Semantic Embedding for Sketch-Based 3D Shape Retrieval

Anran Qi
a.qi@qmul.ac.uk
Yi-Zhe Song
yizhe.song@qmul.ac.uk
Tao Xiang
t.xiang@qmul.ac.uk

SketchX Research Lab
Queen Mary University of London
London, UK

Abstract

The main challenge for sketch-based 3D shape retrieval lies with the large domain gap between 2D sketch and 3D shape. Most existing works attempt to overcome the domain gap by learning a joint feature embedding space to align the two domains. In this work we argue that the large domain gap cannot be effectively bridged in a shared feature space. Instead, we propose to align them in their common class label space. To this end, a novel deep cross-domain semantic embedding model is proposed. Extensive experiments are carried out on two large benchmarking datasets, SHREC'13 and SHREC'14. The results show that the proposed model drastically improves over the state-of-the-art alternatives.

1 Introduction

Sketch-based 3D shape retrieval has been studied extensively in both computer vision and computer graphics [4, 9, 7, 8, 13, 14, 18, 34, 38]. Given a sketch image as query, it aims to retrieve from a gallery set of 3D shapes the ones that belong to the same category as the query sketch. Compared to using 3D shapes as queries, sketches are not only more intuitive to humans, but also more convenient and easier to obtain, thanks to the universal availability of touch screen devices such as smartphones. As a result, sketch-based 3D shape retrieval has received increasing attention both from the research community and industry.

The main challenge for sketch-based 3D shape retrieval lies with the large domain gap, which can be broadly factorised into (i) the dimensionality gap: sketches are represented in 2D, whereas 3D shapes are embodied in a higher dimensional (3D) space, (ii) the view gap: sketches are selected 2D depictions from specific view points, yet 3D shape models are entirely view-independent, and (iii) the abstraction gap: sketches are highly abstract and iconic, however 3D shapes are geometrically realistic. For a sketch based 3D shape retrieval model to work, these gaps must been narrowed/removed so that corresponding sketches and 3D shapes can be matched. Figure 1 offers a visualisation.

All existing works tackle this domain gap problem by learning a joint feature embedding space. Both 2D sketches and 3D shapes are projected into the space where the similarity between a pair of sketch and 3D shape is measured using a feature distance. These methods

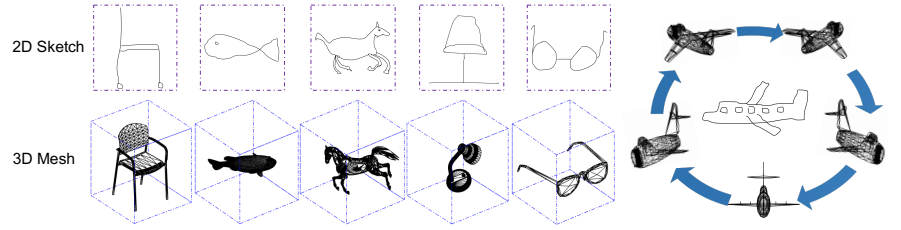

Figure 1: Left: 2D sketches and corresponding 3D shapes. Sketches are 2D images, whereas 3D shapes are represented as 3D meshes. Sketches are highly abstract, while 3D shapes are well defined. Right: a sketch has an associated view point whilst a 3D shape does not.

[7, 8, 34, 38] mostly start by directly projecting 3D models into 2D contour(s), in an effort to alleviate the dimensionality gap. After projection the matching is done between two 2D images, yet large domain discrepancy still remains owning to the abstraction gap. In order to narrow the view gap, it follows that the best view(s) to project all the 3D shapes in the gallery set will also have to be determined. This is non-trivial and importantly, useful information about the 3D shapes can be lost in the projection process, and picking inappropriate views will result in considerable performance degradation. To overcome this, recently a projection-free approach is proposed [5] which directly extracts 3D features from 3D shape models. They attempt to alleviate the domain gap all together by learning a non-linear transformation to directly map 2D and 3D features into a joint feature space to conduct matching. Despite achieving decent retrieval performance, it is inferior to recent projection-based approaches [38] that specifically tackled the aforementioned gaps. Regardless whether a view projection step is required, the existing joint feature embedding learning based approaches are ineffective in bridging the large domain gap.

