the superior performance and robustness of the proposed method.

A comprehensive evaluation has been conducted for the proposed method and compared with the state-of-the-arts on the SHREC 2014 dataset. The corresponding quantitative comparison is provided in Table. 2, in terms of using the standard evaluation

Table 2: Quantitative comparison of the sketch-based 3D shape retrieval methods on the SHREC 2014 dataset. The best results are in bold font.

Methods	Backbones	NN	FT	ST	E	DCG	mAP
CDMR [11]	-	0.109	0.057	0.089	0.041	0.328	0.054
SBR-VC [46]	-	0.095	0.050	0.081	0.037	0.319	0.050
DB-VLAT [49]	-	0.160	0.115	0.170	0.079	0.376	0.131
CAT-DTW [35]	-	0.137	0.068	0.102	0.050	0.338	0.060
Siamese [31]	-	0.239	0.212	0.316	0.140	0.496	0.228
DCML [41]	AlexNet	0.272	0.275	0.345	0.171	0.498	0.286
LWBR [17]	AlexNet	0.403	0.378	0.455	0.236	0.581	0.401
DCHML [42]	AlexNet	0.403	0.329	0.394	0.201	0.544	0.336
DCA [40]	ResNet50	0.770	0.789	0.823	0.398	0.859	0.803
baseline1	AlexNet	0.386	0.294	0.404	0.201	0.556	0.306
baseline2	AlexNet	0.548	0.419	0.538	0.255	0.666	0.449
baseline3	AlexNet	0.555	0.479	0.575	0.276	0.678	0.501
baseline4	AlexNet	0.655	0.647	0.709	0.342	0.775	0.666
Ours (DPSML)	AlexNet	0.677	0.732	0.795	0.379	0.830	0.751
Ours (DPSML)	VGG19	0.748	0.785	0.839	0.406	0.866	0.800
Ours (DPSML)	ResNet34	0.757	0.789	0.832	0.402	0.863	0.800
Ours (DPSML)	**ResNet50**	**0.774**	**0.798**	**0.849**	**0.415**	**0.877**	**0.813**

metrics. For a fair comparison, we report the experimental results based on different CNN backbones (e.g., AlexNet, VGG19 and ResNet34-50) according to the published methods [40–42]. The proposed method outperforms the state-of-the-arts in all evaluation metrics. More specifically, the proposed method significantly exceeds the one of the most recently published methods [42] for the most important measure NN with a gain of 0.274 when AlexNet is adopted. Furthermore, the proposed method also outperforms another most recent work [40] in terms of the measure NN (i.e., 0.774 versus 0.770) when ResNet50 is applied. Nevertheless, our method turns the retrieval problem into the classification problem which can significantly reduce the training complexity. In contrast, most existing methods use the triplet or pairwise losses which requires a more complicated and time-consuming training process.

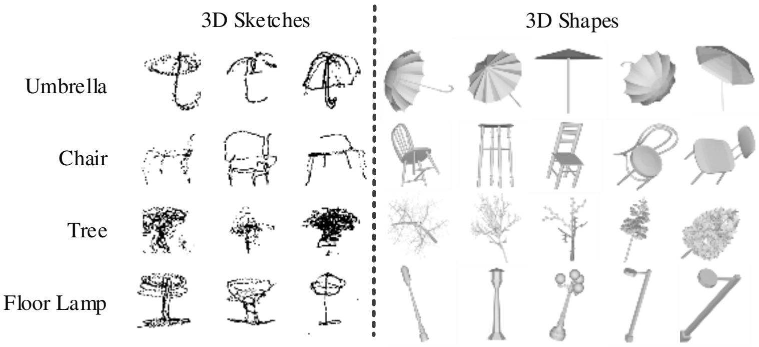

Figure 11: Some examples from the SHREC 2016 dataset. It can be observed that the 3D sketches are just sparse point-clouds.

5.3. Evaluation on the SHREC 2016 dataset

The SHREC 2016 dataset is a new 3D shape retrieval benchmark dataset which is different from both SHREC 2013 and 2014 datasets since it uses the 3D sketches as queries to retrieve 3D shapes. In fact, the 3D sketches are drawn with some sparse point-clouds, which are even more abstract than the 2D ones, as shown in Fig. 11.

