Sketch-Based 3D shape retrieval with multi-view fusion transformer

Zhu, C.; Cui, D.; Jia, Q.; Wang, W.; Liu, Y.; Lew, M.S.K.

ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10446103

SKETCH-BASED 3D SHAPE RETRIEVAL WITH MULTI-VIEW FUSION TRANSFORMER

Cunjuan Zhu [1], *Dongdong Cui* [1], *Qi Jia* [1*], *Weimin Wang* [1], *Yu Liu* [1*], *Michael S. Lew* [2]

[1] International School of Information Science and Engineering, Dalian University of Technology, China
[2] LIACS Media Lab, Leiden University, The Netherlands

ABSTRACT

Sketch-based 3D shape retrieval aims to retrieve similar 3D shapes given a 2D sketch query. Although this task has been studied for years, the inherent cross-modal gap and data imbalance between 2D sketches and 3D shapes remain challenging. To address the problems, we propose a simple and effective framework based on Multi-view Fusion Transformer. To be specific, we project 3D shapes into twelve distinct views, and their CNN features are combined with position embeddings, passing together into a transformer encoder to learn view weights. Then we process them through average pooling and MLPs to obtain the final 3D shape representation. Furthermore, to narrow the data imbalance between 2D sketches and 3D shapes, affine transformation and elastic deformation are fully utilized for sketch augmentation, so as to extract more comprehensive sketch features for feature matching with the multi-view 3D shape representation. Extensive experiments on SHREC13, SHREC14 and PART-SHREC14 datasets demonstrate our method achieves superior performance than previous competitive methods.

Index Terms— 2D sketch, 3D shape retrieval, multi-view fusion, Transformer

1. INTRODUCTION

Different from text-based 3D shape retrieval [1] and instance-based 3D shape retrieval [2], sketch-based 3D shape retrieval aims to retrieve the corresponding 3D shape according to a 2D sketch query [3, 1, 4]. Although sketches provide intuitive shape features similar to 3D shapes, there is still a significant modality gap between them.

To bridge this gap, most works [5, 6] are dedicated to improving feature representation for both modalities. Nowadays, CNN-based networks have been widely adopted in this task to extract sketch and 3D shape representations [6, 5]. Besides, projection-based methods [3, 7, 8, 6, 2] are commonly used to generate multi-view shapes for 3D shape representation, *i.e.*, apply the rendering model to project the 3D shape into 2D figures in different perspectives horizontally. However, these existing works mainly employ average pooling to fuse multi-view 3D shapes, whereas ignore the significant relationships between the views. On the other hand, the methods [3, 9, 6] roughly apply a CNN-based extractor to achieve a single sketch feature, while neglecting the notable data imbalance between 2D sketch and 3D shape.

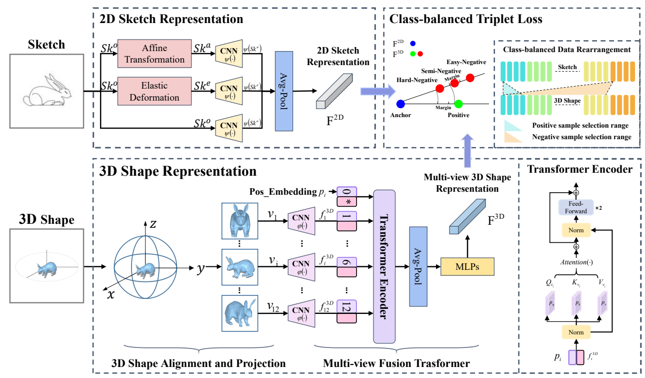


Fig. 1: Framework of our multi-view fusion Transformer for sketch-based 3D shape retrieval.

To address the above problems, we develop a novel framework based on Multi-view Fusion Transformer, which models the relationships between each view and fuse multi-view shape features effectively. As illustrated in Fig. 1, the multi-view 3D shape features extracted from a CNN backbone, accompanied with their position embeddings, are fed together into a Transformer encoder. Then 3D shape representation is generated through average pooling and MLPs. To overcome the data imbalance between 2D sketch and multi-view 3D shape, we propose to apply affine transformation and elastic deformation to augment the input sketch. Then we feed the augmented sketches into CNN-based extractor to obtain more diverse sketch features and adopt average pooling to fuse the augmented features. Furthermore, considering the significance of matching cross-modal representations, we design a class-balanced triplet loss, which can ensure each class constructs the same number of positive and negative samples. Extensive experiments on three widely-benchmarked datasets, including SHREC13, SHREC14, and

*equal corresponding author. This work was supported in part by the NSF of China under Grant Nos. 62102061, 62272083, 62306059 and U22B2052, and in part by the Liaoning Provincial NSF under Grant 2022-MS-137 and 2022-MS-128.

