Learning Barycentric Representations of 3D Shapes for Sketch-based 3D Shape Retrieval

Jin Xie, Guoxian Dai, Fan Zhu, and Yi Fang

NYU Multimedia and Visual Computing Lab
Department of Electrical and Computer Engineering, New York University Abu Dhabi
Department of Electrical and Computer Engineering, NYU Tandon School of Engineering

{jin.xie, guoxian.dai, fan.zhu, yfang}@nyu.edu

Abstract

Retrieving 3D shapes with sketches is a challenging problem since 2D sketches and 3D shapes are from two heterogeneous domains, which results in large discrepancy between them. In this paper, we propose to learn barycenters of 2D projections of 3D shapes for sketch-based 3D shape retrieval. Specifically, we first use two deep convolutional neural networks (CNNs) to extract deep features of sketches and 2D projections of 3D shapes. For 3D shapes, we then compute the Wasserstein barycenters of deep features of multiple projections to form a barycentric representation. Finally, by constructing a metric network, a discriminative loss is formulated on the Wasserstein barycenters of 3D shapes and sketches in the deep feature space to learn discriminative and compact 3D shape and sketch features for retrieval. The proposed method is evaluated on the SHREC'13 and SHREC'14 sketch track benchmark datasets. Compared to the state-of-the-art methods, our proposed method can significantly improve the retrieval performance.

1. Introduction

With the development of touch-screen technology, sketching has became much easier as a way to interact with computer systems such as tablet computers and smart phones. Sketch-based 3D shape retrieval has been receiving more and more attention in the community of computer graphics and computer vision [15, 13, 27, 28]. In comparison to text and 3D shapes as queries, sketches are intuitive and convenient for users to search 3D models.

Compared to 3D shape retrieval using 3D shapes as queries, sketch-based 3D shape retrieval is much more challenging. Since sketches are highly abstract and subjectively drawn, there are usually large variations with them. Moreover, due to the discrepancy between the sketch and 3D shape domains, there are also large cross-domain variations between 2D sketches and 3D shapes, which usually degenerates the performance of the learning model.

In recent years, extensive research efforts have been dedicated to sketch-based 3D shape retrieval. In [13], an integrated descriptor ZFEC is proposed, including the region-based Zernike moments, contour-based Fourier descriptor, eccentricity feature and circularity feature, to describe sketches and projections of 3D shapes. Then, in order to align sketches and 3D shapes, the representative views of 3D shapes are selected with the defined view context. The shape context matching is used to compare the sketch to each selected view with the ZFEC feature for retrieval. In [12], silhouettes of 3D shapes from the defined views are used to represent 3D shapes. Then, three kinds of descriptors, histogram of edge local orientations (HELO), histogram of oriented gradients (HOG) and Fourier descriptors, are extracted to describe sketches and 3D shapes. The KD-tree with the Manhattan distance is used for matching sketches to 2D projections of 3D shapes.

Besides the hand-crafted features, learning-based features are also extracted to represent sketches and 3D shapes for sketch-based 3D shape retrieval. In [15], followed by the Gabor local line-based features (GALIFs), the bag-of-feature (BOF) histograms are extracted to represent sketches and 2D projections of 3D shapes. Then the minimum distance between the BOF histograms of the sketch and 2D projections is used as similarity for retrieval. Furuya *et al.* [10] proposed the BF-SIFT feature to describe sketches and 2D projections, which is the dense SIFT with the BOF method. By constructing a Laplacian graph in the feature space of the sketches and 2D projections of 3D

shapes, the manifold ranking method is employed to calculate the distances between sketches and 2D projections for retrieval.

Recently, due to the success of deep neural networks in different applications, deep features have been proposed for sketch-based 3D shape retrieval. In [27], two views are first selected to characterize 3D shapes, where the difference between their angles is larger than 45 degrees. Then, two Siamese convolutional neural networks (CNNs) are used, one for the sketch domain and the other for the view domain. The within-domain loss and cross-domain loss are defined to learn discriminative sketch and view features for retrieval. Zhu *et al*. [28] employed view-invariant local depth scale-invariant feature transform (LD-SIFT) [8] to characterize 3D shapes. Then, the pyramid cross-domain neural networks are constructed, where the output layers of the two neural networks from the same class are identical. The outputs from the hidden layers are used as the learned features for retrieval.

The 3D shape retrieval task using 3D shapes as queries is also related to our task. In 3D shape retrieval, Shi *et al*. [18] proposed the deep panoramic representation for 3D shape retrieval, where the CNN is applied to the panoramic representation of 3D shapes for learning deep shape features. Based on the CNN features of 2D projections of 3D shapes, Bai *et al*. [1] proposed a real-time shape retrieval method by speeding up multi-view matching. In [23], by performing a max view-pooling operation, multi-view CNN is proposed to learn a compact shape feature from multiple projections of 3D shapes.

In most of the aforementioned approaches, the retrieval task is converted into the problem of matching sketches to multiple views of 3D shapes (sketch-based 3D shape retrieval) or the multi-view matching problem (3D shape retrieval), where multiple separate descriptors from views of 3D shapes are used. In this paper, instead of independently using multiple views to characterize 3D shapes, we propose to learn Wasserstein barycenters of multiple views of 3D shapes in the feature space for sketch-based shape retrieval. It is expected that we can take full advantage of the information of multiple views of 3D shapes simultaneously to characterize 3D shapes. First, we project 3D shapes to a set of rendered views. We employ two deep CNNs to extract the CNN features of sketches and 2D projections. The Wasserstein barycenters of CNN features of 2D projections can then be computed to characterize 3D shapes. Consequently, with a metric network, a discriminative loss is defined on the barycenters of 3D shapes and sketches in the feature space, which can maximize the within-class similarity and minimize the between-class similarity across the sketch and view domains, simultaneously. Experimental results on two benchmark datasets demonstrate the effectiveness of the proposed approach for sketch-based 3D shape retrieval.

To summarize, our main contributions are as follows :

- We propose to use the Wasserstein barycenters of multiple projections of 3D shapes to characterize 3D shapes;
- We formulate a deep metric learning model to learn the Wasserstein barycentric representation;
- We significantly outperform the state-of-the-art sketch-based 3D shape retrieval methods on two large benchmark datasets.

