

Article

# Deep Common Semantic Space Embedding for Sketch-Based 3D Model Retrieval

Jing Bai <sup>1,2,\*</sup>, Mengjie Wang <sup>1</sup> and Dexin Kong <sup>1</sup>

<sup>1</sup> School of Computer Science and Engineering, North Minzu University, Yinchuan 750021, China; sunny\_mj@163.com (M.W.); wow.ak47@163.com (D.K.)

<sup>2</sup> Ningxia Province Key Laboratory of Intelligent Information and Data Processing, Yinchuan 750021, China

\* Correspondence: baijing@nun.edu.cn

Received: 4 March 2019; Accepted: 2 April 2019; Published: 4 April 2019



**Abstract:** Sketch-based 3D model retrieval has become an important research topic in many applications, such as computer graphics and computer-aided design. Although sketches and 3D models have huge interdomain visual perception discrepancies, and sketches of the same object have remarkable intradomain visual perception diversity, the 3D models and sketches of the same class share common semantic content. Motivated by these findings, we propose a novel approach for sketch-based 3D model retrieval by constructing a deep common semantic space embedding using triplet network. First, a common data space is constructed by representing every 3D model as a group of views. Second, a common modality space is generated by translating views to sketches according to cross entropy evaluation. Third, a common semantic space embedding for two domains is learned based on a triplet network. Finally, based on the learned features of sketches and 3D models, four kinds of distance metrics between sketches and 3D models are designed, and sketch-based 3D model retrieval results are achieved. The experimental results using the Shape Retrieval Contest (SHREC) 2013 and SHREC 2014 datasets reveal the superiority of our proposed method over state-of-the-art methods.

**Keywords:** sketch-based retrieval; 3D model retrieval; deep common semantic space embedding; metric learning; cross-entropy

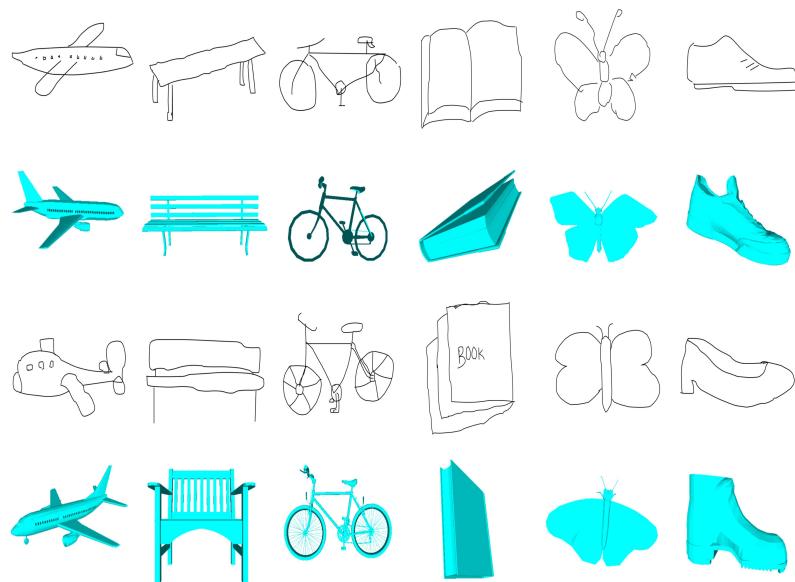
---

## 1. Introduction

With the rapid development of computer hardware, 3D data acquisition, and shape modeling technologies, 3D models have become increasingly useful in various fields, and as a result, 3D model retrieval and reuse has received increasing attention. There are two key challenges to retrieving a 3D model: creating a model that is simple enough for novice users and producing a matching algorithm that is robust enough to work for arbitrary polygonal models [1]. Accordingly, example-based 3D model retrieval has attracted widespread attention. Unfortunately, although example-based 3D model retrieval is intuitively straightforward, it is difficult to achieve [2]. Photo input is another natural method for 3D model retrieval when user has the target 3D object. However, user cannot obtain the 3D object in many applications, such as the conceptual design stage of a new product. Instead, due to its intuitive nature and convenience, sketch-based 3D model retrieval plays a practical role in many applications, including sketch-based rapid modeling, recognition, 3D printing and 3D animation production.

However, sketch-based 3D model retrieval is more challenging than example-based retrieval. This difficulty is due to two main characteristics of sketch-based 3D model retrieval: (1) large interdomain visual perception discrepancies between sketches and 3D models (as Figure 1 shows, 3D models are precise and informative, while sketches are concise and abstract) and (2) large

intradomain visual perception diversities for sketches of the same object, as shown in Figure 1. This diversity occurs because a sketch is a subjective expression of an object that is influenced by factors such as the thought processes, domain background and hand-drawing habits of the person creating the sketch. The above two factors are related to each other and lead to difficulty in sketch-based 3D model retrieval.



**Figure 1.** Sketches and the corresponding 3D models.

Although there are large interdomain visual perception discrepancies and significant intradomain visual perception diversity for sketch-based 3D model retrieval, the 3D models and sketches of the same class share common semantic content. Motivated by these findings, we propose a novel approach for sketch-based 3D model retrieval by constructing a deep common semantic space embedding using a triplet network (DCSSE). First, 3D models are described by a group of views. Then, a common modality space is constructed by translating one mode to another based on cross entropy. Finally, a common semantic space embedding is learned based on a triplet network, and the essential features of sketches and 3D models are generated simultaneously by synthetically considering the two domains. The retrieval experiments on Shape Retrieval Contest (SHREC) 2013 and SHREC 2014 demonstrate the effectiveness of our approach.

This paper makes the following contributions. (1) A cross-entropy-based common modality space is constructed for sketches and 3D models, which reduces interdomain visual perception discrepancies. (2) A DCSSE is generated between sketches and 3D models via synthetical consideration of the sketches, the 3D models and their shared semantics. (3) A novel combination of deep metric learning with cross-domain transformation is adopted, which has more relaxed constraints and is more consistent with the two characteristics of sketch-based 3D model retrieval. (4) The approach outperforms all state-of-the-art methods on two large benchmark datasets.

## 2. Related Work

At present, sketch-based 3D model retrieval is the most concerned hot spot in the field of 3D model retrieval. Researchers have proposed a variety of methods for sketch-based 3D model retrieval [3–6]. Earlier methods such as Histogram of Gradient [7], Gabor local line [8], View Context [9,10], part-based features [11,12], cross-domain manifold ranking (CDMR) [13], Shape2Vec [14] and composite features [15] algorithms have extracted handcrafted features for sketches and 3D models. However, due to the complexity of 3D models and the large interdomain visual perception discrepancies, these

methods cannot objectively and reasonably capture the essential features of 3D models and sketches, and consequently, their accuracies are insufficient.