Only a few previous works [42, 47] tested the SHREC 2016 dataset in their experiments. Therefore, with the consideration of a fair comparison of our method with results reported in the work [42], we use the front view image of 3D sketch as the query input. Table. 3 gives the quantitative comparison of the proposed method as well as the state-of-the-art 3D sketch to shape retrieval methods on the SHREC 2016 dataset using the standard evaluation metrics. It can be observed that the proposed method significantly outperforms the existing methods for all the metrics. Specifically, the value of the important measure "NN" obtained by the proposed method significantly exceeds the most recent work [42] with a gain of 0.312 based on the same CNN backbone of AlexNet. By using deeper CNN backbone (e.g., VGG19), the proposed method can intuitively achieve better performance. The performance of the proposed method on the SHREC 2016 dataset verified the superiority performance and generalization ability of the proposed method when it is extended to the task of 3D sketch–based 3D shape retrieval.

Table 3: Quantitative comparison of the 3D sketch to shape retrieval methods on the SHREC 2016 dataset. The best results are in bold font.

Methods	Backbones	NN	FT	ST	E	DCG	mAP
Siamese [31]	-	0.000	0.031	0.108	0.048	0.293	0.072
CNN-SBR [47]	-	0.222	0.251	0.320	0.286	0.471	0.314
DCHML [42]	AlexNet	0.117	0.106	0.148	0.086	0.327	0.147
Ours (DPSML)	AlexNet	0.429	0.478	0.563	0.279	0.609	0.499
Ours (DPSML)	**VGG19**	**0.476**	**0.510**	**0.572**	**0.290**	**0.640**	**0.533**

5.4. Ablation Study

We conduct more experiments to evaluate the effects of different modules of our method. For avoiding the over-fitting to the test dataset, we also randomly select 1/5 samples from the training set of the SHREC 2013 dataset as the validation (val) set for the following series of contrast tests on hyper-parameters. Note that, we only report the results using the AlexNet as backbone for original feature extraction and use the same hyper-parameters as described in Subsection 4.3 due to the space limitation. It is believed that, the other CNN backbones share similar performance trend.

Effects of number of rendered views of 3D shapes. In this experiment, we set different numbers of rendered views of 3D shapes in order to figure out the its effects on the performance. Specifically, the view numbers are set to 3, 4, 6 ,12, 18 and 36 by placing the "virtual cameras" every 120, 90, 60, 30, 20 and 10 degrees. Tab. 4 and Tab. 5show the corresponding quantitative comparison using the standard evaluation metrics. We notice that 12 rendered views perform best on the val set and the test set from SHREC 2013 dataset even the performance is decrease for the reduction of training samples. It is demonstrated that the retrieval results have improved while the number of rendered views increasing within a certain amount. However, slight decrease of the retrieval accuracy when the number is over 12 rendered views of a 3D shape are used. More rendered views lead to more representative power but less discriminative power for different categories while they are projected to high dimension space. In addition, more "representative views" are needed to combine more original views jointly. Besides, the increase of the view number will inevitably bring more computational burden, which can significantly affect the efficiency of a method. From the quantitative comparison, one can see that the number of 12 achieves the highest retrieval performance and reasonable

Table 4: Quantitative comparison of different numbers of rendered view on the testing set of SHREC 2013 dataset. The best results are in bold font.

Numbers	NN	FT	ST	E	DCG	mAP
3	0.688	0.679	0.316	0.140	0.496	0.228
4	0.712	0.729	0.392	0.341	0.802	0.751
6	0.731	0.755	0.810	0.375	0.822	0.772
12	**0.741**	**0.761**	**0.821**	**0.385**	**0.836**	**0.785**
18	0.734	0.753	0.818	0.377	0.831	0.782
36	0.722	0.753	0.820	0.382	0.830	0.781

Table 5: Quantitative comparison of different numbers of rendered view on the val set of SHREC 2013 dataset. The best results are in bold font.