PART-SHREC14, demonstrate the superiority of our method, achieving promising improvements over previous competitive methods. Additional ablation studies valid the effectiveness of the components in our method.

2. METHOD

Our framework for sketch-based 3D shape retrieval contains three parts: 3D Shape Representation in Sec. 2.1, 2D Sketch Representation in Sec. 2.2, and Class-balanced Triplet Loss in Sec. 2.3. We further elaborate their details below.

2.1. 3D Shape Representation

Follow the commonly-used operation in [2], we render each 3D shape into twelve views with a 30° interval along the horizontal plane, *i.e.*, $V = \{v_1, v_2, ..., v_{12}\}$.

3D Shape Alignment and Projection. 3D shapes could be rotated and observed at any angle, thus different views of unaligned 3D shapes may show distinct projections as mentioned in [3]. The fewer projection areas may result in unclear recognition of the unaligned 3D shape class, affecting the cross-modal matching with 2D sketches. To this end, we introduce a 3D shape alignment module to alleviate this problem. Specifically, we can suppose that the perspective V with the larger projection area S is more important for 3D shapes. We determine that the initially selected plane is the horizontal plane, and the perpendicular plane is the overlook plane. In Eq. 1, we utilize S^h and S^o to denote the maximum area of horizontal and vertical plane projection, respectively.

$$\begin{aligned} S^h &= \max(S^h_{v_1}, S^h_{v_2}, ..., S^h_{v_{12}}) \\ S^o &= \max(S^o_{v_1}, S^o_{v_2}, ..., S^o_{v_{12}}) \end{aligned} \tag{1}$$

When S^h is smaller than S^o, in other words, if the effectiveness of vertical projection is greater than horizontal projection, the 3D shape should be rotated 90° in the vertical direction and then projected horizontally in each 30° cycle. This process can be formulated by Eq. 2.

$$V = \begin{cases} v^h_1, v^h_1, ..., v^h_{12} & if\ S^h > S^o \\ v^o_1, v^o_1, ..., v^o_{12} & else \end{cases} \tag{2}$$

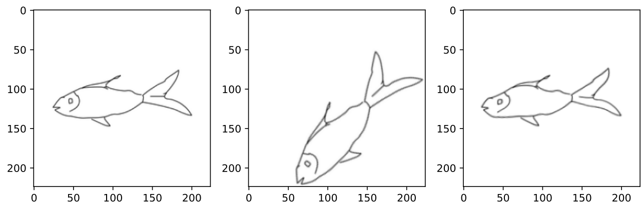
Multi-view Fusion Transformer. Subsequently, twelve projections are fed into 3D shape feature extractor $\varphi(\cdot)$, *i.e.*, Inception-ResNet-V2 [10], to generate projection features as $f^{3D}_i, f^{3D}_2, ..., f^{3D}_{12}$, where

$$f^{3D}_i = \varphi(v_i), 1 \le i \le 12. \tag{3}$$

Previous works on multi-view feature fusion generally include feature pooling and multi-view CNN, whereas cannot model feature correlation among all perspectives well. To alleviate this problem, we develop a multi-view feature fusion based on a transformer encoder to capture the correlations between different views. First of all, we employ a simple and effective way to mark each view position, denoted as $p_1, p_2, ..., p_{12}$, where $p_i = i$. Afterwards, the extracted feature vectors f^{3D}_i and position embeddings p_i are combined into the multi-view Transformer, which contains three linear layers $p_q(\cdot), p_k(\cdot), p_v(\cdot)$ to project 3D shape views into Q_{v_i}, K_{v_i} and V_{v_i}. The attention score is calculated by Q_{v_i} and K_{v_i} with dot product. The score after scale is multiplied by V_{v_i} to obtain the final weighted feature, which allows the integration of important parts of the multi-view features. It can be formulated with

$$\mathrm{Attention}(Q_{v_i}, K_{v_i}, V_{v_i}) = \mathrm{softmax}(\frac{p_q(v_i)p_k(v_i)^T}{\sqrt{d_k}})p_v(v_i) \tag{4}$$

We utilize a feed-forward network, which consists of two linear layers, to enhance feature representation and narrow the modality gap of the weighted features. Besides, we apply average pooling to fuse these features, which are then processed by MLPs to obtain the final 3D shape representation F^{3D}.