In this paper, a completely different approach is proposed. Specifically, we argue that no matter how hard we try, it is impossible to learn a joint embedding space where a sketch and its corresponding 3D shape can be perfectly aligned due to the various gaps mentioned above. However, since both sketches and 3D shapes belong to the same set of object classes, their class label space are shared. When such a space is used as joint embedding space, perfect alignment is intrinsically achievable. Such a space is semantic, a vital difference from the feature embedding space. To learn such a joint semantic embedding space, we formulate a deep neural network consisting heterogeneous branches for the sketch and 3D shape domains respectively. More specifically, Inception-ResNet-v2 [32] is adopted to map input sketches to the embedding space, while PointNet [23] is employed for the 3D shape input. To make sure that the learned embedding space is both semantic and discriminative, classification and triplet ranking losses are imposed as learning objectives.

The contributions of our work are listed as follows: 1) For the first time, we propose to perform sketch based 3D shape retrieval in a joint semantic embedding space, instead of the joint feature embedding space adopted by existing methods. 2) A novel heterogeneous deep network with classification and triplet ranking losses is formulated to learn the joint semantic embedding space for effective and efficient cross-domain matching. 3) Extensive experiments are carried out on the two largest benchmark datasets. We show that our model outperforms existing models by large margins. In addition, we propose a more rigorous experiment setting, under which the advantage of the proposed method is shown to be even more pronounced.

2 Related Work

Since the proposed model aims to project sketch and 3D shape into a shared label space using a multi-branch deep neural network, each branch is essentially solving a recognition problem in its corresponding domain. We thus first review some existing sketch and 3D shape recognition works.

Sketch Recognition Early studies on sketch recognition work with professional CAD or artistic drawings as input [19, 41]. Free-hand sketch recognition has attracted much attention since Eitz *et al*. [6] released the first large-scale TU-Berlin sketch dataset. It has 20,000 free-hand sketches across 250 categories of daily objects. A number of approaches have since been proposed to recognise freehand sketches. Early works [6, 16, 27] use SVM as the classifier with hand-crafted features such as HOG and SIFT as representation. More recently, Convolutional Neural Networks (CNNs) have dominated the top benchmark results on various visual recognition challenges such as ImageNet ILSVRC [25]. Sketch recognition is no exception. In [40], Yu *et al*. propose Sketch-a-Net, a CNN specifically designed for modelling sketches. In [26], Sarvadevabhatla *et al*. use two popular CNN architectures pre-trained on ImageNet and fine-tune on the TU-Berlin sketch dataset. In our work, we use the Inception-ResNet-v2 [32] as the 2D sketch branch in our heterogeneous cross-domain matching network to project a 2D sketch to its class label space.

3D Shape Recognition Most recently proposed 3D shape representation models are also based on deep neural networks. Various volumetric CNNs [22, 36] have been proposed to model voxel shapes. Each 3D mesh is represented as a binary tensor: 1 indicates that a voxel is inside the mesh surface, and 0 otherwise. Volumetric representation is constrained by data sparseness and high computational cost incurred by 3D convolution. In contrast, in a multi-view CNNs based approach [31], 2D images are rendered from 3D point cloud or meshes before 2D CNNs are employed to classify them. The current state-of-the-art is PointNet [23]. Using this model, each point in a 3D point cloud is represented by its three coordinates (x, y, z) and some point features that encode its statistical properties. A deep network composing of multi-layer perception is then designed to provide a unified architecture for 3D object classification, part segmentation, and scene semantic tasks. PointNet is used in our 3D shape branch due to its strong classification performance and low computational cost.

Sketch-based 3D Shape Retrieval Most earlier sketch-based 3D shape retrieval methods focus on finding the "best views" for projecting 3D shapes for matching them with sketches in 2D. [4] proposes a set of uniformly distributed viewpoints for 3D shape projection. Differently, [7] learns a view classifier which is used to select the best view for projection given a query sketch. Both methods use hand-crafted features, whilst later approaches are all deep learning based. The method in [34] adopts a Siamese network to learn a joint embedding space for the two modalities. Similarly, [38] uses two deep convolutional neural networks to extract deep features of sketches and 2D projections of 3D shapes with uniformly sampled viewpoints first. Then the Wasserstein barycentres of deep features of multiple projections of 3D shape are calculated to form a barycentric representation. In contrast to these projection-based methods, in [3], features for both sketches and 3D shapes are extracted first, then deep metric learning is performed to transform the raw features of both domains into a non-linear joint feature embedding space. All the existing methods aim to learn a joint feature embedding space; they are thus very different from our joint semantic embedding based approach. We show that our model is much more effective in bridging the domain gap because the semantic labels are intrinsically shared, making alignment in the space easier.