The rest of the paper is organized as follows. Section 2 introduces the background of the Wasserstein distance and Wasserstein barycenters. In Section 3, we propose the learning-based Wasserstein barycentric representation for sketch-based 3D shape retrieval. Section 4 presents the experimental results and Section 5 concludes the paper.

2. Background

In this section, we briefly review the definition of the Wasserstein distance and introduce the Wasserstein barycenters.

2.1. Wasserstein distance

The Wasserstein distance [3] defines a distance between two probability distributions, which arises in the theory of optimal transformation [26]. The Wasserstein distance has been widely used in computer vision [17, 21] and machine learning [16, 5]. Let $\boldsymbol{p} \in \mathbb{R}^{r\times 1}$ and $\boldsymbol{q} \in \mathbb{R}^{s\times 1}$ be two probability distributions, respectively. The set of transportation plans between probability distributions $\boldsymbol{p}$ and $\boldsymbol{q}$ is defined as follows:

$$R(\boldsymbol{p},\boldsymbol{q}) = \left\{\boldsymbol{T} \in \mathbb{R}_+^{r\times s}; \boldsymbol{T}\mathbf{1} = \boldsymbol{p}, \boldsymbol{T}^T\mathbf{1} = \boldsymbol{q}\right\} \quad (1)$$

where $\boldsymbol{T}$ is the transportation plan and $\mathbf{1}$ is a column vector whose elements are 1. The Wasserstein distance $D(\boldsymbol{p},\boldsymbol{q})$ between $\boldsymbol{p}$ and $\boldsymbol{q}$ can be defined as the following optimal value:

$$D(\boldsymbol{p},\boldsymbol{q}) = \min_{\boldsymbol{T}\in R(\boldsymbol{p},\boldsymbol{q})} < \boldsymbol{M}, \boldsymbol{T} > \quad (2)$$

where $\boldsymbol{M} \in \mathbb{R}^{r\times s}$ is a pairwise distance matrix between $\boldsymbol{p}$ and $\boldsymbol{q}$, called the ground metric, $< \boldsymbol{M}, \boldsymbol{T} >$ is the inner-product of $\boldsymbol{M}$ and $\boldsymbol{T}$, i.e., $< \boldsymbol{M}, \boldsymbol{T} >= tr(\boldsymbol{M}^T\boldsymbol{T})$. The Wasserstein distance $D(\boldsymbol{p},\boldsymbol{q})$ can be viewed as the cost of the optimal transportation plan that transports the mass of $\boldsymbol{p}$ to the mass of $\boldsymbol{q}$.

Since in many cases the linear program problem in Eq. (2) does not have a unique solution, Cuturi [5] proposed to add an entropy regularization term to Eq. (2) as:

$$D(\boldsymbol{p},\boldsymbol{q}) = \min_{\boldsymbol{T}\in R(\boldsymbol{p},\boldsymbol{q})} < \boldsymbol{M}, \boldsymbol{T} > +\gamma < \boldsymbol{T}, log\boldsymbol{T} > \quad (3)$$

where $< \boldsymbol{T}, log\boldsymbol{T} >$ is the negative entropy and γ is the regularization parameter. The problem in Eq. (3) can have a unique solution and the optimal solution can be obtained as:

$$\hat{\boldsymbol{T}} = diag(\boldsymbol{u})\boldsymbol{K}diag(\boldsymbol{v}) \tag{4}$$

where $\boldsymbol{K} = e^{-\boldsymbol{M}/\gamma}$, and vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ can be solved using the Sinkhorn's algorithm [20].

2.2. Wasserstein barycenters

The barycenters $\boldsymbol{p}_b$ of a set of probability distributions $\boldsymbol{p}_i \in \mathbb{R}^{r\times 1}$, $i = 1, 2, \cdots, n$, can be defined as [7, 6, 4]:

$$argmin_{\boldsymbol{p}_b} \sum_{i=1}^{n} \lambda_i D(\boldsymbol{p}_b, \boldsymbol{p}_i) \tag{5}$$

where $D(\boldsymbol{p}_b, \boldsymbol{p}_i)$ is the Wasserstein distance between $\boldsymbol{p}_b$ and $\boldsymbol{p}_i$ and λ_i is the weight. Based on the entropy regularized Wasserstein distance, Benamou *et al*. [2] proposed to use the Sinkhorn fixed-point algorithm to solve Eq. (5). The regularized Wasserstein barycenters $\boldsymbol{p}_b$ can be obtained by iteratively computing $\boldsymbol{p}_b^t$:

$$\begin{aligned} \boldsymbol{p}_b^t &= \prod_{i=1}^{n} (\boldsymbol{K}^T \boldsymbol{a}_i^t)^{\lambda_i} \\ \boldsymbol{c}_i^{t+1} &= \frac{\boldsymbol{p}_b^t}{\boldsymbol{K}^T \boldsymbol{a}_i^t}; \boldsymbol{a}_i^{t+1} = \frac{\boldsymbol{p}_i}{\boldsymbol{K}\boldsymbol{c}_i^{t+1}} \end{aligned} \tag{6}$$

where $\boldsymbol{p}_b^t$ is the tth iteration of the Wasserstein barycenters $\boldsymbol{p}_b$, $\boldsymbol{a}_i^t$ and $\boldsymbol{c}_i^t$ are auxiliary variables, $\boldsymbol{a}_i^1 = \boldsymbol{1}$, $t = 1, 2, \cdots, Q$, and Q is the iteration number. It is noted that in Eq. (6) $\prod \boldsymbol{x}_i$ and $\frac{\boldsymbol{x}}{\boldsymbol{y}}$ are the element-wise product and division operations. The readers can refer to Proposition 2 in [2] and Proposition 1 in [4] for more details.

3. Proposed Approach

In this section, we present our learned Wasserstein barycenters of 3D shapes for sketch-based 3D shape retrieval. In subsection 3.1, we propose barycentric representations of projections of 3D shapes in the CNN feature space. In subsection 3.2, we present cross-domain matching with the learned Wasserstein barycenters.