(a) Original input sketch Sk^o (b) Affine transformed sketch Sk^a (c) Elastic deformed sketch Sk^e

Fig. 2: Illustration of sketch augmentation with affine transformation and elastic deformation.

2.2. 2D Sketch Representation

Considering the fact that 2D sketch contains less information than 3D shape, we address this data imbalance issue by imposing affine transformation $Affine(\cdot)$ and elastic deformation $Elastic(\cdot)$ for sketch augmentation on original sketch Sk^o. In this way, it allows to produce multi-perspective sketches, which become more consistent with 3D shapes. We show an example in Fig. 2.

Affine Transformation. It enables the implementation of multiple geometric transformations on the original sketch, including translation, rotation, scaling, and shearing.

$$Sk^a = \mathrm{Affine}(Sk^o), \tag{5}$$

where Sk^o is the input sketch and Sk^a is the affine augmented one. We employ a random module to generate a 3×3 matrix as its required parameters to achieve the effect of random transformation. To be specific, a floating-point number between 0 and 1 will be randomly generated, and the affine transformation is applied to the sketch when this number is smaller than 0.3.

Elastic Deformation. It is adopted to simulate the deformation of elastic objects as an image transformation technique. In this way, the input sketch is distorted by applying local displacements to small image patches. The degree of deformation varies across the entire image, resulting in non-uniform deformation effects that increase data variability and randomness. In this paper, we set the scale factor of the displacement field with 80, and the standard deviation of the Gaussian filter used to smooth the displacement field is 10. The deformed sketch Sk^e is denoted as

$$Sk^e = \text{Elastic}(Sk^o). \tag{6}$$

After affine transformation and elastic deformation, all of the augmented sketches Sk^a, Sk^e and the original input sketch Sk^o are fed into the sketch feature extractor $\psi(\cdot)$, *i.e.*, ResNet-18 [11], for feature extraction. At last, we utilize average pooling on the three sketch representations to generate a final sketch feature F^{2D} as follows

$$F^{2D} = \text{Avg-pool}(\psi(Sk^o), \psi(Sk^a), \psi(Sk^e)). \tag{7}$$

2.3. Class-balanced Triplet Loss

To measure whether 2D sketch and 3D shape match or not, we set sketch as the anchor and choose its corresponding 3D shape as positive samples and other 3D shapes as negative. Euclidean distance is employed to calculate the similarity between sketches and 3D shapes,

$$d(F^{2D}, F^{3D}) = \sqrt{\sum_{i=0}^{n}(F_i^{2D} - F_i^{3D})}, \tag{8}$$

where n is the dimension of sketch and 3D shape feature vector, F_i^{2D} and F_i^{3D} represent vector values of the two features on dimension i. Generally, triplet loss is frequently used to estimate the loss cost. Different from previous methods, we design a class-balanced triplet loss, which can ensure each class constructs the same number of positive and negative samples:

$$\mathcal{L} = \max(d(F^{2D}, F_{pos}^{3D}) - d(F^{2D}, F_{neg}^{3D}) + m, 0), \tag{9}$$

where m is a margin between positive and negative pairs. Specifically, existing class and sample numbers in each dataset are different. Previous works cannot ensure more than one positive sample appeared in each batch, potentially affecting the appropriate selection of positive and negative samples in triplet loss and resulting in unstable training performance. To this end, we propose a class-balanced data rearrangement approach to fix class and sample numbers selected by each batch, ensuring each batch consists of the same type but different domains of 3D shapes corresponding to sketch anchors. For instance, N classes ($N = 16$ in our experiments) are selected in each batch, where four sketches and four 3D shapes are chosen, respectively.

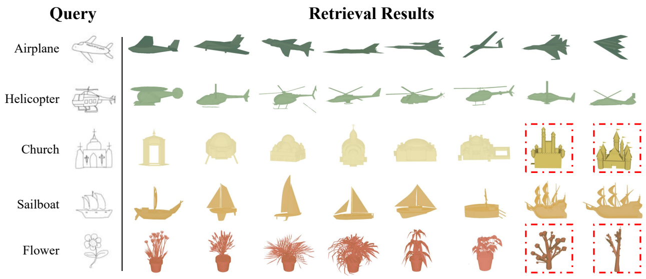


Fig. 3: 3D shape retrieval results for five sketch queries on SHREC13. The red dotted boxes indicate incorrect results.