With the wide application of deep learning in computer vision, several deep feature learning-based methods are proposed. As 3D model representation is unstructured, it cannot be directly inputted into the deep learning model. Some of these methods describe 3D models using low-level feature vectors, such as physics-constrained deep neural network (PCDNN) [16], deep correlated metric learning (DCML) [17], deep correlated holistic metric learning (DCHML) [18], and semantic embedding space (SEM) [19], and then input the vectors into the deep learning model to generate the final features. These methods try to automatically learn and construct the features of complex models; however, they lose original information when extracting low-level features and fail to make full use of the characteristics of the deep learning algorithm. In addition, these methods do not consider the relationship between the sketches and the 3D models when extracting their features, resulting in unsatisfactory retrieval accuracies.

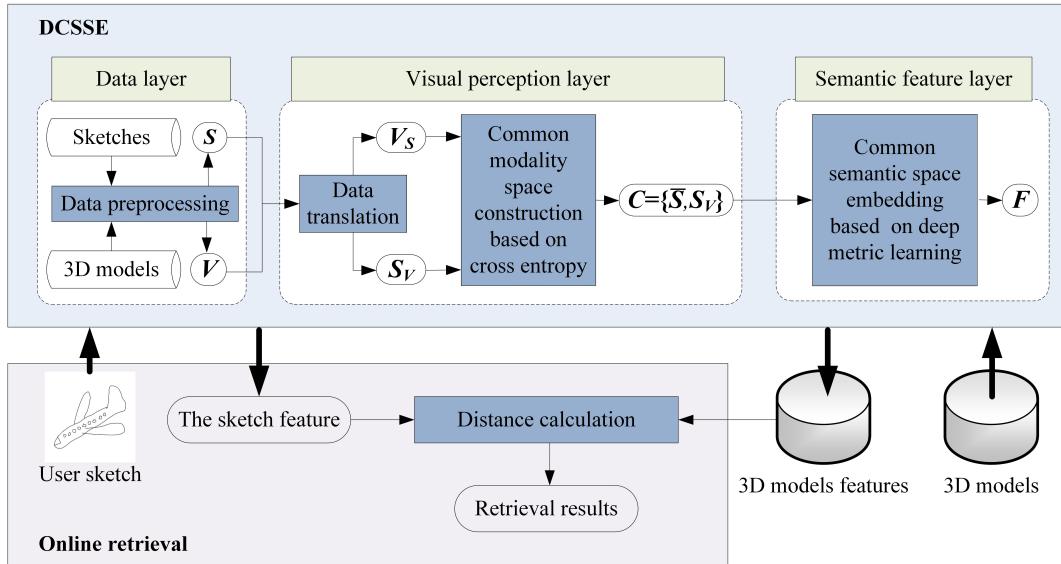
Another kind of deep feature learning-based method for sketch-based 3D model retrieval describes 3D models using a group of projected views, and then separately adopts two convolutional neural networks (CNNs) for the views and sketches, finally combines them by constructing the specific loss between the features of the two domains using methods such as the Siamese network (Siamese) [20], the learned Wasserstein barycentric representation (LWBR) [21], deep cross-modality adaptation (DCA) [22], and multiview attention network (MVAN) [23]. These methods have achieved state-of-the-art performance. However, the sketches are generally quite abstract with large local and global deviations from the original model [3]. As a result, any methods that extract features of views and sketches separately fail to explore the related features between the sketches and the 3D models. In addition, these methods construct a metric network by minimizing the loss to compel the features of the sketches and the 3D models of the same class to be identical or almost identical, which may be too restrictive for heterogeneous data.

### 3. Proposed Method

As illustrated in Figure 2, the proposed approach consists of two parts: the DCSSE model and online retrieval.

- (1) DCSSE is composed of three layers: the data layer, the visual perception layer and the semantic feature layer. The inputs of DCSSE are the labeled 3D model dataset  $M_i = \{M_i, 1 \leq i \leq n_m\}$  and the sketch dataset  $S = \{S_i, 1 \leq i \leq n_s\}$ , where  $n_m$  is the number of 3D models and  $n_s$  is the number of sketches. In the data layer, a 3D model  $M_i$  can be represented as  $l$  views  $\{V_i^j, 1 \leq j \leq l\}$  by data preprocessing, which ensures both the inputted 3D models and the sketches occupy 2D space and share the same data form. Then, in the visual perception layer, data translation is used to construct a common modality space based on cross entropy, impelling all inputted data to share a similar visual perception. Furthermore, in the semantic feature layer, for any 3D model  $M_i$  and any sketch  $S_i$ , their feature embeddings  $\{f(V_i^j), 1 \leq j \leq l\}$  and  $f(S_i)$  can be generated after deep metric learning, and a common semantic space  $F$  is constructed.
- (2) The online retrieval: Based on DCSSE, we can further extract the features of the user sketch and 3D models, calculate the distance between the user sketch and 3D models and return the retrieval results.

In the remainder of this section, we will elaborate on each step.



**Figure 2.** Framework of our proposed approach.

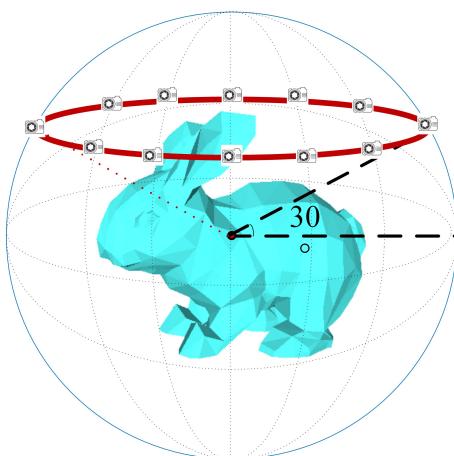
### 3.1. The Data Layer: Intradomain Data Preprocessing

Given the 3D model dataset  $M$  and the sketch dataset  $S$ , the purpose of data preprocessing in the data layer is to uniformly convert the data of the two different modalities into 2D images with the same data form.

#### 3.1.1. 3D Model Preprocessing

For the 3D model dataset  $M$ , a mapping function  $\phi_m : M \rightarrow \mathbf{R}^{m \times m \times l}$  is established and a multiview representation set  $V = \left\{ \phi_m(M_i) = \left\{ V_i^j, 1 \leq j \leq l, V_i^j \in \mathbf{R}^{m \times m} \right\}, 1 \leq i \leq n_m \right\}$  is obtained by 3D model preprocessing, where  $l$  is the number of views (images) and  $m \times m$  is the size of the views. Comparing all kinds of multiview 3D model retrieval algorithms, we can see that the view-rendering method proposed by Multi-view Convolutional Neural Networks (Su-MVCNN) is excellent [24,25]. Therefore, in this paper, we use this method to construct a multiview representation of a given 3D model.

Taking 12 views as an example, the multiview rendering of a 3D model is shown in Figure 3. First, the 3D model is normalized into a unit sphere via translation and scaling. Second, as shown in Figure 3, along the red circle on the unit sphere surface, 12 virtual cameras are set at an interval of 30 degrees, and the cameras are positioned to point toward the sphere center. Finally, each view is rendered using the Phong reflection model [26].