Numbers	NN	FT	ST	E	DCG	mAP
3	0.438	0.380	0.441	0.214	0.553	0.394
4	0.451	0.413	0.459	0.223	0.561	0.428
6	0.487	0.413	0.463	0.255	0.593	0.446
12	**0.534**	**0.456**	**0.509**	**0.261**	**0.614**	**0.475**
18	0.523	0.443	0.493	0.254	0.606	0.468
36	0.511	0.431	0.487	0.251	0.604	0.464

computational efficiency. As a result, the number of 12 is chosen as the number of rendered views for 3D shapes in this work.

Effect of number of representative views selection. As described in Subsection 3.2, the number of selected most representative views by RVS module can affect the retrieval performance of the proposed method. As mentioned above, each 3D shape is represented by 12 rendered views, and some of the them can be considered as redundancy due to the their appearance similarity. Consequently, a RVS module is introduced to reduce such redundancy and results in different number of representative views".

Considering the input number is 12 views, we conduct an experiment on the on the val set and testing set of SHREC 2013 dataset by varying different number of the resulted "representative views" (e.g., 2, 5, 8 and 10) aiming to evaluate its effect on the performance caused by the redundancy reduction. Table 5 shows the quantitative comparison using the common evaluation metrics. Note that, number of 12 means using

Table 6: Quantitative comparison of different numbers of "representative views" on the test set of SHREC 2013 dataset. The best results are in bold font.

Numbers	NN	FT	ST	E	DCG	mAP
2	0.725	0.746	0.804	0.382	0.822	0.768
5	**0.741**	**0.761**	**0.821**	**0.385**	**0.836**	**0.785**
8	0.736	0.755	0.817	0.378	0.832	0.778
10	0.710	0.726	0.791	0.366	0.817	0.751
12	0.715	0.737	0.801	0.369	0.821	0.752

Table 7: Quantitative comparison of different numbers of "representative views" on the val set of SHREC 2013 dataset. The best results are in bold font.

Numbers	NN	FT	ST	E	DCG	mAP
2	0.496	0.412	0.477	0.246	0.588	0.443
5	**0.534**	**0.456**	**0.509**	**0.261**	**0.614**	**0.475**
8	0.512	0.430	0.474	0.254	0.598	0.456
10	0.440	0.379	0.443	0.229	0.552	0.396
12	0.488	0.402	0.461	0.237	0.576	0.428

all the 12 rendered views without RVS module. It can be observed that, 5 "representative views" results in the best retrieval performance whether on the val set or the test set, which verifies our hypothesis about "redundancy information".

Effect of hyper-parameter λ: As described in Subsection 3.3, the overall loss function of the proposed method contains two terms, i.e., between-class term and within-class term. The relative distance loss tends to maximize the between-class distance, while the absolute distance loss tends to minimize within-class distance. Therefore, there is a hyper-parameter λ to balance the total loss terms between such two terms. Note that, the order of magnitude of absolute distance loss is larger than relative distance loss obviously. $\lambda = 0$ means that we only use relative distance loss function without considering the absolute distance function. Specifically, we conduct an experiment on the val set and the test set of SHREC 2013 dataset by varying different values of hyper-parameter λ (e.g., 0, 0.0001, 0.001, 0.01, 0.1, 1) aiming to evaluate its effect on the retrieval performance caused by the different contributions of the two terms. Fig.12 shows the mAP versus λ when testing on the val set and the test set of SHREC 2013

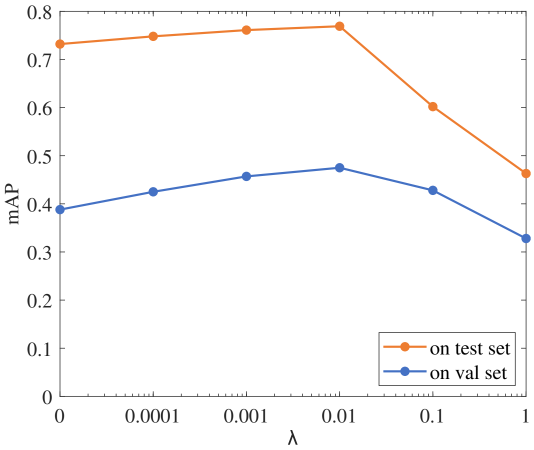

Figure 12: The mAP of proposed method versus λ when testing on the val set and the test set of SHREC 2013 dataset.

dataset based on AlexNet as the CNN backbone. We notice that the optimal trade-off weight is same for both val set and test set. Therefore, setting hyper-parameter $\lambda = 0.01$ leads to the best performance.