3.1. Wasserstein barycentric representations of 3D shapes

The Wasserstein barycenters can be used to estimate the mean of a family of probability distributions. Specifically, the Wasserstein barycenters have been applied in a variety of vision problems, such as texture synthesis [9] and color editing [4]. Due to the property of the Wasserstein barycenters that can capture the structure of the high-dimensional data well [6], in this work, we propose to use the Wasserstein barycenters of projections from multiple views in the feature space to characterize 3D shapes.

Our Wasserstein barycentric representations start from multiple projections of 3D shapes. Following the settings in [23], we uniformly scale each 3D shape and put the centroid of the shape at the origin of the spherical coordinate system. By placing V virtual cameras around the 3D shape evenly, V rendered views can be obtained. For each view, we render the 3D shape to a greyscale image as a projection.

Once 2D projections of 3D shapes are obtained, we can extract the deep CNN features of 2D projections. In this work, we employ the AlexNet [11] for the CNN features, which consists of five convolutional layers followed by three fully connected layers. For each projection, the last fully connected layer after the ReLU non-linear activation function is used as the deep feature, whose feature size is 4096.

Suppose that there are n_1 shapes and we use S_i to represent the ith shape. For the shape S_i, we denote the deep CNN feature of the projection from view j by $\boldsymbol{x}_{i,j}$, $j = 1, 2, \cdots, V$. Since the Wasserstein distance is defined in the space of probability distributions, we need to guarantee each element $\boldsymbol{x}_{i,j}(l)$ in the CNN feature $\boldsymbol{x}_{i,j}$ to be $\boldsymbol{x}_{i,j}(l) \geq 0$ and $\sum_l \boldsymbol{x}_{i,j}(l) = 1$. In the final fully convolutional layer of the AlexNet we use the ReLU function as the activation fuction. Therefore, we can normalize $\boldsymbol{x}_{i,j}$ to be $\sum_l \boldsymbol{x}_{i,j}(l) = 1$ as input to compute the Wasserstein barycenters. Given the normalized CNN feature $\boldsymbol{x}_{i,j} \in \mathbb{R}_+^{L\times 1}$, $j = 1, 2, \cdots, V$, the isotropic Wasserstein barycenters $\boldsymbol{h}_i \in \mathbb{R}^{L\times 1}$ of these features can be obtained as:

$$argmin_{\boldsymbol{h}_i} \sum_{j=1}^{V} \frac{1}{V} D(\boldsymbol{h}_i, \boldsymbol{x}_{i,j}) \tag{7}$$

where $D(\boldsymbol{h}_i, \boldsymbol{x}_{i,j})$ is the entropy regularized Wasserstein distance between $\boldsymbol{h}_i$ and $\boldsymbol{x}_{i,j}$. By iteratively computing Eq. (6), the Wasserstein barycenters $\boldsymbol{h}_i$ can be solved.

Fig. 1 shows a visualization of the Wasserstein barycenters of the deep features of projects from ten classes of 3D shapes. We use the t-distributed stochastic neighbor embedding (t-SNE) technique [25] to reduce the dimension to two for visualization. From this figure, we can see that in most cases the shapes from the same class are grouped together (marked in the same color) while the shapes from different classes are separated. This implies that although there are complex geometric structural variations with 3D shapes the Wasserstein barycenters of the deep features of the projections has the potential to represent 3D shapes discriminatively.

It is noted that the Wasserstein barycenters $\boldsymbol{h}_i$ are different from the linear average $\sum_{j=1}^{V} \frac{1}{V} \boldsymbol{x}_{i,j}$, corresponding to the Euclidean distance based barycenters. The linear averaging of the feature $\boldsymbol{x}_{i,j}$, as a linear element-wise operation, does not consider the geometric structure of $\boldsymbol{x}_{i,j}$. Therefore, the linear averaging is geometrically-oblivious. Based

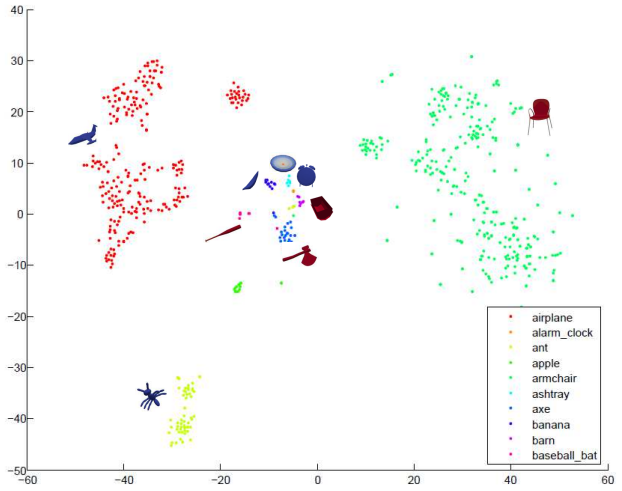

Figure 1. Visualization of the Wasserstein barycenters of the deep features of projections from ten classes of 3D shapes: airplane, alarm_clock, ant, apple, armchair, ashtray, axe, banana, barn and baseball_bat. The Wasserstein barycenters from the same class of shapes are marked in the same color.

on the ground metric, the Wasserstein distance seeks an optimal transportation plan to transport the mass of one histogram to the mass of the other, which considers the information of the non-corresponding bins. Thus, based on the Wasserstein distance, the Wasserstein barycentric representation as a non-linear operation can capture the structure of the deep features $\boldsymbol{x}_{i,j}$ better.

3.2. Cross-domain matching with learned Wasserstein barycenters

In this subsection, we present to learn the Wasserstein barycenters of the deep CNN features of 3D shapes for sketch-based 3D shape retrieval. As mentioned in Section 3.1, we employ two AlexNets to extract deep CNN features of 2D projections and sketches, respectively. Once the CNN features from 3D shapes and sketches are obtained, we use fully connected layers to construct a metric network. Fig. 2 illustrates the cross-domain matching framework with two networks, one for sketch network and the other for view network. In the view network, the CNNs which correspond to different views share the same parameters. Moreover, the structure of the AlexNet for sketches is the same to that of the AlexNet for 3D shapes. Nonetheless, the weights and biases of the two networks are different.