3. EXPERIMENTS

3.1. Experimental Settings

Datasets and metrics. We perform the experiments on three datasets, *i.e.*, SHREC13 [21], SHREC14 [22], and PART-SHREC14 [15]. Following existing works [21, 22, 15], we employ six most commonly utilized evaluation metrics, including nearest neighbor (NN), first tier (FT), second tier (ST), E-measure (E), discounted cumulative gain (DCG) and mean average precision (mAP).

Implementation details. In our experiments, we employ ResNet-18 and Inception-ResNet-V2 to extract sketch and 3D shape features. Regarding the hyper-parameter settings, this model is trained for 300 epochs with a batch size of 64. We utilize Adam optimizer with a learning rate of 1×10^{-3}. StepLR is adopted as the weight adjuster with a decay period of every 650 iterations and a weight decay exponent of 0.9. The margin m for the triplet loss is set to 3.

3.2. Comparison with State-of-the-arts

Quantitative analysis. We compare with DCMLH [1], Semantic [15], DSSH [3], and other classic methods to verify the validity on three datasets, as shown in Tab. 1 and Tab. 2. For SHREC13, our method outperforms DSSH with an average performance boost of 2.1%, particularly achieving 6.5% gains in E metric. However, the performance of most methods declines on SHREC14, particularly by around 4% in certain metrics, which may be due to the increasing number of categories. Nonetheless, our method maintains its superiority in the last five metrics. Notably, our method achieves a remarkable 15% improvement in E compared to DSSH. As reported in Tab. 2, our method exhibits even greater boosts on PART-SHREC14. It outperforms DSSH by 5.5% in FT, 5.9% in ST, 6.7% in E, 3.4% in nDCG, and 5.5% in mAP.

Qualitative analysis. To qualitatively analyze the superior performance of our network, we conduct on the SHREC13 with various sketch queries. In Fig. 3, we present the top-8 retrieved results. As can be observed visually, most of the retrieval candidates are correct, however, there is also a small portion of false positives (in red dashed boxes). For

Table 1: Performance comparison with state-of-the-art methods on SHREC13 and SHREC14 datasets.

Method	SHREC13						SHREC14					
	NN	FT	ST	E	nDCG	mAP	NN	FT	ST	E	nDCG	mAP
DCMLH [1]	0.730	0.715	0.773	0.368	0.816	0.744	0.730	0.715	0.773	0.368	0.816	0.744
LWBR [12]	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 [1]	0.650	0.634	0.719	0.348	0.766	0.674	0.272	0.274	0.345	0.171	0.498	0.286
DCML-ResNet [1]	0.740	0.752	0.797	0.365	0.829	0.774	0.578	0.591	0.647	0.823	0.351	0.615
DCA [13]	0.783	0.796	0.829	0.376	0.856	0.813	0.770	0.789	0.823	0.398	0.859	0.803
Shape2Vec [14]	-	-	-	-	-	-	0.714	0.697	0.748	0.360	0.811	0.720
Semantic [15]	0.823	0.828	0.860	0.403	0.884	0.843	**0.804**	0.749	0.813	0.395	0.870	0.780
BV-CDSM [16]	0.811	0.830	0.878	0.406	0.885	0.849	0.784	0.796	0.831	0.404	0.868	0.809
DPSML [17]	0.819	0.834	0.875	0.415	0.892	0.857	0.774	0.798	0.849	0.415	0.877	0.813
SL [18]	0.827	0.841	0.895	0.418	0.895	0.864	0.781	0.804	0.847	0.415	0.872	0.820
DSSH-512 [3]	0.831	0.844	0.886	0.411	0.893	0.858	0.796	0.813	0.851	0.412	0.881	0.826
SSML [19]	**0.836**	0.833	0.883	0.411	0.896	0.853	-	-	-	-	-	-
Ours	0.826	**0.848**	**0.905**	**0.476**	**0.911**	**0.865**	0.795	**0.815**	**0.859**	**0.562**	**0.889**	**0.829**

Table 2: Performance comparison on PART-SHREC14 dataset.