Effect of classes with similar appearance: Discriminating the 3D shapes with similar appearance is a challenging task. To evaluate the distinctiveness of our proposed method under visually similar categories, we generate a subset of 3D shapes from the SHREC 2013 dataset called Sim_SHREC 2013 dataset, which includes 3,600 sketches and 706 3D shapes belonging to 45 classes. As illustrated in Fig. 13, the Sim_SHREC 2013 dataset contains pairs of 3D shapes with similar apperance, which is more chanllenging than origininal SHREC 2013 dataset.

A quantitative comparison of the proposed method on the SHREC 2013 and Sim_SHREC 2013 datasets is shown in Tab. 8. As a result,the performance of our method on this challenging dataset is worse than that on its original counterpart, e.g. the NN measure is decreased by 0.037. However, this performance is still reasonable, which demonstrates the discriminative power of the proposed representation. The main reason for this deterioration is that the overlap may exist between similar shape feature subspaces, especially those ones sharing similar key components. For example, both bicycle and motorbike have one frame, one handle and two wheels. Therefore, such shared key elements may lead to similar feature encoding results in the feature subspaces and different degree of incorrect recognition. However, our proposed method still

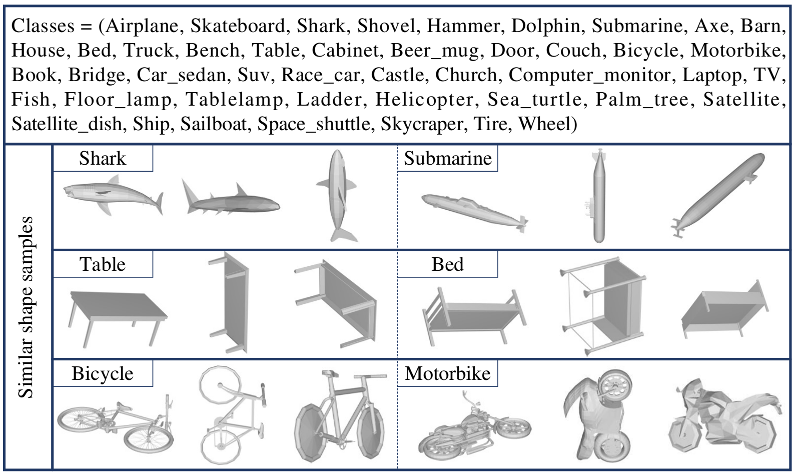
Classes = (Airplane, Skateboard, Shark, Shovel, Hammer, Dolphin, Submarine, Axe, Barn, House, Bed, Truck, Bench, Table, Cabinet, Beer_mug, Door, Couch, Bicycle, Motorbike, Book, Bridge, Car_sedan, Suv, Race_car, Castle, Church, Computer_monitor, Laptop, TV, Fish, Floor_lamp, Tablelamp, Ladder, Helicopter, Sea_turtle, Palm_tree, Satellite, Satellite_dish, Ship, Sailboat, Space_shuttle, Skycraper, Tire, Wheel)

Figure 13: The complete list of class names and some examples from the Sim_SHREC 2013 dataset. Note that, each row contains a pair of 3D shapes with similar appearance, e.g., shark/submarine, table/bed and bicyvle and motorbike.

Table 8: Quantitative comparison the proposed method on the SHREC 2013 dataset with full 90 classes and the Sim_SHREC 2013 dataset with 45 selected classes.