**Figure 3.** Multiview rendering of 3D models.

As shown in Figure 3, because the views are uniformly located in different views of the 3D model, there is a strong mutual relationship between them. In addition, there is strong complementarity and low correlation between views; thus, the multiview representation obtained by this method constitutes a relatively complete description of the 3D model.

### 3.1.2. 2D Sketch Preprocessing

For the sketch dataset  $S$ , a mapping function  $\phi_s : S \rightarrow \mathbf{R}^{m \times m}$  is established and a 2D image  $\bar{S} = \{\phi_s(S_i) = \bar{S}_i, \bar{S}_i \in \mathbf{R}^{m \times m}, 1 \leq i \leq n_s\}$  is obtained by sketch preprocessing, where  $m \times m$  is the size of the output images. Here, we directly use the bilinear interpolation algorithm to complete the size transformation of the sketch image.

## 3.2. The Visual Perception Layer: Interdomain Data Translation

Although both the 3D models and the hand-drawn sketches have been represented in a 2D space with the same image sizes by intradomain data preprocessing, they are significantly different in terms of visual perception. Specifically, the multiview representations are accurate and informative, while the hand-drawn sketches are concise and abstract. To further narrow the interdomain differences and highlight the commonality of categories, we analyze the characteristics of views and sketches, achieve sketch-to-view and view-to-sketch interdomain translation, evaluate translation validity using cross entropy, and construct a 2D common modality space in this subsection.

### 3.2.1. View-To-Sketch Translation

Since views contain complete visual information from a certain perspective of a 3D model and hand-drawn sketches mainly contain outline information of the 3D model, the translation from view-to-sketch can be accomplished by choosing a reasonable edge extraction algorithm. In this work, the model in [8], which detects Canny edges on the depth buffer (also known as the z-buffer), is used to translate the multiview representation set  $V$  to the extended sketch set  $S_V = \{\tau_{v-s}(V_i^j), 1 \leq i \leq n_m, 1 \leq j \leq l\}$ .

### 3.2.2. Sketch-To-View Translation

Compared with the views of the 3D model, the information contained in sketches is abstract, finite, and ambiguous, making it extremely difficult to generate high-quality views directly from sketches. In recent years, GAN [27], the generative adversarial network, has been proposed for high-quality image-to-image translation. Accordingly, inputting standardized sketch set  $\bar{S}$  without any paired data, we achieve sketch-to-view translation and generate the extended view set  $V_S = \{\tau_{s-v}(\bar{S}_i), 1 \leq i \leq n_s\}$  using CycleGAN [28].

### 3.2.3. Construction of 2D Common Modality Space Based on Cross Entropy

Some translation results of the view-to-sketch and sketch-to-view translations using the methods mentioned in Sections 3.2.1 and 3.2.2 are shown in Figure 4. Compared to the original sketches, the sketch translated from a given view loses some of the local detail information, such as the eyes of the dolphin; compared to original views, the view translated from a given sketch has problems with missing or additional texture information. In conclusion, neither translation is perfect.

To evaluate which kind of translation is more effective for the construction of the 2D common modality space to support cross-domain retrieval, cross-entropy [29], a metric of the difference between two probability distributions, is introduced to evaluate the two kinds of translation. Specifically, the gray-level histograms of images are used as statistical features. Let  $r = 0, \dots, 255$  be the gray value,  $V(k, r)$  be the average probability value of all views corresponding to the  $k$ -class 3D models in the database when their gray value is  $r$ ,  $S_V(k, r)$  be the average probability value of all sketches translated from views of the  $k$ -class 3D models when the gray value is  $r$ ,  $S(k, r)$  be the average probability value

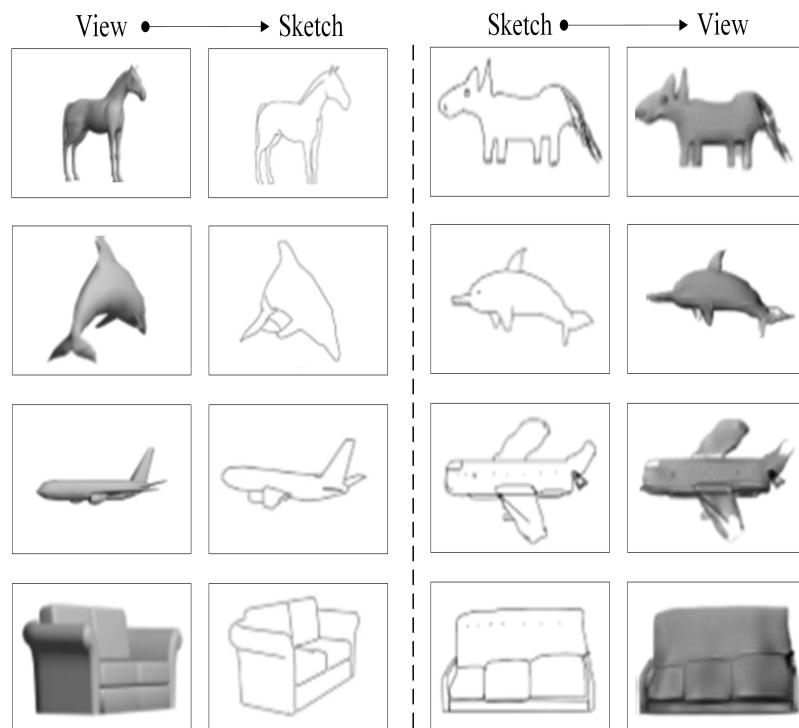
of all  $k$ -class sketches in the database when their gray value is  $r$ , and  $V_S(k, r)$  be the average probability value of all views translated from  $k$ -class sketches when the grayscale value is  $r$ . The cross entropy of view-to-sketch translation is defined as follows:

$$H(S, S_V) = \sum_{k=1}^c \sum_{r=0}^{255} S(k, r) \log \frac{1}{S_V(k, r)} \quad (1)$$

where  $c$  is the number of sketch classes in the database (equal to the number of 3D model classes). The cross entropy of sketch-to-view translation is defined as follows:

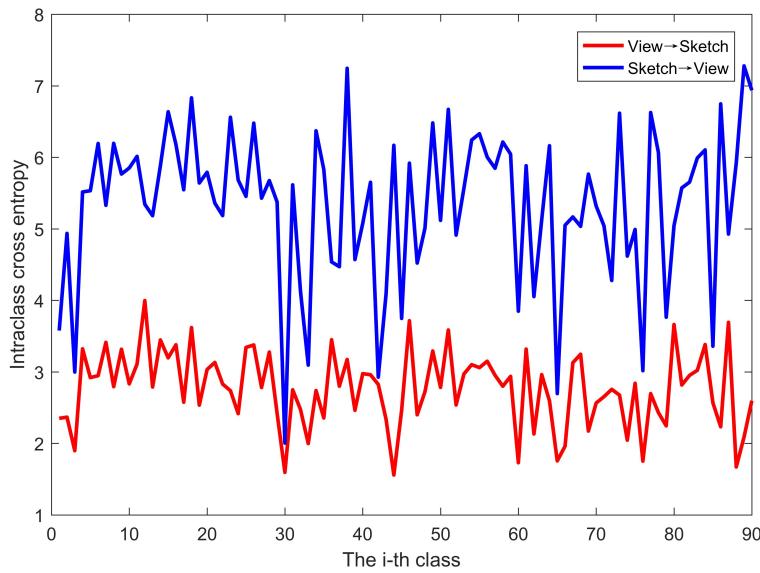
$$H(V, V_S) = \sum_{k=1}^c \sum_{r=0}^{255} V(k, r) \log \frac{1}{V_S(k, r)} \quad (2)$$

where  $c$  is the number of sketch classes in the database (equal to the number of 3D model classes).