The constructed metric network can map the Wasserstein barycenters of 2D projections and the sketch features to the corresponding outputs. Let the outputs of the view network and sketch network be $\boldsymbol{z}_i^1$ and $\boldsymbol{z}_j^2$, $i = 1, 2, \cdots, n_1$, $j = 1, 2, \ldots, n_2$, n_1 and n_2 are the numbers of 3D shapes and sketches, respectively. In the transformed non-linear feature space, it is desirable that across the domains the similarity between the features $\boldsymbol{z}_i^1$ and $\boldsymbol{z}_j^2$ from the same class is as large as possible and the similarity between the features from different classes is as small as possible while within the domains both outputs $\boldsymbol{z}_i^1$ and $\boldsymbol{z}_j^2$ are discriminative.

To this end, we propose the following discriminative loss function:

$$(\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_2) = argmin_{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2} \frac{1}{\sum n_j} \sum_{j=1}^{n_2} \sum_{i \in c(j)} \|\boldsymbol{z}_j^2 - \boldsymbol{z}_i^1\|_2^2 + \frac{1}{\sum m_j} \sum_{j=1}^{n_2} \sum_{i \notin c(j)} max(0, \alpha - \|\boldsymbol{z}_j^2 - \boldsymbol{z}_i^1\|_2^2) + \beta_1 L_1 + \beta_2 L_2 \tag{8}$$

where $max(0, \alpha - \|\boldsymbol{z}_j^2 - \boldsymbol{z}_i^1\|_2^2)$ is a hinge loss function to penalize the similarity from different classes that is less than the threshold α, $c(j)$ denotes the class label of the feature $\boldsymbol{z}_j^2$, n_j and m_j are the numbers of positive and negative shape samples of the jth sketch, L_1 and L_2 are the regularization terms to achieve discrimination of the within-domain shape and sketch features, $\boldsymbol{\theta}_1 = \{\boldsymbol{W}_1, \boldsymbol{b}_1\}$ and $\boldsymbol{\theta}_2 = \{\boldsymbol{W}_2, \boldsymbol{b}_2\}$ are the sets of weights and bias of the view and sketch networks, respectively, β_1 and β_2 are the regularization parameters. In Eq. (8), the regularized discrimination term L_1 is defined as:

$$L_1 = tr(\sum_{i=1}^{n_1} (\boldsymbol{z}_i^1 - \boldsymbol{m}_{c(i)}^1)(\boldsymbol{z}_i^1 - \boldsymbol{m}_{c(i)}^1)^T) - tr(\sum_{i=1}^{n_1} (\boldsymbol{m}_{c(i)}^1 - \boldsymbol{m}^1)(\boldsymbol{m}_{c(i)}^1 - \boldsymbol{m}^1)^T) \tag{9}$$

where $\boldsymbol{m}_{c(i)}^1$ is the mean of the shape features from class label $c(i)$, $\boldsymbol{m}^1$ is the mean of the shape features from all classes, the first term is the within-class scatter of the shape features and the second term is the between-class scatter of the shape features. Similar to Eq. (9), the discrimination term L_2 is defined as:

$$L_2 = tr(\sum_{j=1}^{n_2} (\boldsymbol{z}_j^2 - \boldsymbol{m}_{c(j)}^2)(\boldsymbol{z}_j^2 - \boldsymbol{m}_{c(j)}^2)^T) - tr(\sum_{j=1}^{n_2} (\boldsymbol{m}_{c(j)}^2 - \boldsymbol{m}^2)(\boldsymbol{m}_{c(j)}^2 - \boldsymbol{m}^2)^T) \tag{10}$$

where $\boldsymbol{m}_{c(j)}^2$ is the mean of the sketch features from class label $c(j)$, $\boldsymbol{m}^2$ is the mean of the sketch features from all classes.

In the proposed cross-domain matching model Eq. (8), the first two terms minimize the within-class distances and maximize the between-class distances between pairs of cross-domain features, simultaneously. The regularization terms minimize the within-class scatter and maximize the between-class scatter of the 3D shape and sketch features, respectively.

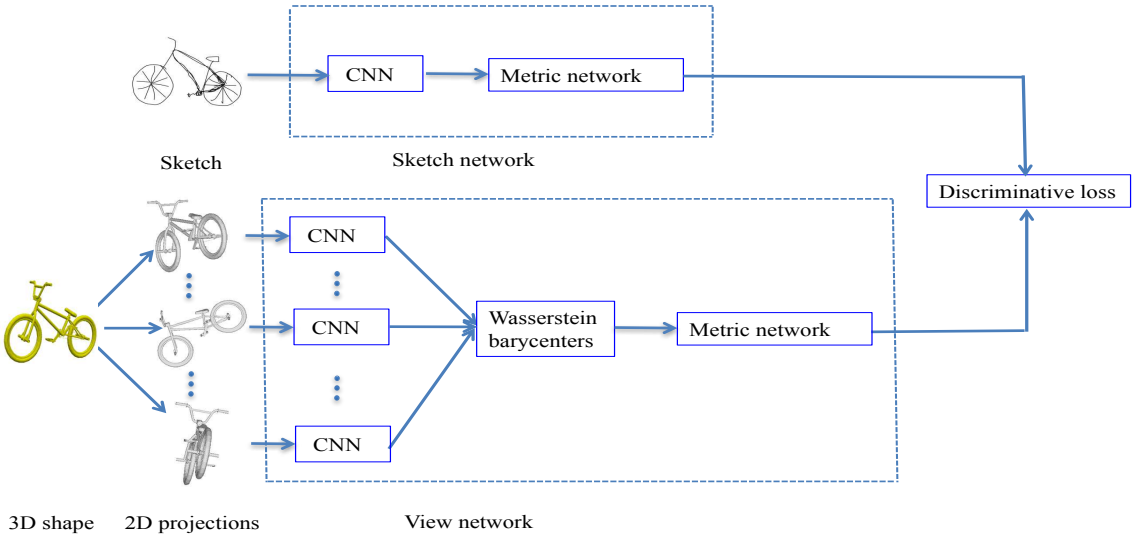

Figure 2. The cross-domain matching framework for sketch-based 3D shape retrieval. By rendering 3D shapes at multiple views, we extract deep CNN features of 2D projections. The Wasserstein barycenters of the deep CNN features are computed to represent 3D shapes. With the metric network of fully connected layers, we then formulate a discriminative loss to learn sketch and shape features for cross-domain retrieval.