Dataset	NN	FT	ST	E	DCG	mAP
SHREC 2013	0.741	0.761	0.821	0.385	0.836	0.785
Sim_SHREC 2013	0.704	0.736	0.802	0.380	0.828	0.765

achieves a reasonable performance for all the evaluation measures, which validate the robustness of the proposed method dealing with the 3D shapes with similar apperance.

6. Conclusions

In this paper, we propose a novel DPSML framework for sketch-based 3D shape retrieval. First, the raw features for both sketches and 3D shapes (represented by 12 rendered views) are extracted via pre-trained deep models (AlexNet, VGG and ResNet). Second, a RVS module is introduced to reduce the redundancy of the rendered views and results in a set of most representative views. Then, the sketch is projected into a feature point and the 3D shape is projected into a subspace which is spanned by the obtained basis feature vectors from the selected representative views. Finally, the similarity of the query sketch and a 3D shape is defined as the distance of the query sketch feature vector and the closest point in the spanned space of the 3D shape, which reflects the viewpoint information determined by the input query sketch. More specifically, we formulate the representation learning problem as the classification problem for the sketch side and the multi-instance classification problem for the 3D shape side, which guarantees the training efficiency. The overall loss function consists of two parts, i.e., the relative distance part and an absolute distance part. The first part aims to learn a class center for minimize the between-class distance and the second part aims to maximize the within-class distance. We demonstrated the effectiveness of the proposed method on three publicly available large-scale datasets (i.e., SHREC 2013, 2014 and 2016), and a superior retrieval performance over the state-of-the-arts was achieved.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (NSFC) with Grant Nos. 61403265, 61602499, 61725202, 61751212 and 61771088. Lingqiao Liu was in part supported by ARC DECRA Fellowship DE170101259. This work was also partially supported by the Fundamental Research Funds for the Central Universities (No. 18lgzd06).

[1] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, The princeton shape benchmark, in: Proceedings of the Shape Modeling Applications, IEEE, 2004, pp. 167–178.

[2] J. Shih, C. Lee, J. Wang, A new 3D model retrieval approach based on the elevation descriptor, Pattern Recognition 40 (1) (2007) 283–295.

[3] D. Pickup, X. Sun, P. L. Rosin, R. R. Martin, Euclidean-distance-based canonical forms for non-rigid 3D shape retrieval, Pattern Recognition 48 (8) (2015) 2500–2512.

[4] P. Dou, Y. Wu, S. K. Shah, I. A. Kakadiaris, Monocular 3D facial shape reconstruction from a single 2D image with coupled-dictionary learning and sparse coding, Pattern Recognition 81 (2018) 515–527.

[5] M.-L. Torrente, S. Biasotti, B. Falcidieno, Recognition of feature curves on 3D shapes using an algebraic approach to hough transforms, Pattern Recognition 73 (2018) 111–130.

[6] C. Lv, Z. Wu, X. Wang, M. Zhou, K. A. Toh, Nasal similarity measure of 3D faces based on curve shape space, Pattern Recognition 88 (2019) 458–469.

[7] J. W. Tangelder, R. C. Veltkamp, A survey of content based 3D shape retrieval methods, in: Proceedings of Shape Modeling Applications, IEEE, 2004, pp. 145–156.

[8] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, K. Ramani, Three-dimensional shape searching: state-of-the-art review and future trends, Computer-Aided Design 37 (5) (2005) 509–530.

[9] M. Eitz, K. Hildebrand, T. Boubekeur, M. Alexa, Sketch-based shape retrieval, ACM Transactions on Graphics (TOG) 31 (4) (2012) 1–10.

[10] B. Gong, J. Liu, X. Wang, X. Tang, Learning semantic signatures for 3D object retrieval, IEEE Transactions on Multimedia 15 (2) (2013) 369–377.

[11] T. Furuya, R. Ohbuchi, Ranking on cross-domain manifold for sketch-based 3D model retrieval, in: Proceedings of the Cyberworlds (CW), IEEE, 2013, pp. 274–281.