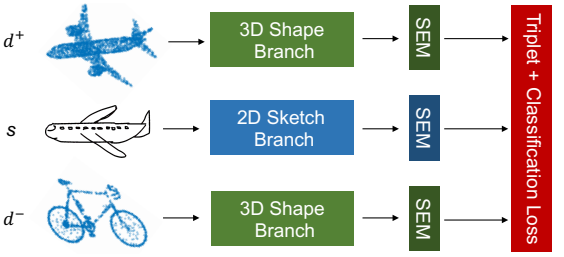

Figure 2: A schematic of the proposed network architecture (SEM: Semantic Embedding Space).

Cross-domain Matching Beyond sketch-based 3D shape retrieval, many other vision problems also require solving cross-domain matching problems. For visual versus near infrared (VIS-NIR) face recognition [9, 12, 17], faces are matched across two image modalities. Most models adopt a Siamese network architecture to align NIR and VIS images in a joint deep feature embedding space. Such a feature embedding space is also learned in sketch-based image retrieval (SBIR) [5, 11, 24, 29, 39] and person re-identification (re-id) [2, 10, 15, 30, 37]. In summary, existing approaches to various cross-domain matching problems are also dominated by joint feature embedding learning based approaches. Such an approach seems to be more effective for these tasks because the domain gap is narrower: all domains feature 2D images. It is worth noting that the re-id work in [35] also explores a label space for cross-domain alignment. However, instead of using a shallow model as in [35], our model learns a joint semantic embedding space by end-to-end training a deep neural network.

3 Methodology

3.1 Network Architecture

Overview The overall network architecture of the proposed sketch-based 3D model retrieval method is illustrated in Figure 2. It consists of two subnets: (1) a sketch subnet that aims to map an input sketch into a shared label embedding space whose dimension is the same as the number of object classes modelled, and (2) a Siamese 3D shape subnet, where each branch has the same base network architecture and they share parameters; this subnet aims to map input 3D shapes into the joint semantic embedding space. In the space, the learned two projections are subject to a triplet ranking loss, a sketch classification loss and a 3D shape classification loss to ensure that the space is both semantic (*i.e.*, each object class is represented as a one-shot vector and the projection of sketches/3D shapes of that class should be close to the vector) and cross-domain discriminative (*i.e.*, a pair of sketch and 3D sketch that belong to the same class should be close whilst those of different classes should be farther apart).

2D Sketch Branch The sketch classification architecture is based on that of Inception-ResNet-v2 [32], which combines the Inception architecture with residual connections. To regularise the network, we use label smoothing both in training and testing process [21].

3D Shape Branch The 3D classification architecture is based on PointNet [23], which directly takes the 3D coordinates of points in a 3D shape point cloud as input. Specifically, n points are used as input to feed into the network; they go through feature transformations layers and five fully connected layers before being aggregated into a shape feature vector by

max pooling. Label smoothing is also used to regularise the network.

3.2 Model Learning

Optimisation Objective Suppose the sketch subnet and the 3D shape subnet learn mapping functions Φ_S and Φ_D, parameterised by Θ_S and Θ_D respectively. The model takes a triplet as input consisting of a query sketch, a positive 3D shape of the same class and a negative 3D shape from a different class. Given N triplets $\chi = \{x_i^s, x_i^{d^+}, x_i^{d^-}\}_{i=1}^N$, where x_i^s denotes the anchor query sketch, $x_i^{d^+}$ a positive 3D shape and $x_i^{d^-}$ a negative 3D shape. Our learning objective is:

$$\arg\min_{\Theta_S,\Theta_D} L = \arg\min_{\Theta_S,\Theta_D}(L_S + \lambda_D L_D + \lambda_T L_T) \tag{1}$$