Method	NN	FT	ST	E	nDCG	mAP
Siamese [20]	0.118	0.076	0.132	0.073	0.400	0.067
Semantic [15]	0.840	0.634	0.745	0.526	0.848	0.676
DSSH-512 [3]	0.838	0.777	0.848	0.624	0.888	0.806
Ours	**0.841**	**0.832**	**0.907**	**0.691**	**0.924**	**0.861**

Table 3: Effect of the multi-view Transformer on SHREC13.

Method	NN	FT	ST	E	nDCG	mAP
Ours w/o Transformer	0.817	0.836	0.891	0.470	0.903	0.855
Ours w/ Transformer	**0.826**	**0.848**	**0.905**	**0.476**	**0.911**	**0.865**

Table 4: Effect of sketch augmentation on SHREC13.

Method	NN	FT	ST	E	nDCG	mAP
Baseline	0.786	0.804	0.867	0.464	0.882	0.826
Elastic	0.795	0.812	0.881	0.465	0.889	0.835
Affine	0.803	0.819	0.887	0.469	0.894	0.842
Elastic&Affine	**0.809**	**0.826**	**0.896**	**0.470**	**0.899**	**0.848**

Table 5: Effect of different class numbers on SHREC13.

class_n	sample_n	NN	FT	ST	E	nDCG	mAP
64	1	0.786	0.804	0.867	0.464	0.882	0.826
16	4	**0.796**	**0.815**	**0.870**	**0.467**	**0.887**	**0.837**
8	8	0.791	0.810	0.867	0.463	0.886	0.833

the "church" category, the red boxes depict visually similar shapes but belong to different semantic categories, possibly due to varying perceptions of annotators. For the "Flower" category, the red boxes show closer semantic similarity to the input sketch but deviate significantly from the true category, highlighting significant intra-class variance in both sketch and 3D shape modalities.

3.3. Ablation Study

To validate the effectiveness of our method, we further report ablation studies on SHREC13 dataset. Notably, *we evaluate the effectiveness of class numbers and sketch augmentation based on the baseline, without any other optimization modules. Subsequently, we assess the fusion module on top of optimizing the preceding two modules.*

Effectiveness of class-balanced numbers. With a fixed batch size of 64, we design three settings for class number and sample number. As depicted in Tab. 5, as the increase of sample numbers per class, the performance initially rises and subsequently declines. This outcome can be attributed to the limited samples. When selecting too few samples, it can become challenging to construct effective triplets, and potentially lead to overfitting and poor performance. Furthermore, the class with four 3D shapes accounts for only 1/9 of the total datasets. Thus, blindly increasing the sample numbers may not increase positive samples, instead, results in hard samples being identified as positive samples, resuliting in a performance decrease.

Effectiveness of sketch augmentation. As can be seen in Tab. 4, the network performance is improved by approximately 0.8% with elastic deformation alone, by 1.43% with affine transformation alone, and by 2% when both deformations are combined together, compared to the network without any deformation. These findings confirm the hypothesis in this paper regarding the data imbalance between 2D sketch and 3D shape. It verifies that our sketch augmentation alleviates this problem effectively.

Effectiveness of multi-view transformer. As reported in Tab. 3, it is evident that our fusion strategy yields almost 1% performance improvement over the average pooling, which indicates our multi-view transformer can effectively capture associations between different views.

4. CONCLUSION

We have proposed to address sketch-based 3D shape retrieval via a Multi-view Fusion Transformer, which utilizes a Transformer mechanism to fuse multi-view 3D shape projections and learn the correlation between the views. Besides, we apply affine transformation and elastic deformation for sketch augmentation to produce multi-perspective sketch features. Moreover, our class-balanced triplet loss improves the training process. Extensive experiments validate the effectiveness of our method on three benchmarks in terms of various metrics. In future work, we will focus more on intra-class variance in sketch quality and the existence of abnormal sketches.

5. REFERENCES

[1] Guoxian Dai, Jin Xie, and Yi Fang, "Deep correlated holistic metric learning for sketch-based 3d shape retrieval," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3374–3386, 2018.

[2] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *ICCV*, 2015, pp. 945–953.

[3] Jiaxin Chen, Jie Qin, Li Liu, Fan Zhu, Fumin Shen, Jin Xie, and Ling Shao, "Deep sketch-shape hashing with segmented 3d stochastic viewing," in *CVPR*, 2019, pp. 791–800.