Let $\|\boldsymbol{z}_j^2 - \boldsymbol{z}_i^1\|_2^2$ be $J_1(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$. For the sketch network, $\frac{\partial J_1(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}{\partial \boldsymbol{W}_2^k}$, $\frac{\partial J_1(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}{\partial \boldsymbol{b}_2^k}$, $\frac{\partial \boldsymbol{L}_2}{\partial \boldsymbol{W}_2^k}$ and $\frac{\partial \boldsymbol{L}_2}{\partial \boldsymbol{b}_2^k}$ can be easily computed with the back-propagation method. In the view network, the Wasserstein barycenters $\boldsymbol{h}_i$ (the input of the metric network) are computed from the deep CNN features $\boldsymbol{x}_{i,j}$ (the final fully connected layer) of multiple views. Therefore, $\frac{\partial J_1(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}{\partial \boldsymbol{x}_{i,j}}$ can be calculated from $\frac{\partial J_1(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}{\partial \boldsymbol{h}_i}$. Suppose that the Wasserstein barycenters $\boldsymbol{h}_i$ can converge after Q iterations. Thus, from Eq. (6), we can have

$$\frac{\partial J_1(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}{\partial \boldsymbol{x}_{i,j}} = \frac{1}{V}\boldsymbol{A}^T \frac{\partial J_1(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}{\partial \boldsymbol{h}_i}$$
$$\boldsymbol{A} = [(\frac{\boldsymbol{K}^T\boldsymbol{x}_{i,j}}{\boldsymbol{K}\boldsymbol{c}_j^Q})^{\frac{1}{V}-1} \cdot \frac{(\boldsymbol{K}^T)_1}{\boldsymbol{K}\boldsymbol{c}_j^Q} \cdot \prod_{j'=1;j'\neq j}^{V} (\boldsymbol{K}^T\boldsymbol{a}_{j'}^Q)^{\frac{1}{V}}$$
$$, \cdots, (\frac{\boldsymbol{K}^T\boldsymbol{x}_{i,j}}{\boldsymbol{K}\boldsymbol{c}_j^Q})^{\frac{1}{V}-1} \cdot \frac{(\boldsymbol{K}^T)_L}{\boldsymbol{K}\boldsymbol{c}_j^Q} \cdot \prod_{j'=1;j'\neq j}^{V} (\boldsymbol{K}^T\boldsymbol{a}_{j'}^Q)^{\frac{1}{V}}] \tag{11}$$

where $(\boldsymbol{K}^T)_l$ is the lth column of the matrix $\boldsymbol{K}^T$, $l = 1, 2, \cdots, L$, $\frac{\boldsymbol{x}}{\boldsymbol{y}}$, $\prod \boldsymbol{x}_i$ and $\boldsymbol{x} \cdot \boldsymbol{y}$ are the element-wise operations. The calculation of $\frac{\partial J_1(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}{\partial \boldsymbol{x}_{i,j}}$ is summarized in Algorithm. 1. Similarly, $\frac{\partial L_1}{\partial \boldsymbol{x}_{i,j}}$ can also be computed from $\frac{\partial L_1}{\partial \boldsymbol{h}_i}$. Once $\frac{\partial J_1(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}{\partial \boldsymbol{x}_{i,j}}$ and $\frac{\partial L_1}{\partial \boldsymbol{x}_{i,j}}$ are obtained, $\frac{\partial J_1(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}{\partial \boldsymbol{W}_1^k}$, $\frac{\partial J_1(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}{\partial \boldsymbol{b}_1^k}$, $\frac{\partial L_1}{\partial \boldsymbol{W}_1^k}$ and $\frac{\partial L_1}{\partial \boldsymbol{b}_1^k}$ can be computed with the back-propagation method in the CNN.

Once the view and sketch networks are trained, the outputs of the networks are used as the final features of 3D shapes and sketches. For each query sketch, the Euclidean distance between the sketch and 3D shape features is then used as similarity for sketch-based 3D shape retrieval.

Algorithm 1 Calculation of the gradient $\frac{\partial J_1(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}{\partial \boldsymbol{x}_{i,j}}$ in the view network .

Input: CNN features $\boldsymbol{x}_{i,j}$; view number V; kernel matrix $\boldsymbol{K}$; gradient $\frac{\partial J_1(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}{\partial \boldsymbol{h}_i}$.

Output: gradient $\frac{\partial J_1(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}{\partial \boldsymbol{x}_{i,j}}$.

Initialize $\boldsymbol{a}_j^1$: $\boldsymbol{a}_j^1 = \boldsymbol{1}, j = 1, 2, \cdots, V$.

For $t = 1, 2, \cdots, Q$:

1. Compute $\boldsymbol{h}_i$: $\boldsymbol{h}_i = \prod_{j=1}^{V} (\boldsymbol{K}^T\boldsymbol{a}_j^t)^{\frac{1}{V}}$;
2. Compute $\boldsymbol{c}_j^{t+1}$ and $\boldsymbol{a}_j^{t+1}$:
   $\boldsymbol{c}_j^{t+1} = \frac{\boldsymbol{h}_i}{\boldsymbol{K}^T\boldsymbol{a}_j^{t+1}}$;
   $\boldsymbol{a}_j^{t+1} = \frac{\boldsymbol{x}_{i,j}}{\boldsymbol{K}\boldsymbol{c}_j^{t+1}}$.

Calculate the gradient $\frac{\partial J_1(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}{\partial \boldsymbol{x}_{i,j}}$ with Eq. (11) until the difference between $\boldsymbol{h}_i$ in successive iterations is smaller than a setting threshold.

4. Experimental Results

In this section, we first evaluate our learned Wasserstein barycentric representation method for sketch-based 3D shape retrieval, and then compare it to the state-of-the-art sketch-based 3D shape retrieval methods on two benchmark datasets, i.e., SHREC'13 [13] and SHREC'14 [14] sketch track benchmark datasets.