where L_S, L_D are the cross-entropy loss for sketch and 3D classification weighted by λ_D, that is,

$$L_S = -\sum_{i=1}^{N}(p_i{}^s \log(\hat{p}_i^s)) \tag{2}$$

$$L_D = -\sum_{i=1}^{N}\left(p_i{}^{d^+} \log(\hat{p}_i^{d^+}) + p_i{}^{d^-} \log(\hat{p}_i^{d^-})\right), \tag{3}$$

and L_T is the triplet ranking loss with a soft-margin weighted by λ_T [10] defined as:

$$L_T = \sum_{i=1}^{N} \ln\left(1 + \exp(\|\Phi_S(x_i^s) - \Phi_D(x_i{}^{d^+})\|_2 - \|\Phi_S(x_i^s) - \Phi_D(x_i{}^{d^-})\|_2)\right) \tag{4}$$

This ranking loss considers the differences in the joint semantic embedding space measured using Euclidean distance. The goal is to learn a discriminative semantic embedding space where the positive 3D shape $x_i^{d^+}$ is ranked above the negative 3D $x_i^{d^-}$ in terms of its distance to the query sketch s, and to help the two classification losses to align the two domains in the space.

Hard Training Sample Mining The hard training sample mining strategy in [10] is adapted here for our problem. Concretely, we form batches by randomly sampling N sketches from N different classes (*i.e.*, one sketch per class) and then randomly sampling K 3D shapes for each class, resulting in a mini-batch size of $(NK+N)$. For each sample x^s in the sketch classes, we select the hardest positive and the hardest negative 3D samples within the mini-batch when forming the triplets for computing the loss. In order to capture the semantic relationship, we select hard negative samples according to the word vector distance [20] among different classes. Specifically, for each sketch x^s, we first choose the hardest negative 3D class based on the word vector distance so that the two classes are semantically as similar as possible. Then from that class the hardest negative 3D sample is chosen from the K 3D shape instances.

4 Experiments

4.1 Datasets and Settings

Three datasets are used for evaluation, which are summarised in Table 1.

SHREC'13 SHREC'13 is a large-scale benchmark for sketch-based 3D shape retrieval.

Dataset	Number of Classes	Number of Sketches	Number of 3D Shapes	Train/Test Split for 3D Shapes
SHREC'13	90	7,200	1,258	No
SHREC'14	171	13,680	8,987	No
PART-SHREC'14	48	4,320	7,238	Yes

Table 1: A summary of the three benchmark datasets.

The dataset is created by collecting common classes from both the Princeton Shape Benchmark [28] and the TU-Berlin sketch dataset [6]. There are 1,258 3D shapes and 7,200 sketches from 90 classes. The 80 sketches in each class is split in two sets: 50 for training and 30 for testing, whilst all the 3D shapes in the dataset are used in both training and testing. It is noteworthy to mention that the number of 3D shapes varies among different classes. For example, the largest class has 184 instances but there are 23 classes containing no more than 5 shapes.

SHREC'14 SHREC'14 [14] is a more challenging dataset compared to SHREC'13. In particular, the number of 3D shapes is increased to 8,987, and the number of classes to 171. The number of sketches per class remains unchanged with the same training/test split.

PART-SHREC'14 SHREC'13 and SHREC'14 are the largest and most widely used dataset for sketch-based 3D shape retrieval. However, it is noted that all existing works use all the 3D shapes for both training and testing. This setting is clearly not rigorous and unable to evaluate how well a model can generalise to unseen 3D shapes. In contrast, for other cross-domain matching problems such as VIS-NIR face recognition and person re-id, all benchmarks are organised different to make sure that the testing data and training data have no overlap. Following this practice, we propose a new benchmark, PART-SHREC'14, which is a subset of SHREC'14. Specifically, it consists of the classes in SHREC'14 that have more than 50 instances. With this selection criterion, there are 48 classes, 7,238 3D shapes and 3,840 sketches in total. We follow the same 50/30 training/test split for sketches as in SHREC'13 and SHREC'14 [3, 13, 34, 38]. Meanwhile, the 3D shapes are randomly split into a training set of 5,812 samples and a test set of 1,426.