**Figure 4.** Translation results of the view-to-sketch and sketch-to-view translations.

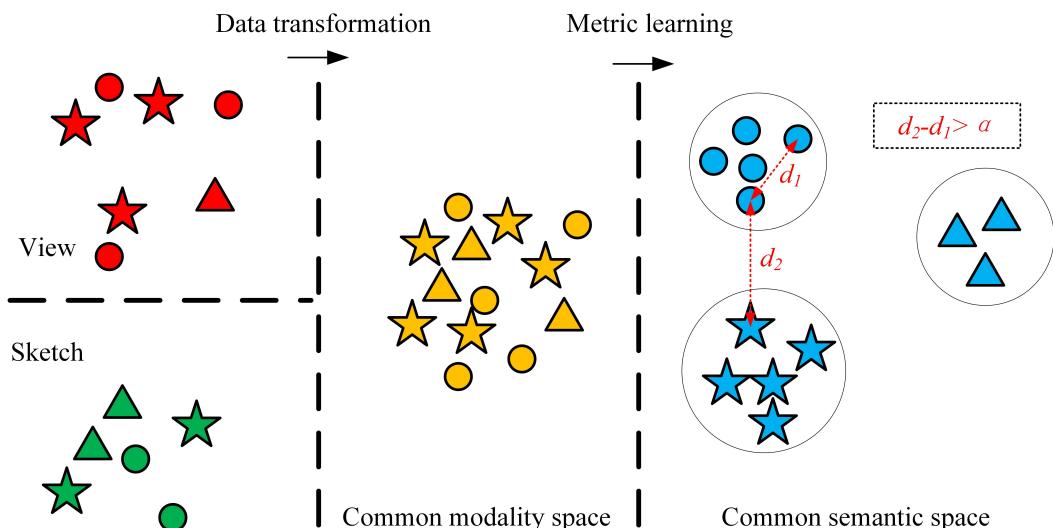
Figure 5 shows the intraclass cross entropy values of the view-to-sketch translation and sketch-to-view translations on the SHREC 2013 dataset. It can be seen that, regardless of the class to which the images belong, the cross-entropy values of the view-to-sketch translation are smaller than those of the sketch-to-view translation, demonstrating that the sketches translated by views can better simulate the probability distribution of the original sketches. We performed the same experiment on the SHREC14 dataset, and the results are the same. Based on this discovery, we choose the view-to-sketch translation to construct the common modality space  $C = \bar{S} \cup S_V$  for sketches and 3D models to impel the inputted data to share a similar visual perception.



**Figure 5.** Comparison of intra-class cross entropy.

### 3.3. The Semantic Feature Layer: Cross-Domain Common Semantic Space Embedding

As shown in Figure 6, the modality sharing between 3D models and sketches has been constructed through the interdomain data translation; however, data of different classes in this common modality space are mixed and indistinguishable. To solve this issue, a deep metric learning model is introduced to build a common feature space, narrowing the distance of the same class and widening the sample distance of different classes.



**Figure 6.** Illustration of the common modality space and the common semantic space. The color represents the modality and the shape represents the class. Input data come from two modalities (i.e., sketches and views of 3D shapes). First, the two types of input data are translated into a common modality space, where data of different modalities share a common space; however, data in different classes are mixed and indistinguishable. Then, a common semantic space is constructed using deep metric learning, where the same kinds of samples are nearer and different kinds of samples are farther.

Typical deep metric learning includes Siamese and triplet networks. Compared with the Siamese network with contrastive loss, the triplet network [30] learns a ranking function for retrieval, which has more relaxed constraints. Therefore, considering the large interdomain visual perception discrepancies and the significant intradomain visual perception diversity for sketch-based 3D model retrieval,

the triplet network with deep ranking loss is chosen to construct a common semantic space  $F$  for cross-modality retrieval. As shown in Figure 7, the details are as follows:

- (1) Selection of triples. The anchor training samples in the triples are selected from the normalized sketch dataset  $\bar{S}$ , and positive and negative training samples are selected from the extended sketch dataset  $S_V$ . Positive samples must have the same class as a given anchor training sample, while negative training samples must have a different class.
- (2) Construction of the CNNs. Sketch-based 3D model retrieval is very complex, but the information in each sketch is relatively sparse. Therefore, taking AlexNet as the prototype [31], a medium-sized CNN is constructed. This network consists of eight layers; the first five layers are convolutional layers, the middle two layers are fully connected layers, and the last layer is the feature output layer Feat. The details are shown in Table 1.
- (3) Establishment of the loss function. Given the  $i^{th}$  triplets  $x_i^a, x_i^p, x_i^n$  of the objects, we take  $x_i^a$  to be the anchor sample,  $x_i^p$  to be the same class as  $x_i^a$ ,  $x_i^n$  to be of a different class,  $f(x)$  to be the embedded feature representation of the network, and  $F$  to be the embedded feature space. The metric function should satisfy the following:

$$\begin{aligned} & \|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2 \\ & \forall f(x_i^a), f(x_i^p), f(x_i^n) \in F \end{aligned} \quad (3)$$

where  $\alpha$  is the interval threshold and requires the minimum distance difference between the same class and different classes to be  $\alpha$ .

The corresponding loss function can be expressed as follows:

$$L(f) = \operatorname{argmin}_i \left( \|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right)_+ \quad (4)$$

Here,  $(g)_+ = 0$  if  $g < 0$ , otherwise  $(g)_+ = g$ .

- (4) Implementation.

**Data augmentation.** For each sketch and each view of the 3D models, the image is first resized to [256, 256]. Then, a  $225 \times 225$  image is randomly cropped from the image, or its deformation is rotated 30 degrees forward and backward, with the per-pixel mean subtracted.

**Data shuffle.** Notably, to enhance the generalizability of the network, we randomly disrupted the order of the training data when generating training sets to prevent the same kind of training samples from appearing in a single batch during training.