4.1. Datasets and experimental settings

We test our proposed method on the SHREC'13 and SHREC'14 sketch track benchmark datasets. The SHREC'13 sketch-based 3D shape retrieval benchmark dataset is built on a collection of human-drawn sketches [15] and the Princeton shape benchmark (PSB) [19]. The human-drawn sketch dataset consists of 20000 sketches of 250 classes, each with 80 sketches. The PSB dataset includes the train and test sub-datasets, where there are 903 3D shapes with 92 and 90 classes, respectively. By finding the shared classes in both human sketch dataset and PSB dataset, the SHREC'13 sketch track benchmark dataset is constructed, including 7200 sketches and 1258 3D shapes from 90 classes. For each class, these sketches are divided into two subsets: 50 samples for training and 30 samples for testing. Fig. 3 shows sketch examples and corresponding shapes in this dataset.

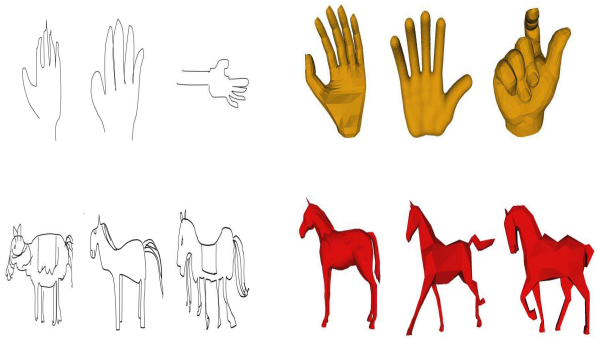

Figure 3. Sketch examples and corresponding 3D shapes in the SHREC'13 sketch track benchmark dataset.

The SHREC'14 sketch track benchmark dataset is a large-scale dataset, which contains 13680 sketches and 8987 3D shapes from 171 classes. There are 80 sketches per class while there are about 53 3D models per class on average. The sketches are further split into the training and testing sub-datasets, which contain 8550 and 5130 sketches, respectively. Fig. 4 shows some sketch and corresponding 3D shape examples in this dataset. From this figure, one can see that for each class there are large variations with 2D sketches and 3D shapes. Due to the large within-class variations in this dataset, the sketch-based shape retrieval task is very challenging.

In our proposed method, for each 3D shape, 12 uniformly sampled rendered views every 30 degrees are used to obtain 2D projections, i.e., $V = 12$. In the sketch and view networks, the deep CNN features are extracted from the "fc7" layer of the AlexNet, whose feature size is 4096. For the metric network, the layers are set as 4096-1000-300-100. Moreover, in Eq. (8), parameters α, β_1 and β_2 are set to 1, 0.0001 and 0.001, respectively. For each training sketch sample, we randomly chose two shape samples from the same class as the positive samples and chose 12 shape samples from different classes as the negative samples. For the calculation of the Wasserstein barycenters, the Euclidean distance is used as the ground metric and γ is set to 80 to compute the matrix $\boldsymbol{K}$. The parameters in the AlexNet are initialized by the model pre-trained on ImageNet images from 1K categories and the parameters in the metric network are randomly initialized. Then, the AlexNet and metric network are jointly learned via back propagation to train the parameters.

Figure 4. Sketch examples and corresponding 3D shapes in the SHREC'14 sketch track benchmark dataset.

We follow the experimental settings in [13] to use 50 sketches as the training samples and 30 sketches as the queries per class. Moreover, the training samples and the query samples do not overlap. We use the following criteria to evaluate our proposed method: precision-recall curve (PR curve), nearest neighbor (NN), first tier (FT), second tier (ST), E-measure (E), discounted cumulated gain (DCG) and mean average precision (mAP).

4.2. Evaluation of the proposed method

In order to demonstrate the effectiveness of the proposed method, we compare the proposed method to the max view-pooling operation of CNN features from multiple views on the SHREC'14 benchmark dataset. In [23], the authors proposed a multi-view CNN structure for 3D shape retrieval. In the last convolutional layer, the element-wise maximum operation across views is performed to form a max view-pooling layer. The feature in the "fc7" layer is used as the shape descriptor for 3D shape retrieval. We perform the same max view-pooling operation across views in our view network to obtain the deep CNN features to represent 3D shapes. Then we use the constructed metric network to learn the final sketch and shape features for sketch-based 3D shape retrieval.

We compare our learned Wasserstein barycentric representation method to the max view-pooling operation in

the multi-view CNN on the SHREC'14 benchmark dataset for sketch-based shape retrieval. The learned Wasserstein barycentric representation method for sketch-based shape retrieval is denoted by LWBR. Fig. 5 shows the precision-recall curves for the max view-pooling operation and the proposed LWBR method. As can be seen in this figure, compared to the max view-pooling operation, our proposed LWBR can obtain better performance. In the max view-pooling method, the maximum element-wise operation across views cannot exploit the information of all views simultaneously. Nonetheless, in our proposed method, the Wasserstein barycentric representation can be viewed as a highly non-linear transform performed on all views, which can make full use of the information of all views. In the transformed non-linear space, the Wasserstein barycentric representation may characterize the manifold where the multi-view features lie better, which can lead to better retrieval performance.

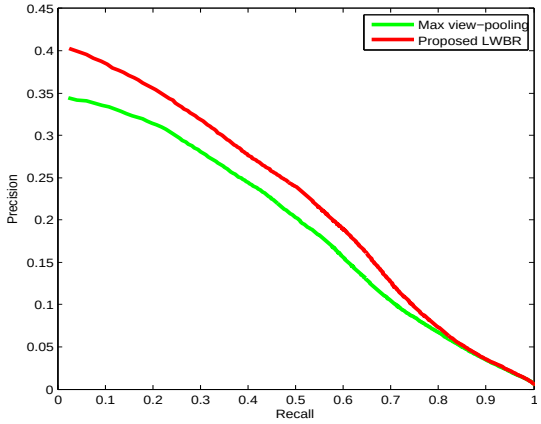

Figure 5. The precision-recall curves for the max view-pooling operation and proposed LWBR method on the SHREC'14 benchmark dataset.