4.2 Implementation Details

For data augmentation, akin to [40] we randomly perform affine transformations on each sketch to generate more variations. Fifteen augmentations are generated for each sketch in the dataset. For 3D shape, we uniformly sample 2,048 points on the mesh faces according to face area and normalise them into a unit sphere. We also argument the 3D data using rotation and jitter following [23]. Once trained, *Cosine* distance is used to compare a query sketch and a gallery 3D shape in the joint embedding space.

Our model is implemented on Tensorflow with a single NVIDIA GeForce GTX 1080 Ti GPU with Adam optimiser. We first pre-train the sketch subnet on ImageNet and 3D shape subnet for the training 3D shape classification task. The whole network is then fine-tuned for 30 epochs. We set the important weights for different subsets to: $\lambda_D = 1$ and $\lambda_T = 20$ (see Eq. (1)). The learning rate is set as 10^{-4} with decay rate = 0.9 and decay step = 20000. Each mini-batch is composed of 8 sketches that belong to different categories and 4 corresponding 3D shapes for each sketch.

4.3 Evaluation Metrics

The following widely-used evaluation metrics are adopted [8, 28]: 1) Nearest neighbour (NN) matching accuracy, which is the percentage of the closest matches that belong to the same class as the query sketch, 2) first/second tier (FT/ST), which is the percentage of 3D shapes in the query's class that appear within the top K matches, where K depends on the size of the query's class, 3) E-Measure (E), which is a composite measure of the precision and recall for a fixed number of retrieved results, 4) Discounted cumulated gain (DCG), which is a statistic that weighs correct results near the front of the list more than correct results towards the bottom of the ranked list under the assumption that a user is less likely to consider any results near the bottom of the list and 5) mean Average Precision (mAP).

4.4 Competitors

We compare the proposed method (Ours) to four state-of-the-art alternatives including Shape-2vec [33], Siamese [34], LWBR [38], DCML [3], and SBR-VC [13]. Shape2vec [33] trains a CNN to predict labels first, then updates fully connected layers to generate shape descriptors that lie close to the word vector representation of the shape class. Siamese [34] learns a Siamese Convolutional Neural Network, one for the 3D shapes projected into different viewpoints and one for the sketches. The loss function is defined on both within-domain and the cross-domain similarities in the feature space. LWBR [38] uses two deep convolutional neural networks to extract deep features of sketches and 2D projections of 3D shapes first. Then the Wasserstein barycentres of deep features of multiple projections of 3D shapes are calculated to form a barycentric representation. Finally, a discriminative loss is formulated on the deep feature space for two domains. [3] jointly trains one network for sketch and another for 3D shape using features from the corresponding domain with a loss designed to learn two deep non-linear transformations to map features from both domains into a non-linear feature space. SBR-VC [13] proposes a 3D shape visual complexity metric to decide the number of representative views of the 3D shapes. Then, a Fuzzy C-Means view clustering is performed on each selected views. Finally, shape context matching [1] is used to perform online retrieval.

4.5 Results on SHREC'13 and SHREC'14

Table 2 compares our model with the four state-of-the-art alternatives on SHREC'13 and SHREC'14. We have the following observations: (1) On both datasets and across all metrics, our model achieves the best performance. (2) On the more challenging SHREC'2014 dataset, all four compared models' performance degrades drastically. In contrast, the performance drop of our model is much smaller. As a result, the gaps between our model and the alternative are much more significant. For example, compared to the nearest competitor LWBR [38], our model almost doubles the performance measured on all metrics. This suggests that by learning a semantic embedding space instead of a feature embedding space, our model is much more effective in tackling the domain gap issue. (3) Among the four compared models, as expected, the hand-crafted feature based SBR-VC model is the weakest. (4) For the three compared deep learning based models, LWBR is clearly better. This is because it uses the Wasserstein barycentres to synchronously aggregate the information of different project views for 3D shapes, which has the advantage over the Siamese model which treats each view independently.