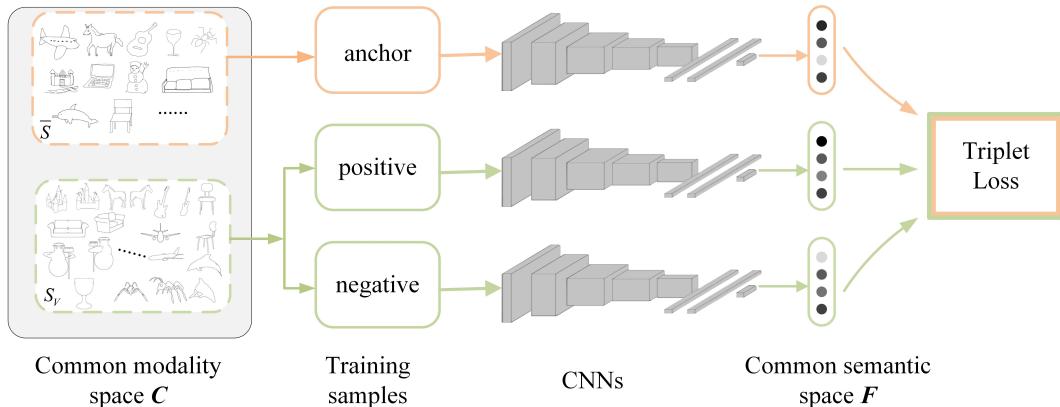
**Training.** DCSSE is trained by the stochastic gradient descent (SGD) + Newton momentum method with a mini-batch size of 125. Here, SGD training is fast and can converge at a faster speed for large datasets; however, there is an instability problem. The introduction of Newton momentum can restrain oscillation and enhance the stability of network learning when the gradient direction before is different from that after iteration. The update formulas are as follows:

$$\theta_{new} = \theta - \lambda \nabla \theta + \varepsilon v_{t-1} \quad (5)$$

where,  $\theta_{new}$  is the set of updated parameters,  $\theta$  is the set of original parameters,  $\lambda$  is the current learning rate,  $\nabla \theta$  is the gradient of the parameters at the current position,  $v_{t-1}$  is the momentum accumulated in all previous steps, and  $\varepsilon$  is the weight of momentum. Here, the learning rate  $\lambda$  is initialized as 0.0001 and  $\varepsilon$  is set to 0.9. Furthermore, as in Formula (6), an algorithm of adaptive learning rates adjusts the network's weight to enhance the convergence speed.

$$\lambda_{new} = \lambda / \sqrt[k]{(1 + \gamma * \alpha)} \quad (6)$$

Here,  $\lambda_{new}$  is the learning rate,  $\lambda$  is the original learning rate,  $\alpha$  is the number of iterations, and  $k$  and  $\gamma$  are parameters used to update the learning rate and are set to 0.75 and 0.0001, respectively.



**Figure 7.** Construction of the common semantic space  $F$  based on triplet convolutional neural networks (CNNs).

**Table 1.** Network structure of the constructed CNN. Abbreviations: Convolution (Conv), Rectified Linear Unit (Relu), Normalization (Norm), Pooling (Pool), Full connected layer (Fc), and Feature (Feat).

Layer	Filter Size	Stride	Pad	Feature Maps	Input Size	Output Size
Conv1	$11 \times 11$	4	0	96	$225 \times 225 \times 1$	$54 \times 54 \times 96$
Relu1	—	—	—	—	$54 \times 54 \times 96$	$54 \times 54 \times 96$
Norm1	—	—	—	—	$54 \times 54 \times 96$	$54 \times 54 \times 96$
Pool1	$3 \times 3$	2	0	96	$54 \times 54 \times 96$	$27 \times 27 \times 96$
Conv2	$5 \times 5$	—	2	256	$27 \times 27 \times 96$	$27 \times 27 \times 256$
Relu2	—	—	—	—	$27 \times 27 \times 256$	$27 \times 27 \times 256$
Norm2	—	—	—	—	$27 \times 27 \times 256$	$27 \times 27 \times 256$
Pool2	$3 \times 3$	2	0	256	$27 \times 27 \times 256$	$13 \times 13 \times 256$
Conv3	$3 \times 3$	0	1	384	$13 \times 13 \times 256$	$13 \times 13 \times 384$
Relu3	—	—	—	—	$13 \times 13 \times 384$	$13 \times 13 \times 384$
Conv4	$3 \times 3$	0	1	384	$13 \times 13 \times 384$	$13 \times 13 \times 384$
Relu4	—	—	—	—	$13 \times 13 \times 384$	$13 \times 13 \times 384$
Conv5	$3 \times 3$	0	1	256	$13 \times 13 \times 384$	$13 \times 13 \times 256$
Relu5	—	—	—	—	$13 \times 13 \times 256$	$13 \times 13 \times 256$
Pool5	$3 \times 3$	2	0	256	$13 \times 13 \times 256$	$6 \times 6 \times 256$
Fc6	—	—	—	—	$6 \times 6 \times 256$	4096
Relu6	—	—	—	—	4096	4096
Dropout6	—	—	—	—	4096	4096
Fc7	—	—	—	—	4096	4096
Relu7	—	—	—	—	4096	4096
Dropout 7	—	—	—	—	4096	4096
Feat	—	—	—	—	4096	200

### 3.4. Cross-Domain Distance Metric

Given a sketch  $S_i \in S$ , let its feature be denoted as the  $d$ -dimensional vector  $x = \{f(S_i) = (x_1, x_2, \dots, x_d), 1 \leq i \leq n_s\}$ ; given a 3D model  $M_k \in M$ , let its feature be the  $l \times d$  dimensional matrix  $y = \{y^j = f(V_k^j) = (y_1^j, y_2^j, \dots, y_d^j), 1 \leq j \leq l, 1 \leq k \leq n_m\}$ , where  $l$  is the number of views,  $n_s$  and  $n_m$  are the number of sketches and 3D models, respectively. The distance from the sketch  $S_i$  to the 3D model  $M_k$  can be denoted as  $D(x \rightarrow y)$ . Four kinds of distance metrics are proposed in this paper, as follows.

- (1)  $D(x \rightarrow y)$  is defined as the average distance from a sketch feature to all view features of a 3D model, calculated as follows:

$$D(x \rightarrow y) = \frac{\sum_j d(x, y^j)}{l}, 1 \leq j \leq l \quad (7)$$

- (2)  $D(x \rightarrow y)$  is defined as the minimum distance from a sketch feature to all view features of a 3D model, calculated as follows:

$$D(x \rightarrow y) = \min_j d(x, y^j), 1 \leq j \leq l \quad (8)$$

In Formulas (7) and (8),  $d(x, y^j)$  is the distance between the sketch feature vector  $x$  and the feature vector  $y^j$  of the  $j$ th view of the 3D model. Here, both the Euclidean distance and the Wasserstein distance (earthmover's distance) can be selected; they are calculated as follows:

Euclidean distance:

$$d(x, y^j) = \sqrt{\sum_{i=1}^d (x_i - y_i^j)^2} \quad (9)$$

Wasserstein distance:

$$d(x, y^j) = \sum_{i=1}^d \sum_{m=1}^d d_{im} f_{im} \quad (10)$$

where,  $d_{im}$  is the spatial distance from  $x_i$  to  $y_m^j$ , and  $f_{im}$  is the amount of soil moved from  $x_i$  to  $y_m^j$ .