4.3. Comparison with state-of-the-art methods

4.3.1 SHREC'13 benchmark dataset

For the SHREC'13 sketch track benchmark dataset, we compare our proposed LWBR method to several state-of-the-art methods: Fourier descriptors on 3D model silhouettes (FDC) [12], spatial proximity method (SP) [22], edge-based Fourier spectra descriptor (EFSD) [12], sketch-based retrieval method with view clustering (SBR-VC) [12], cross domain manifold ranking method (CDMR) [10] and Siamese network (Siamese) [27]. We evaluate these methods using PR curve, NN, FT, ST, E, DCG and mAP. The PR curves for the FDC, EFSD, SBR-VC and proposed LWBR methods are plotted in Fig. 6. From this figure, one can see that the performance of the proposed LWBR method is significantly superior to that of these methods. We also compare our proposed LWBR method to the CDMR, SBR-VC, SP, FDC and Siamese methods with NN, FT, ST, E, DCG and mAP. The comparison results are listed in Table . 1. Compared to these methods, our proposed LWBR method can significantly improve the retrieval performance.

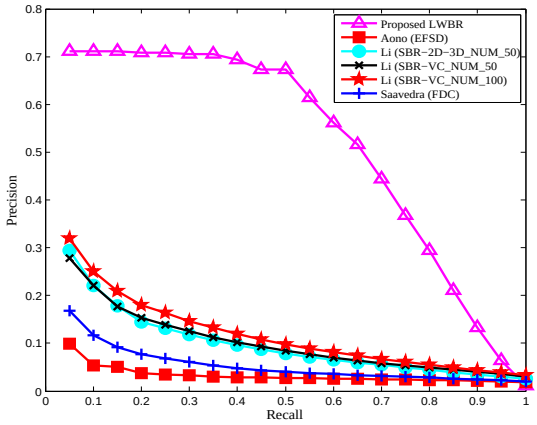

Figure 6. The precision-recall curves for the FDC, EFSD, SBR-VC and proposed LWBR methods on the SHREC'13 benchmark dataset.

Table 1. Retrieval results on the SHREC'13 benchmark dataset.

Methods	NN	FT	ST	E	DCG	mAP
CDMR [10]	0.279	0.203	0.296	0.166	0.458	0.250
SBR-VC [12]	0.164	0.097	0.149	0.085	0.348	0.116
SP [22]	0.017	0.016	0.031	0.018	0.240	0.026
FDC [12]	0.110	0.069	0.107	0.061	0.307	0.086
Siamese [27]	0.405	0.403	0.548	0.287	0.607	0.469
LWBR	**0.712**	**0.725**	**0.785**	**0.369**	**0.814**	**0.752**

Among these methods, CDMR, Siamese and the proposed LWBR methods employ metric learning for sketch-based 3D shape retrieval. In the CDMR and Siamese methods, the features from multiple views are learned separately and the minimum distance between the query sketch and multiple views of the 3D shape in the feature space is used as similarity for retrieval, which means that the views are independently used to characterize 3D shapes. Different from these methods, by learning the Wasserstein barycenters of multiple views, our proposed LWBR method can simultaneously aggregate the information of all views to form a discriminative representation for 3D shapes. Thus, our proposed LWBR method can obtain better performance. For example, our proposed method can obtain the mAP of 0.752 while the CDMR and Siamese methods can obtain the mAPs of 0.250 and 0.469, respectively.

4.3.2 SHREC'14 benchmark dataset

We also compare our proposed LWBR method to several state-of-the-art methods on the SHREC'14 sketch track benchmark dataset: cross domain manifold ranking method (CDMR) [10], sketch-based retrieval method with view

clustering (SBR-VC) [12], depth-buffered vector of locally aggregated tensors (DB-VLAT) [24], overlapped pyramid of histograms of orientation gradients (SCMR-OPHOG) [14], BOF junction-based extended shape context (BOF-JESC) [14], Siamese network (Siamese) [27]. We also use PR curve, NN, FT, ST, E, DCG and mAP to evaluate these methods. From the PR curves plotted in Fig. 7, we can see that when the recall is less than about 0.75 the precision of the proposed method is higher than that of these methods. Nonetheless, when the recall is larger than about 0.75, the precision of the proposed method is slightly lower than that of the SCMR-OPHOG method. The comparison results with NN, FT, ST, E, DCG and mAP are also listed in Table. 2 . Although the SHREC'14 sketch track benchmark dataset is very challenging, our proposed method can significantly outperform these methods. Particularly, in comparison to the deep learning based method such as Siamese, the proposed LWBR can achieve the mAP of 0.401 while the Siamese method can achieve the mAP of 0.228.

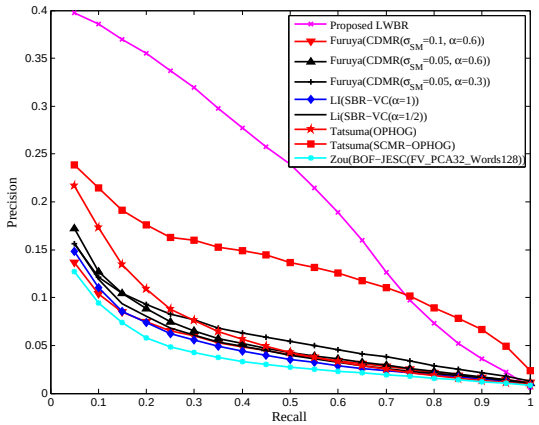

Figure 7. The precision-recall curves for the CDMR, SBR-VC, SCMR-OPHOG, BOF-JESC and proposed LWBR methods on the SHREC'14 benchmark dataset.

Table 2. Retrieval results on the SHREC'14 benchmark dataset.

Methods	NN	FT	ST	E	DCG	mAP
CDMR [10]	0.109	0.057	0.089	0.041	0.328	0.054
SBR-VC [12]	0.095	0.050	0.081	0.037	0.319	0.050
DB-VLAT [24]	0.160	0.115	0.170	0.079	0.376	0.131
Siamese [27]	0.239	0.212	0.316	0.140	0.496	0.228
LWBR	**0.403**	**0.378**	**0.455**	**0.236**	**0.581**	**0.401**

Finally, we conduct sketch-based 3D shape retrieval experiments on the SHREC'14 benchmark dataset by setting different numbers of rendered views. Different numbers of rendered views can be obtained by varying the number of placed virtual cameras around the 3D shape. In this experiment, we set the view number V to 3, 4, 6 and 12, corresponding to placing virtual cameras every 120, 90, 60 and 30 degrees. In the cases of 3, 4, 6 and 12 rendered views, the final mAPs are 0.325, 0.363, 0.390 and 0.401, respectively. One can see that when the view number increases the retrieval performance can be improved. Nonetheless, if the rendered views are too much, the computational cost is expensive.