[4] Anran Qi, Yulia Gryaditskaya, Jifei Song, Yongxin Yang, Yonggang Qi, Timothy M. Hospedales, Tao Xiang, and Yi-Zhe Song, "Toward fine-grained sketch-based 3d shape retrieval," *IEEE Transactions on Image Processing*, vol. 30, pp. 8595–8606, 2021.

[5] Hanhui Li, Hefeng Wu, Xiangjian He, Shujin Lin, and Xiaonan Luo, "Multi-view pairwise relationship learning for sketch based 3d shape retrieval," in *IEEE International Conference on Multimedia Expo*, 2017.

[6] Jie Qin, Shuaihang Yuan, Jiaxin Chen, Boulbaba Ben Amor, Yi Fang, Nhat Hoang-Xuan, Chi-Bien Chu, Khoi-Nguyen Nguyen-Ngoc, Thien-Tri Cao, Nhat-Khang Ngo, et al., "Shrec'22 track: Sketch-based 3d shape retrieval in the wild," *Computers & Graphics*, vol. 107, pp. 104–115, 2022.

[7] Yutong Feng, Yifan Feng, Haoxuan You, Xibin Zhao, and Yue Gao, "Meshnet: Mesh neural network for 3d shape representation," in *Proceedings of the AAAI conference on artificial intelligence*, 2019, vol. 33, pp. 8279–8286.

[8] Ishan Misra, Rohit Girdhar, and Armand Joulin, "An end-to-end transformer model for 3d object detection," in *ICCV*, 2021, pp. 2906–2917.

[9] Yan Jiang, Wei Quan, Cheng Han, and Hua Li, "Sketch-based 3d shape retrieval via attention," in *2020 International Conference on Virtual Reality and Visualization (ICVRV)*. IEEE, 2020, pp. 5–9.

[10] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016.

[11] Muhammad Shafiq and Zhaoquan Gu, "Deep residual learning for image recognition: A survey," *Applied Sciences*, vol. 12, no. 18, pp. 8972, 2022.

[12] Jin Xie, Guoxian Dai, Fan Zhu, and Yi Fang, "Learning barycentric representations of 3d shapes for sketch-based 3d shape retrieval," in *CVPR*, 2017, pp. 5068–5076.

[13] Jiaxin Chen and Yi Fang, "Deep cross-modality adaptation via semantics preserving adversarial learning for sketch-based 3d shape retrieval," in *ECCV*, 2018, pp. 605–620.

[14] Flora Ponjou Tasse and Neil Dodgson, "Shape2vec: semantic-based descriptors for 3d shapes, sketches and images," *ACM Transactions on graphics (TOG)*, vol. 35, no. 6, pp. 1–12, 2016.

[15] Anran Qi, Yi-Zhe Song, and Tao Xiang, "Semantic embedding for sketch-based 3d shape retrieval.," in *BMVC*, 2018, vol. 3, pp. 11–12.

[16] Yongzhe Xu, Jiangchuan Hu, Kanoksak Wattanachote, Kun Zeng, and YongYi Gong, "Sketch-based shape retrieval via best view selection and a cross-domain similarity measure," *IEEE Transactions on Multimedia*, vol. 22, no. 11, pp. 2950–2962, 2020.

[17] Yinjie Lei, Ziqin Zhou, Pingping Zhang, Yulan Guo, Zijun Ma, and Lingqiao Liu, "Deep point-to-subspace metric learning for sketch-based 3d shape retrieval," *Pattern Recognition*, vol. 96, pp. 106981, 2019.

[18] Hairui Yang, Yu Tian, Caifei Yang, Zhihui Wang, Lei Wang, and Haojie Li, "Sequential learning for sketch-based 3d model retrieval," *Multimedia Systems*, pp. 1–18, 2022.

[19] Yu Xia, Shuangbu Wang, Lihua You, and Jianjun Zhang, "Semantic similarity metric learning for sketch-based 3d shape retrieval," in *ICCS*. Springer, 2021, pp. 59–69.

[20] Fang Wang, Le Kang, and Yi Li, "Sketch-based 3d shape retrieval using convolutional neural networks," in *CVPR*, 2015, pp. 1875–1883.

[21] Bo Li, Yijuan Lu, Afzal Godil, Tobias Schreck, Masaki Aono, Henry Johan, Jose M Saavedra, and Shoki Tashiro, *SHREC'13 track: Large scale sketch-based 3D shape retrieval*, 2013.

[22] 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," in *Eurographics workshop on 3D object retrieval*, 2014, vol. 2014, pp. 121–130.