[12] B. Li, Y. Lu, A. Godil, T. Schreck, B. Bustos, A. Ferreira, T. Furuya, M. J. Fonseca, H. Johan, T. Matsuda, et al., A comparison of methods for sketch-based 3D shape retrieval, Computer Vision and Image Understanding 119 (2014) 57–80.

[13] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3D ShapeNets: A deep representation for volumetric shapes, in: Proceedings of the Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 1912–1920.

[14] A. Garcia Garcia, F. Gomez Donoso, J. Garcia Rodriguez, S. Orts Escolano, M. Cazorla, J. Azorin Lopez, PointNet: A 3D convolutional neural network for real-time object class recognition, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, pp. 1578–1584.

[15] H. Cheng, S. M. Chung, Orthogonal moment-based descriptors for pose shape query on 3D point cloud patches, Pattern Recognition 52 (2016) 397–409.

[16] X. Liu, Z. Han, Y. Liu, M. Zwicker, Point2Sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network, arXiv preprint arXiv:1811.02565.

[17] J. Xie, G. Dai, F. Zhu, Y. Fang, Learning barycentric representations of 3D shapes for sketch-based 3D shape retrieval, in: Proceedings of the Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 3615–3623.

[18] A. Kanezaki, Y. Matsushita, Y. Nishida, RotationNet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints, in: Proceedings of the Computer Vision and Pattern Recognition (CVPR), IEEE, 2018, pp. 5010–5019.

[19] Y. Feng, Z. Zhang, X. Zhao, R. Ji, Y. Gao, GVCNN: Group-view convolutional neural networks for 3D shape recognition, in: Proceedings of the Computer Vision and Pattern Recognition (CVPR), IEEE, 2018, pp. 264–272.

[20] H. You, Y. Feng, R. Ji, Y. Gao, PVNet: A joint convolutional network of point cloud and multi-view for 3D shape recognition, in: Proceedings of the ACM International Conference on Multimedia, ACM, 2018, pp. 1310–1318.

[21] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, in: Proceedings of the Neural Information Processing Systems (NeurIPS), 2012, pp. 1–9.

[22] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

[23] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 770–778.

[24] J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Feifei, ImageNet: A large-scale hierarchical image database, in: Proceedings of the Computer Vision and Pattern Recognition (CVPR), IEEE, 2009, pp. 248–255.

[25] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3D shape descriptors, in: Proceedings of the Symposium on Geometry Processing (SGP), 2003, pp. 156–164.

[26] J. Xie, G. Dai, F. Zhu, E. Wong, Y. Fang, Deepshape: Deep-learned shape descriptor for 3D shape retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 39 (7) (2016) 1335–1345.

[27] Y. Fang, J. Xie, G. Dai, M. Wang, F. Zhu, T. Xu, E. Wong, 3D deep shape descriptor, in: Proceedings of the Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 2319–2328.

[28] H. Su, S. Maji, E. Kalogerakis, E. Learnedmiller, Multi-view convolutional neural networks for 3D shape recognition, in: Proceedings of the Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 945–953.

[29] X. He, Y. Zhou, Z. Zhou, S. Bai, X. Bai, Triplet-center loss for multi-view 3D object retrieval, in: Proceedings of the Computer Vision and Pattern Recognition (CVPR), IEEE, 2018, pp. 1945–1954.

[30] T. Yu, J. Meng, J. Yuan, Multi-view harmonized bilinear network for 3D object recognition, in: Proceedings of the Computer Vision and Pattern Recognition (CVPR), IEEE, 2018, pp. 186–194.

[31] F. Wang, L. Kang, Y. Li, Sketch-based 3D shape retrieval using convolutional neural networks, in: Proceedings of the Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 1875–1883.

[32] B. Li, Y. Lu, C. Li, A. Godil, T. Schreck, M. Aono, M. Burtscher, Q. Chen, N. K. Chowdhury, B. Fang, A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries, Computer Vision and Image Understanding 131 (C) (2015) 1–27.

[33] B. Li, Y. Lu, A. Godil, T. Schreck, M. Aono, H. Johan, J. M. Saavedra, S. Tashiro, Shrec‘13 track: Large scale sketch-based 3D shape retrieval, in: Proceedings of the Eurographics Workshop on 3D Object Retrieval (3DOR), Eurographics Association, 2012, pp. 89–96.