Four kinds of distance metrics, EUD\_Ave, EUD\_Min, EMD\_Ave, and EMD\_Min are acquired by substituting the Euclidean distance and the Wasserstein distance into the expressions (7) and (8) respectively. Accordingly, the similarity between a query sketch and all 3D models can be calculated and compared so that the sketch-based 3D model retrieval can be realized.

## 4. Experiments

### 4.1. Datasets and Evaluation Metrics

We conduct experiments on two large-scale sketch-based 3D model retrieval datasets, namely SHREC 2013 and SHREC 2014.

**SHREC 2013 [3,5]:** This dataset is composed of 7200 sketches and 1258 3D models, divided into 90 classes. The number of 3D models in each class varies from 4 to more than 184, while the number of sketches for each class is equal to 80. In the experiment, 50 sketches per class are randomly selected for training and the remaining 30 sketches per class are used for testing; a total of 1258 3D models are targeted for retrieval.

**SHREC 2014 [4,6]:** Compared with the SHREC 2013 dataset, this dataset has more classes and models. Specifically, this dataset has 13,680 sketches and 8978 3D models, divided into 171 classes. The number of 3D models in the different classes varies greatly from less than 10 to more than 300. The number of sketches for each class is also equal to 80, and for each class 50 sketches are for training, and 30 sketches are for testing.

To comprehensively evaluate the performance of the proposed method, we employ seven commonly adopted performance metrics in the 3D model retrieval field [5,6]: The precision-recall diagram, the nearest neighbor (NN), the first tier (FT), the second tier (ST), the E measures (E), the discounted cumulative gain (DCG) and the average precision (AP).

### 4.2. Comparison of Different Distances

In this section, we will compare the effects of the proposed retrieval method using four different distances. Table 2 shows the comparison results on the SHREC 2013 dataset. In Table 2, for all evaluation criteria, the method using the EUD\_Ave distance outperforms the other methods. Through analysis, we found two reasons for this result: First, the average distance could fully consider the comprehensive information of all views' features; and second, there is no direct comparison

between the different feature components. Therefore, the Wasserstein distance is not as good as the Euclidean distance.

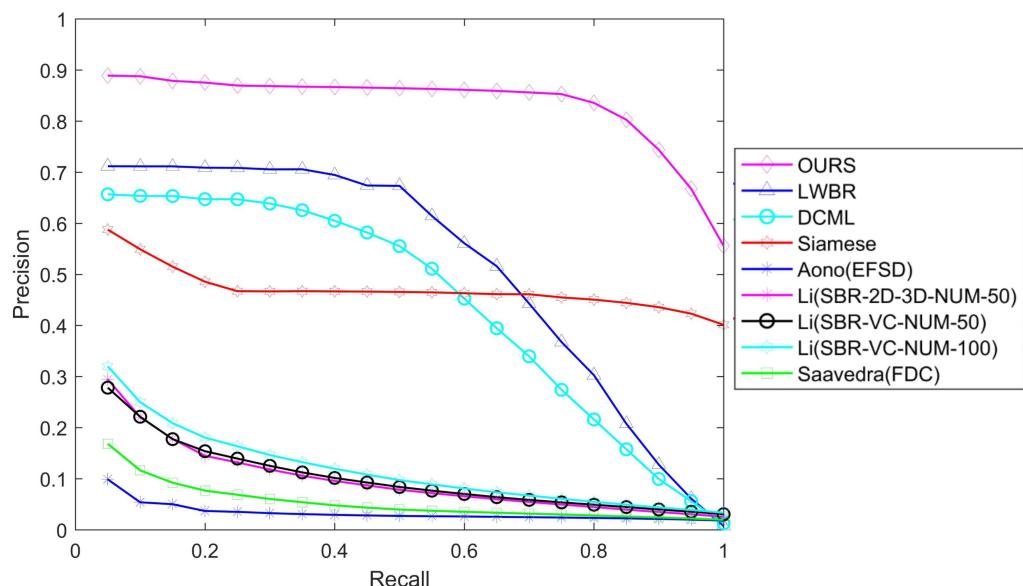
**Table 2.** Retrieval performance comparisons using different distances on the SHREC 2013 dataset. The nearest neighbor (NN), the first tier (FT), the second tier (ST), the E measures (E), the discounted cumulative gain (DCG) and the average precision (AP). The best performance indicators are marked as bold.

-	NN	FT	ST	E	DCG	AP
EUD_Ave	<b>0.849</b>	<b>0.772</b>	<b>0.858</b>	<b>0.410</b>	<b>0.888</b>	<b>0.817</b>
EUD_Min	0.816	0.744	0.847	0.406	0.869	0.790
EMD_Ave	0.844	0.764	0.852	0.407	0.886	0.812
EMD_Min	0.813	0.741	0.845	0.406	0.868	0.787

#### 4.3. Comparison with the State-Of-The Art Methods

**Retrieval from the SHREC 2013 dataset.** In this subsection, the proposed method is tested on the SHREC 2013 dataset and compared with the state-of-the-art methods, including the cross domain manifold ranking method (CDMR) [13], the sketch-based retrieval method with view clustering (SBR-VC) [5], the spatial proximity method (SP) [32], Fourier descriptors on 3D model silhouettes (FDC) [5], the Siamese network (Siamese) [20], the deep correlated metric learning (DCML) [17], the learned Wasserstein barycentric representation method (LWBR) [21], and semantic embedding space (SEM) [19] methods.

We use standard precision-recall curves to visualize our results. Figure 8 shows that our proposed method significantly outperforms the state-of-the-art methods. (1) Our method has the highest precision. For every recall value, our method gives better precision than the competing methods. On average, the precision values are 63%, 89%, and 77% higher than those of LWBR, DCML and Siamese, respectively. (2) The precision of our method is very stable: The precision-recall curve is closer to the horizontal line when the recall is less than 80%, and higher than 55% until the recall is 100%. This finding indicates that 80% of the related models within the target object's class are returned as the top hits.



**Figure 8.** Precision-recall comparisons on the SHREC 2013 dataset

Except for the precision–recall curve, other standard metrics, including NN, FT, ST, E, DCG and AP, are also calculated and compared with the state-of-art methods. Table 3 shows the results on the

SHREC 2013 dataset, which indicate that the proposed method is comparable to that of SEM [19] and has an evident advantage over other classical methods for every criterion.

**Table 3.** Performance comparisons of different evaluation criteria on the SHREC 2013 dataset. The best performance indicators are marked as bold.