5. Conclusions

In this paper, we proposed to learn Wasserstein barycentric representations of 3D shapes for sketch-based 3D shape retrieval. We employed two AlexNets to extract deep CNN features of sketches and projections of 3D shapes, respectively. The Wasserstein barycenters of the deep CNN features of multiple projections are computed to represent 3D shapes. We then constructed a metric network to learn the Wasserstein barycenters for retrieval by formulating a discriminative loss across the sketch and shape domains. The outputs of the metric network are used as the final features for retrieval. Experimental results demonstrate that our proposed method can yield good performance on the SHREC'13 and SHREC'14 sketch track benchmark datasets.

In future, we will investigate to employ the anisotropic Wasserstein barycentric representations to characterize multiple projections of 3D shapes. Moreover, we will study how to effectively learn the anisotropic Wasserstein barycenters of 3D shapes for retrieval.

References

[1] S. Bai, X. Bai, Z. Zhou, Z. Zhang, and L. Jan Latecki. Gift: A real-time and scalable 3D shape search engine. In *IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA*, pages 5023–5032, 2016.

[2] J. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal. Scientific Computing*, 37(2), 2015.

[3] V. I. Bogachev and A. V. Kolesnikov. The monge-kantorovich problem: achievements, connections, and perspectives. *Russian Math Surveys*, 67(5):785, 2012.

[4] N. Bonneel, G. Peyré, and M. Cuturi. Wasserstein barycentric coordinates: histogram regression using optimal transport. *ACM Trans. Graphics*, 35(4):71, 2016.

[5] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems, Lake Tahoe, Nevada, USA*, pages 2292–2300, 2013.

[6] M. Cuturi and A. Doucet. Fast computation of wasserstein barycenters. In *The International Conference on Machine Learning, Beijing, China*, pages 685–693, 2014.

[7] M. Cuturi and G. Peyré. A smoothed dual approach for variational wasserstein problems. *SIAM Journal. Imaging Sciences*, 9(1):320–343, 2016.

[8] T. Darom and Y. Keller. Scale-invariant features for 3-D mesh models. *IEEE Trans. Image Processing*, 21(5):2758–2769, 2012.

[9] S. Ferradans, G. Xia, G. Peyré, and J. Aujol. Static and dynamic texture mixing using optimal transport. In *Scale Space and Variational Methods in Computer Vision, Schloss Seggau, Leibnitz, Austria*, pages 137–148, 2013.

[10] T. Furuya and R. Ohbuchi. Ranking on cross-domain manifold for sketch-based 3D model retrieval. In *International Conference on Cyberworlds, Yokohama, Japan*, pages 274–281, 2013.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems, Lake Tahoe, Nevada, USA*, pages 1106–1114, 2012.

[12] B. Li, Y. Lu, A. Godil, T. Schreck, M. Aono, H. Johan, J. M. Saavedra, and S. Tashiro. Shrec'13 track: Large scale sketch-based 3D shape retrieval. In *Eurographics Workshop on 3D Object Retrieval, Girona, Spain*, pages 89–96, 2013.

[13] B. Li, Y. Lu, A. Godil, T. Schreck, B. Bustos, A. Ferreira, T. Furuya, M. J. Fonseca, H. Johan, T. Matsuda, R. Ohbuchi, P. B. Pascoal, and J. M. Saavedra. A comparison of methods for sketch-based 3D shape retrieval. *Computer Vision and Image Understanding*, 119:57–80, 2014.

[14] B. Li, Y. Lu, C. Li, A. Godil, T. Schreck, M. Aono, M. Burtscher, H. Fu, T. Furuya, H. Johan, J. Liu, R. Ohbuchi, A. Tatsuma, and C. Zou. Extended large scale sketch-based 3D shape retrieval. In *Eurographics Workshop on 3D Object Retrieval, Strasbourg, France*, pages 121–130, 2014.

[15] E. Mathias, R. Ronald, B. Tamy, H. Kristian, and A. Marc. Sketch-based shape retrieval. *ACM Trans. Graphics*, 31(4):31:1–31:10, 2012.

[16] A. Rolet, M. Cuturi, and G. Peyré. Fast dictionary learning with a smoothed wasserstein loss. In *The International Conference on Artificial Intelligence and Statistics, Cadiz, Spain*, pages 630–638, 2016.

[17] Y. Rubner, C. Tomasi, and L. J. Guibas. The Earth Mover's Distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[18] B. Shi, S. Bai, Z. Zhou, and X. Bai. Deeppano: Deep panoramic representation for 3-D shape recognition. *IEEE Trans. Signal Processing Letters*, 22(12):2339–2343, 2015.

[19] P. Shilane, P. Min, M. M. Kazhdan, and T. A. Funkhouser. The princeton shape benchmark. In *International Conference on Shape Modeling and Applications, Genova, Italy*, pages 167–178, 2004.

[20] R. Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74(4):402–405, 1967.

[21] J. Solomon, F. de Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. J. Guibas. Convolutional wasserstein distances: efficient optimal transportation on geometric domains. *ACM Trans. Graphics*, 34(4):66, 2015.

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

[23] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *IEEE International Conference on Computer Vision, Santiago, Chile*, pages 945–953, 2015.

[24] A. Tatsuma, H. Koyanagi, and M. Aono. A large-scale shape benchmark for 3D object retrieval: Toyohashi shape benchmark. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, Hollywood, CA, USA*, pages 1–10, 2012.

[25] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[26] C. Villani. *Optimal transport : old and new*. Springer, Berlin, 2009.

[27] F. Wang, L. Kang, and Y. Li. Sketch-based 3D shape retrieval using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA*, pages 1875–1883, 2015.

[28] F. Zhu, J. Xie, and Y. Fang. Learning cross-domain neural networks for sketch-based 3D shape retrieval. In *AAAI Conference on Artificial Intelligence, Phoenix, Arizona, USA*, pages 3683–3689, 2016.