[34] F. Zhu, J. Xie, Y. Fang, Learning cross-domain neural networks for sketch-based 3D shape retrieval, in: Proceedings of the American Association for Artificial Intelligence (AAAI), 2016, pp. 3683–3689.

[35] Z. Yasseen, A. Verroust-Blondet, A. Nasri, View selection for sketch-based 3D model retrieval using visual part shape description, The Visual Computer 33 (5) (2017) 565–583.

[36] G. J. Yoon, M. Y. Sang, Sketch-based 3D object recognition from locally optimized sparse features, Neurocomputing 267.

[37] B. Li, Y. Lu, H. Johan, R. Fares, Sketch-based 3D model retrieval utilizing adaptive view clustering and semantic information, Multimedia Tools and Applications 76 (24) (2017) 26603–26631.

[38] S. Bai, X. Bai, Z. Zhou, Z. Zhang, L. Jan Latecki, Gift: A real-time and scalable 3D shape search engine, in: Proceedings of the Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 5023–5032.

[39] K. Sarkar, B. Hampiholi, K. Varanasi, D. Stricker, Learning 3D shapes as multi-layered height-maps using 2D convolutional networks, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2018, pp. 74–89.

[40] J. Chen, Y. Fang, Deep cross-modality adaptation via semantics preserving adversarial learning for sketch-based 3D shape retrieval, arXiv preprint arXiv:1807.01806.

[41] G. Dai, J. Xie, F. Zhu, Y. Fang, Deep correlated metric learning for sketch-based 3D shape retrieval., in: Proceedings of the American Association for Artificial Intelligence (AAAI), 2017, pp. 4002–4008.

[42] G. Dai, J. Xie, Y. Fang, Deep correlated holistic metric learning for sketch-based 3D shape retrieval, IEEE Transactions on Image Processing (TIP) 27 (7) (2018) 3374.

[43] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the computer vision and pattern recognition (CVPR), IEEE, 2015, pp. 815–823.

[44] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A discriminative feature learning approach for deep face recognition, in: European Conference on Computer Vision (ECCV), Springer, 2016, pp. 499–515.

[45] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, Choosing multiple parameters for support vector machines, Machine learning 46 (1-3) (2002) 131–159.

[46] B. Li, Y. Lu, C. Li, A. Godil, T. Schreck, M. Aono, M. Burtscher, H. Fu, T. Furuya, H. Johan, et al., Shrec'14 track: Extended large scale sketch-based shape retrieval, in: Proceedings of the Eurographics Workshop on 3D Object Retrieval (3DOR), Eurographics Association, 2014, pp. 121–120.

[47] B. Li, Y. Lu, F. Duan, S. Dong, Y. Fan, L. Qian, H. Laga, H. Li, Y. Li, P. Lui, et al., Shrec'16 track: 3D sketch-based 3D shape retrieval, in: Proceedings of the Eurographics Workshop on 3D Object Retrieval (3DOR), Eurographics Association, 2016.

[48] B. Li, A. Godil, M. Aono, X. Bai, T. Furuya, L. Li, R. J. López-Sastre, H. Johan, R. Ohbuchi, C. Redondo-Cabrera, et al., Shrec'12 track: Generic 3D shape retrieval, in: Proceedings of the Eurographics Workshop on 3D Object Retrieval (3DOR), Eurographics Association, 2012, pp. 119–126.

[49] A. Tatsuma, H. Koyanagi, M. Aono, A large-scale shape benchmark for 3D object retrieval: Toyohashi shape benchmark, in: Proceedings of the Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), IEEE, 2012, pp. 1–10.

[50] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 1026–1034.

[51] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

[52] P. Sousa, M. J. Fonseca, Sketch-based retrieval of drawings using spatial proximity, Journal of Visual Languages and Computing 21 (2) (2010) 69–80.

[53] H. Tabia, H. Laga, Learning shape retrieval from different modalities, Neurocomputing 253 (2017) 24–33.