	SHREC'13						SHREC'14					
	NN	FT	ST	E	DCG	mAP	NN	FT	ST	E	DCG	mAP
Shape2vec [33]	0.620	0.628	0.684	0.354	0.741	0.650	0.714	0.697	0.748	0.360	0.811	0.720
Siamese [34]	0.405	0.403	0.548	0.287	0.607	0.469	0.239	0.212	0.316	0.140	0.496	0.228
LWBR [33]	0.712	0.725	0.785	0.369	0.814	0.752	0.403	0.378	0.455	0.236	0.581	0.401
DCML [8]	0.650	0.634	0.719	0.348	0.766	0.674	0.272	0.275	0.345	0.171	0.498	0.286
SBR-VC [14]	0.164	0.097	0.149	0.085	0.348	0.116	0.095	0.050	0.081	0.037	0.319	0.050
Ours	**0.823**	**0.828**	**0.860**	**0.403**	**0.884**	**0.843**	**0.804**	**0.749**	**0.813**	**0.395**	**0.870**	**0.780**

Table 2: Retrieval results on the SHREC'13 and SHREC'14 datasets.

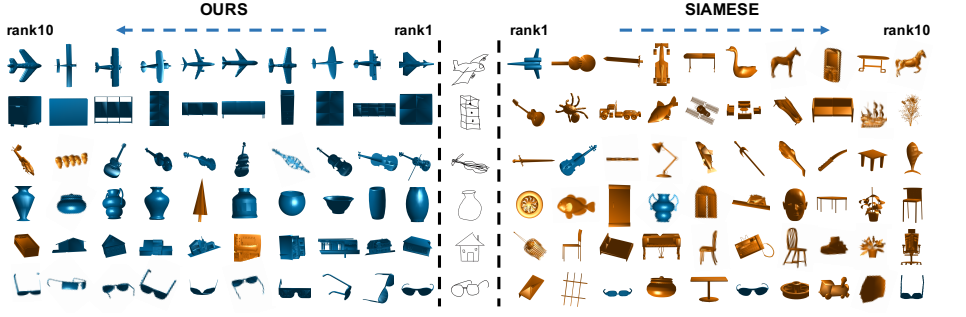

Figure 3: Qualitative results on PART-SHREC'14. The query sketches are listed in the middle column. The top 10 retrieved shapes of our model are listed on the left of the query and those of Siamese [34] on the right, based on their ranking orders. The correct retrieval results are rendered in navy blue, whilst the wrong results are shown in golden.

4.6 Results on PART-SHREC'14

Since only the source code of Siamese [34] is publicly available, we can only compare with it using this new benchmark. We conduct experiments under two settings: under the Old Setting (OS), all 3D shapes are used for both training and testing, while under the New Setting (NS) they are split into non-overlapping training and test sets. The comparative results are shown in Table 3. It can be seen that the performance of Siamese [34] drops dramatically when the setting changes from OS to NS – more than halved on metrics such as NN and mAP. This suggests that the model tends to overfit to the training data. In contrast, using our model, the performance degrades much more gracefully. We thus conclude that our model's advantage over the existing feature embedding based models are even more pronounced under this more rigorous setting. This is not surprising: our joint semantic space not only makes it easier to align the two domains, it typically has much lower dimensions compared to a feature space as well, so it is less likely to suffer from model overfitting. Figure 3 shows some qualitative retrieval results on PART-SHREC'14 under NS.

	NN	FT	ST	E	DCG	mAP
Siamese [34] OS/ NS	0.267/0.118	0.183/0.076	0.278/0.132	0.104/0.073	0.603/0.400	0.152/0.067
Ours OS/ NS	**0.846/0.840**	**0.832/0.634**	**0.892/0.745**	**0.372/0.526**	**0.931/0.848**	**0.854/0.676**

Table 3: Retrieval results on the PART-SHREC'14 benchmark dataset

		NN	FT	ST	E	DCG	mAP
SHREC'13	Ours-semantic-space	**0.823**	**0.828**	**0.860**	**0.403**	**0.884**	**0.843**
	Ours-feature-space	0.0111	0.0116	0.0231	0.0124	0.2316	0.0226
SHREC'14	Ours-semantic-space	**0.804**	**0.749**	**0.813**	**0.395**	**0.870**	**0.780**
	Ours-feature-space	0.0074	0.0057	0.0112	0.0034	0.2525	0.0094
PART-SHREC'14	Ours-semantic-space	**0.840**	**0.634**	**0.745**	**0.526**	**0.848**	**0.676**
	Ours-feature-space	0.0229	0.0209	0.0422	0.0202	0.3296	0.0275