Methods	NN	FT	ST	E	DCG	AP
CDMR [13]	0.279	0.203	0.296	0.166	0.458	0.250
SBR-VC [5]	0.164	0.097	0.149	0.085	0.348	0.116
SP [32]	0.017	0.016	0.031	0.018	0.240	0.026
FDC [5]	0.053	0.038	0.068	0.041	0.279	0.051
Siamese [20]	0.405	0.403	0.548	0.287	0.607	NA
DCML [17]	0.650	0.634	0.719	0.348	0.766	NA
LWBR [21]	0.712	0.725	0.785	0.369	0.814	NA
SEM [19]	0.823	<b>0.828</b>	<b>0.860</b>	0.403	0.884	NA
OURS	<b>0.849</b>	0.772	0.858	<b>0.410</b>	<b>0.888</b>	<b>0.817</b>

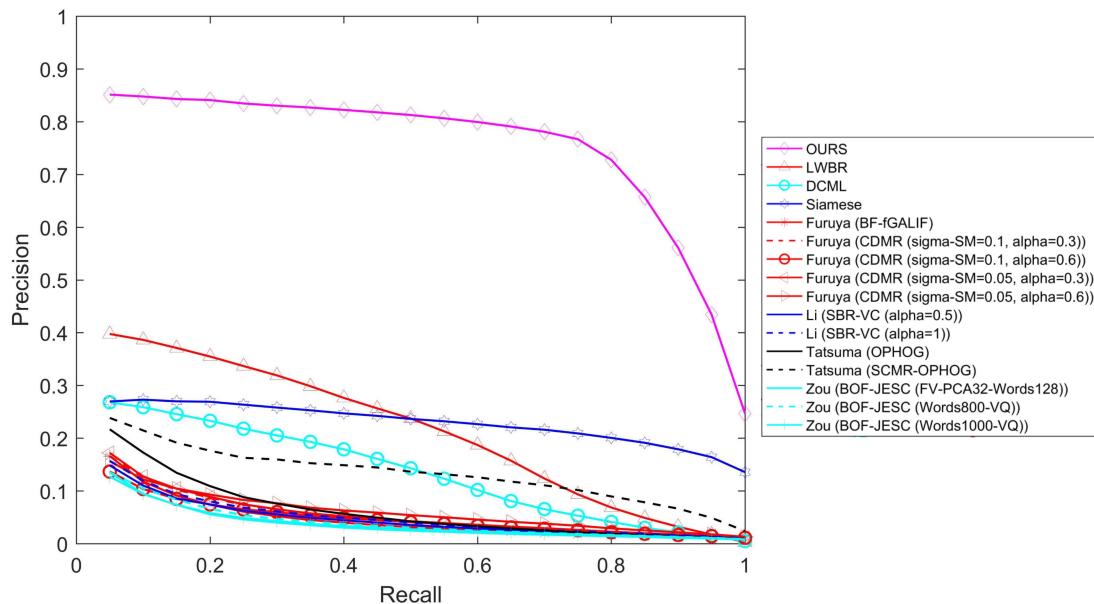
Figure 9 presents some examples of sketch-based 3D model retrieval results using the proposed method. The query's labels and sketches are listed on the left, and the top 10 retrieved models are listed on the right side in ascending order of distance. The correct results are framed in black, and the incorrect results are framed in red. For the classes of airplane, chair, hand, guitar and palm tree that all 10 retrieved models are correct, while for the classes of dog, dolphin and horse, the first few results retrieved are correct, and the last ones are incorrect. This finding is because the classes of dog, dolphin and horse contain only 7, 5 and 6 models, respectively.



**Figure 9.** Some retrieval examples on the SHREC 2013 dataset.

**Retrieval from the SHREC14 dataset.** In this subsection, the proposed method is tested on the SHREC 2014 dataset and compared with the state-of-the-art methods, including BF-fGALIF [8], CDMR [13], SBR-VC [5], Siamese network [20], DCML [17], LWBR [21], SEM [19], BOF-JESC [33], and MVPR [34].

Figure 10 shows the precision–recall curves. Obviously, the proposed method is still significantly better than other methods and very stable for every recall value. Table 4 shows performance comparisons of different evaluation criteria on the SHREC 2014 dataset and reveals that the proposed method also dominates for every criterion when using a large dataset.



**Figure 10.** Precision–recall comparisons on the SHREC 2014 dataset.

**Table 4.** Performance comparisons of different evaluation criteria on the SHREC 2014 dataset. The best performance indicators are marked as bold.

Methods	NN	FT	ST	E	DCG	AP
BF-fGALIF [8]	0.115	0.051	0.078	0.036	0.321	0.044
CDMR [13]	0.109	0.057	0.089	0.041	0.328	0.054
SBR-VC [5]	0.095	0.050	0.081	0.037	0.319	0.050
BOF-JESC (Words800-VQ) [33]	0.086	0.043	0.068	0.030	0.310	0.041
Siamese [20]	0.239	0.212	0.316	0.140	0.496	NA
DCML [17]	0.272	0.275	0.345	0.171	0.498	NA
LWBR [21]	0.403	0.378	0.455	0.236	0.581	NA
MVPR [34]	0.546	0.506	0.642	0.301	0.715	0.543
SEM [19]	0.804	<b>0.749</b>	<b>0.813</b>	<b>0.395</b>	0.870	NA
OURS	<b>0.830</b>	0.708	0.807	0.384	<b>0.871</b>	<b>0.745</b>

## 5. Conclusions

In this paper, we presented an effective approach for sketch-based 3D model retrieval using deep common semantic space embedding (DCSSE). To reduce the visual perception gap between sketches and 3D models, we transformed the data of two different domains into one common modality based on cross entropy evaluation. Furthermore, we learned the features of the sketches and 3D models simultaneously via deep common semantic embedding. We trained the deep embedding using a triplet network according to the characteristics of the cross-domain data. Finally, four distance metrics are used to calculate the sketch query and 3D models in the database. The experiments on the

large-scale datasets SHREC 2013 and SHREC 2014 demonstrate that the proposed approach is superior to state-of-the-art algorithms.

**Author Contributions:** J.B. proposed the framework and ideas. M.W. realized the ideas and wrote the paper. D.K. assisted in coding and writing paper.