Table 4: Comparison on joint feature embedding space and semantic embedding space

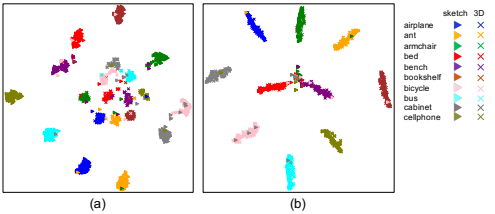

Figure 4: A visualisation of the mapped sketches and 3D shapes using our model in the joint (a) feature space and (b) semantic label space.

4.7 Ablation Study

We have shown convincingly that the proposed joint semantic embedding based model is far superior to the existing feature embedding based models. However, it is also noted that different base networks are used in our model, compared to those in existing models [3, 34, 38]. To evaluate how much exactly the change in the learned joint embedding space contributes to our model's performance, in this experiment, we compare our full model (Ours-semantic-space) with a variant termed Ours-feature-space. This variant has exactly the same network architecture and the only difference is that the triplet loss is added to a 256D feature embedding layer shared by the sketch and 3D shape branches. As shown in Table 4, the performance of this variant is extremely low on all three datasets. These results indicate that learning the joint semantic space and performing cross-domain matching in that space is indeed the main reason why our model works so well. Figure 4 visualises 10 classes of sketches and 3D shapes in the two joint embedding space. From Figure 4(a), it is clear that each class of sketches and 3D shapes form two separate clusters with a fair amount of distance between them, causing the miserable matching accuracy. In contrast, in the joint semantic space (Figure 4(b)), the two domains are perfectly aligned with sketches and 3D shapes of the same class forming a single cluster. It is noted that our joint feature space is far worse than those in the compared existing models [3, 34, 38]. This is expected: our network is heterogeneous with very different sketch and 3D shape subnets (Inception-ResNet-v2 vs. PointNet). Although each of them can produce state-of-the-art classification results on each modality, thus being well suited as a mapping function to the shared label space, the feature output of the two subnets are highly heterogeneous as well. As a result, the two domains are much harder to align in our feature space than those of the Siamese or similar 2D image subnets employed by existing models [3, 34, 38].

5 Conclusion

We have proposed a novel sketch-based 3D shape retrieval model. The key idea is that, since the sketch domain and 3D shape domain have a large domain gap, rather than focusing on learning a joint feature embedding space to bridge the gap, we argue that learning a joint semantic space is much easier. This is because the two domains share the same object classes thus the same label space which can be learned and re-purposed as a joint semantic embedding space for sketch to 3D shape matching. Extensive experiments show that our model drastically improves the state-of-the-art on a number of large-scale benchmarks.

References

[1] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *TPAMI*, 2002.

[2] Xiaobin Chang, Timothy M Hospedales, and Tao Xiang. Multi-level factorisation net for person re-identification. *arXiv preprint arXiv:1803.09132*, 2018.

[3] Guoxian Dai, Jin Xie, Fan Zhu, and Yi Fang. Deep correlated metric learning for sketch-based 3D shape retrieval. In *AAAI*, 2017.

[4] Petros Daras and Apostolos Axenopoulos. A 3D shape retrieval framework supporting multimodal queries. *IJCV*, 2010.

[5] Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *TVCG*, 2011.

[6] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *SIGGRAPH*, 2012.

[7] Mathias Eitz, Ronald Richter, Tamy Boubekeur, Kristian Hildebrand, and Marc Alexa. Sketch-based shape retrieval. *SIGGRAPH*, 2012.

[8] Afzal A Godil, Bo Li, Yijuan Lu, and Tobias Schreck. Shrec'13 track: Large scale sketch-based 3D shape retrieval. *Eurographics Workshop on 3D Object Retrieval*, 2013.

[9] Debaditya Goswami, Chi Ho Chan, David Windridge, and Josef Kittler. Evaluation of face recognition system in heterogeneous environments (visible vs NIR). In *ICCVW*, 2011.