**Funding:** This research was funded by the National Natural Science Foundation of China (61762003), Natural Science Foundation of Ningxia Province of China (2018AAC03124), First-class discipline construction in Ningxia universities (Electronic Science and Technology: NXYLXK2017A07), Key Research Project of North Minzu University “Research on 3D Model Retrieval for Generalized Design of Complex Products”, Postgraduate Innovation Project of North Minzu University (YCX18049).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Funkhouser, T.; Min, P.; Kazhdan, M.; Chen, J.; Halderman, A.; Dobkin, D.; Jacobs, D. A search engine for 3D models. *ACM Trans. Graph.* **2003**, *22*, 83–105. [[CrossRef](#)]
2. Tangelder, J.W.H.; Veltkamp, R.C. A survey of content based 3D shape retrieval methods. *Multimedia Tools Appl.* **2008**, *39*, 441. [[CrossRef](#)]
3. Li, B.; Lu, Y.; Godil, A.; Schreck, T.; Bustos, B.; Ferreira, A.; Furuya, T.; Fonseca, M.J.; Johan, H.; Matsuda, T.; et al. A comparison of methods for sketch-based 3D shape retrieval. *Comput. Vis. Image Underst.* **2014**, *119*, 57–80. [[CrossRef](#)]
4. Li, B.; Lu, Y.; Li, C.; Godil, A.; Schreck, T.; Aono, M.; Burtscher, M.; Chen, Q.; Chowdhury, N.K.; Fang, B.; et al. A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries. *Comput. Vis. Image Underst.* **2015**, *131*, 1–27. [[CrossRef](#)]
5. Li, B.; Lu, Y.; Godil, A. SHREC’13 Track: Large Scale Sketch-Based 3D Shape Retrieval. In Proceedings of the Eurographics Workshop on 3D Object Retrieval, Girona, Spain, 11–12 May 2013; Eurographics Association: Girona, Spain, 2013.
6. Li, B.; Lu, Y.; Li, C.; Godil, A.; Schreck, T.; Aono, M.; Burtscher, M.; Fu, H.; Furuya, T.; Johan, H.; et al. SHREC’14 track: Extended large scale sketch-based 3D shape retrieval. In Proceedings of the Eurographics Workshop on 3d Object Retrieval, Strasbourg, France, 6 April 2014; Eurographics Association: Strasbourg, France, 2014.
7. Yoon, S.M.; Scherer, M.; Schreck, T.; Kuijper, A. Sketch-based 3D model retrieval using diffusion tensor fields of suggestive contours. In Proceedings of the International Conference on Multimedia, Philadelphia, PA, USA, 29–31 March 2010; pp. 193–200.
8. Eitz, M.; Richter, R.; Boubekeur, T.; Hildebrand, K.; Alexa, M. Sketch-based shape retrieval. *ACM Trans. Graph.* **2012**, *31*, 1–10. [[CrossRef](#)]
9. Li, B.; Johan, H. Sketch-based 3D model retrieval by incorporating 2D-3D alignment. *Multimedia Tools Appl.* **2013**, *65*, 363–385. [[CrossRef](#)]
10. Li, B.; Lu, Y.; Johan, H. Sketch-Based 3D Model Retrieval by Viewpoint Entropy-Based Adaptive View Clustering. In Proceedings of the Eurographics Workshop on 3D Object Retrieval, Girona, Spain, 11–12 May 2013; Eurographics Association: Lower Saxony, Germany, 2013.
11. Huang, Z.; Fu, H.; Lau, R.W.H. Data-driven segmentation and labeling of freehand sketches. *ACM Trans. Graph.* **2014**, *33*, 175. [[CrossRef](#)]
12. Li, B.; Lu, Y.; Shen, J. A semantic tree-based approach for sketch-based 3D model retrieval. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016.
13. Furuya, T.; Ohbuchi, R. Ranking on cross-domain manifold for sketch-based 3D model retrieval. In Proceedings of the 2013 International Conference on Cyberworlds, Yokohama, Japan, 21–23 October 2013; pp. 274–281.
14. Tasse, F.P.; Dodgson, N.A. Shape2vec: Semantic based descriptors for 3D shapes, sketches and images. *ACM Trans. Graph.* **2016**, *35*, 208. [[CrossRef](#)]
15. Li, Y.; Lei, H.; Lin, S.; Luo, G. A new sketch-based 3D model retrieval method by using composite features. *Multimedia Tools Appl.* **2018**, *77*, 2921–2944. [[CrossRef](#)]

16. Zhu, F.; Xie, J.; Fang, Y. Learning Cross-Domain Neural Networks for Sketch-Based 3D Shape Retrieval. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 15–17 February 2016; pp. 3683–3689.
17. Dai, G.; Xie, J.; Zhu, F.; Fang, Y. Deep Correlated Metric Learning for Sketch-based 3D Shape Retrieval. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 4002–4008.
18. Dai, G.; Xie, J.; Fang, Y. Deep Correlated Holistic Metric Learning for Sketch-based 3D Shape Retrieval. *IEEE Trans. Image Process.* **2018**, *27*, 3374–3386. [[CrossRef](#)] [[PubMed](#)]
19. Qi, A.; Song, Y.Z.; Xiang, T. Semantic Embedding for Sketch-Based 3D Shape Retrieval. In Proceedings of the British Machine Vision Conference (BMVC), Newcastle, UK, 2–6 September 2018.
20. Wang, F.; Kang, L.; Li, Y. Sketch-based 3d shape retrieval using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1875–1883.
21. Xie, J.; Dai, G.; Zhu, F.; Fang, Y. Learning barycentric representations of 3D shapes for sketch-based 3D shape retrieval. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Venice, Italy, 21–26 July 2017; pp. 3615–3623.
22. Chen, J.; Fang, Y. Deep Cross-modality Adaptation via Semantics Preserving Adversarial Learning for Sketch-based 3D Shape Retrieval. *arXiv* **2018**, arXiv:1807.01806.
23. Xu, Y.; Hu, J.; Zeng, K.; Gong, Y. Sketch-Based Shape Retrieval via Multi-view Attention and Generalized Similarity. In Proceedings of the 2018 7th International Conference on Digital Home (ICDH), Guilin, China, 30 November–1 December 2018.
24. Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E. Multi-view convolutional neural networks for 3d shape recognition. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 945–953.
25. Guo, H.; Wang, J.; Gao, Y.; Li, J.; Lu, H. Multi-view 3D object retrieval with deep embedding network. *IEEE Trans. Image Process.* **2016**, *25*, 5526–5537. [[CrossRef](#)] [[PubMed](#)]
26. Phong, B.T. Illumination for computer generated pictures. *Commun. ACM* **1975**, *18*, 311–317. [[CrossRef](#)]
27. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–11 December 2014; MIT Press: Boston, MA, USA, 2014.
28. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv* **2017**, arXiv:1703.10593.
29. Vedral, V. The role of relative entropy in quantum information theory. *Rev. Mod. Phys.* **2002**, *74*, 197. [[CrossRef](#)]
30. Hoffer, E.; Ailon, N. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*; Springer: Cham, Switzerland, 2015; pp. 84–92.
31. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems 25 (NIPS 2012), Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
32. Sousa, P.; Fonseca, M.J. Sketch-based retrieval of drawings using spatial proximity. *J. Vis. Lang. Comput.* **2010**, *21*, 69–80. [[CrossRef](#)]
33. Zou, C.; Wang, C.; Wen, Y.; Zhang, L.; Liu, J. Viewpoint-Aware Representation for Sketch-Based 3D Model Retrieval. *IEEE Signal Process. Lett.* **2014**, *21*, 966–970.
34. Li, H.; Wu, H.; He, X.; Lin, S.; Wang, R.; Luo, X. Multi-view pairwise relationship learning for sketch based 3D shape retrieval. In Proceedings of the 2017 IEEE International Conference on Multimedia and Expo (ICME), Hong Kong, China, 10–14 July 2017; pp. 1434–1439.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).