[10] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[11] Rui Hu and John Collomosse. A performance evaluation of gradient field hog descriptor for sketch based image retrieval. *CVIU*, 2013.

[12] Brendan Klare and Anil K Jain. Heterogeneous face recognition: Matching nir to visible light images. In *ICPR*, 2010.

[13] Bo Li, Yijuan Lu, Afzal Godil, Tobias Schreck, Benjamin Bustos, Alfredo Ferreira, Takahiko Furuya, Manuel J Fonseca, Henry Johan, Takahiro Matsuda, et al. A comparison of methods for sketch-based 3D shape retrieval. *CVIU*, 2014.

[14] Bo Li, Yijuan Lu, Chunyuan Li, Afzal Godil, Tobias Schreck, Masaki Aono, Martin Burtscher, Hongbo Fu, Takahiko Furuya, Henry Johan, et al. Shrec'14 track: Extended large scale sketch-based 3D shape retrieval. *Eurographics Workshop on 3D Object Retrieval*, 2014.

[15] Wei Li, Xiatian Zhu, and Shaogang Gong. Person re-identification by deep joint learning of multi-loss classification. *arXiv preprint arXiv:1705.04724*, 2017.

[16] Yi Li, Timothy M Hospedales, Yi-Zhe Song, and Shaogang Gong. Free-hand sketch recognition by multi-kernel feature learning. *CVIU*, 2015.

[17] Shengcai Liao, Dong Yi, Zhen Lei, Rui Qin, and Stan Z Li. Heterogeneous face recognition from local structures of normalized appearance. In *ICB*, 2009.

[18] Jobst Loffler. Content-based retrieval of 3D models in distributed web databases by visual shape information. In *Information Visualization*, 2000.

[19] Tong Lu, Chiew-Lan Tai, Feng Su, and Shijie Cai. A new recognition model for electronic architectural drawings. *Computer-Aided Design*, 2005.

[20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *NIPS*, 2013.

[21] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

[22] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3D data. In *CVPR*, 2016.

[23] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017.

[24] Yonggang Qi, Yi-Zhe Song, Honggang Zhang, and Jun Liu. Sketch-based image retrieval via siamese convolutional neural network. In *ICIP*, 2016.

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

[26] Ravi Kiran Sarvadevabhatla and R Venkatesh Babu. Freehand sketch recognition using deep features. *arXiv preprint arXiv:1502.00254*, 2015.

[27] Rosália G Schneider and Tinne Tuytelaars. Sketch classification and classification-driven analysis using fisher vectors. *TOG*, 2014.

[28] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser. The princeton shape benchmark. In *Shape Modeling Applications*, 2004.

[29] Jifei Song, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Xiang Ruan. Deep multi-task attribute-driven ranking for fine-grained sketch-based image retrieval. In *BMVC*, 2016.

[30] Chi Su, Fan Yang, Shiliang Zhang, Qi Tian, Larry S Davis, and Wen Gao. learning with low rank attribute embedding for person re-identification. In *ICCV*, 2015.

[31] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *ICCV*, 2015.

[32] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017.

[33] Flora Ponjou Tasse and Neil Dodgson. Shape2vec: semantic-based descriptors for 3D shapes, sketches and images. *TOG*, 2016.

[34] Fang Wang, Le Kang, and Yi Li. Sketch-based 3D shape retrieval using convolutional neural networks. In *CVPR*, 2015.

[35] Hanxiao Wang, Shaogang Gong, and Tao Xiang. Highly efficient regression for scalable person re-identification. In *BMVC*, 2016.

[36] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D shapenets: A deep representation for volumetric shapes. In *CVPR*, 2015.

[37] Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *CVPR*, 2016.

[38] Jin Xie, Guoxian Dai, Fan Zhu, and Yi Fang. Learning barycentric representations of 3D shapes for sketch-based 3D shape retrieval. In *CVPR*, 2017.

[39] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen Change Loy. Sketch me that shoe. In *CVPR*, 2016.

[40] Qian Yu, Yongxin Yang, Feng Liu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Sketch-a-net: A deep neural network that beats humans. *IJCV*, 2017.

[41] C Lawrence Zitnick and Devi Parikh. Bringing semantics into focus using visual abstraction. In *CVPR*, 2013.