Fine-Grained VR Sketching: Dataset and Insights.

Ling Luo[1,2] Yulia Gryaditskaya[1,2] Yongxin Yang[1,2] Tao Xiang[1,2] Yi-Zhe Song[1,2]

[1]SketchX, CVSSP, University of Surrey [2]iFlyTek-Surrey Joint Research Centre on Artificial Intelligence

Abstract

We present the first fine-grained dataset of 1,497 3D VR sketch and 3D shape pairs of a chair category with large shapes diversity. Our dataset supports the recent trend in the sketch community on fine-grained data analysis, and extends it to an actively developing 3D domain. We argue for the most convenient sketching scenario where the sketch consists of sparse lines and does not require any sketching skills, prior training or time-consuming accurate drawing. We then, for the first time, study the scenario of fine-grained 3D VR sketch to 3D shape retrieval, as a novel VR sketching application and a proving ground to drive out generic insights to inform future research. By experimenting with carefully selected combinations of design factors on this new problem, we draw important conclusions to help follow-on work. We hope our dataset will enable other novel applications, especially those that require a fine-grained angle such as fine-grained 3D shape reconstruction. The dataset is available at `tinyurl.com/VRSketch3DV21`.

1. Introduction

Virtual Reality (VR) headsets and 3D printers rapidly make their way to consumer markets. With recent interest in virtual reality, VR sketching is gaining increasingly more popularity in industry[1] and academia [58, 49, 33, 77, 81, 3]. In this work, we investigate the potential of *low-effort* VR sketching to become a bridge to the practical adoption of 3D and VR-related technologies by average consumers and professional designers.

In sketching research, the recent focus is on fine-grained tasks revolving around subtle intra-class differences [83, 7, 51, 85, 86]. In particular, 2D sketches were proved to be efficient queries for 2D images fine-grained retrieval [62, 82, 83, 14]. Yet, in the context of the 3D shape retrieval from single or multiple 2D sketches the fine-grained performance was not demonstrated so far [16, 69, 87, 76, 32].

2D sketches contain a 2D projection ambiguity, where multiple 3D shapes can project to the same 2D sketch [75], while sketching the same shape from multiple 2D viewpoints is a non-trivial task. Furthermore, even professionals find it challenging to accurately depict shape proportions and dimensions in a 2D sketch [60]. [50] conclude that sketching in 3D gives better shape understanding. 3D sketching allows to (1) alleviate the problem of a 2D projection ambiguity and (2) naturally evaluate and depict shapes' proportions (Fig. 1). Driven by these observations, we study, for the first time, the problem of *fine-grained* intra-category 3D sketch to 3D shape retrieval.

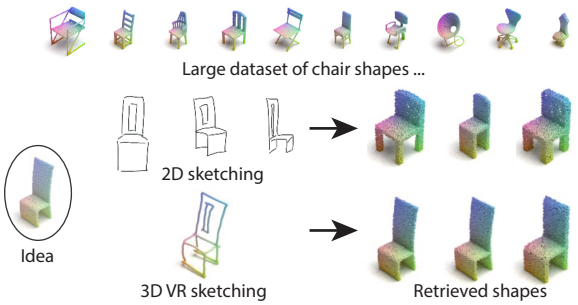


Figure 1. We investigate the potential of low-effort VR sketching as the enabler of the practical adoption of 3D and VR-related technologies and study *fine-grained* 3D sketch to 3D shape retrieval.

Little work thus far was done on 3D sketch to 3D shape retrieval, exploiting either a small set of inaccurate sketches collected with Microsoft Kinect [37, 80], or dense colorful sketches drawn on top of the initial 3D model[20]. Research on free-hand VR sketches is hindered by a lack of training data. As a first step, Luo et al. [47] collected a small dataset of 167 sketches for testing, and proposed a heuristic method to generate 3D sketches with different abstractness levels. They were the first to study the retrieval problem from sparse human VR sketches, and showed that a reasonable *inter-class* retrieval accuracy on human sketches can be achieved by training on synthetic sketches, even though inferior to the accuracy on synthetic data. Nevertheless, the *intra-class* Top-1 accuracy of their method is low. We aim at increasing intra-class accuracy, enabling the retrieval of a particular instance of a shape within a single category (fine-grained). We meticulously evaluate alternative network designs, comparing several state-of-the-art encoders [55, 73] and losses [71, 34].

[1]https://www.gravitysketch.com/, https://www.tiltbrush.com/ and https://coolpaintrvr.com/en/

While in the absence of human sketches, synthetic sketches are the only option, the method for synthetic sketch generation proposed in [47] requires an input mesh to be a clean manifold curvature-aligned quad-dominant mesh. For new categories, collecting human sketches might be more feasible than obtaining 3D shapes with required properties. However, collecting sketch datasets is a labor-intensive task [15, 25, 88, 23]. We therefore focus on a single chair category and collect 1,497 fine-grained 3D VR sketch and shape pairs, which we use to drive out insights on the desirable properties of 3D VR sketch datasets for fine-grained tasks. We select the chair category due to the large variability of shapes, dimensions, and details, including shapes with variable genus values, shapes dominated by planar or non-planar surfaces (Fig. 3). Such large intra-class variability makes this category ideal for our fine-grained goal. We perform exhaustive evaluations on (i) how the dataset size and its composition affect retrieval accuracy, (ii) what the expected performance gain from training on human sketches compared to synthetic data is, (iii) how human and synthetic sketches can complement each other, (iv) the role of data augmentation, (v) how human ability to sketch in 3D increases over time, and (vi) how sketching style differences affect the performance.

Targeting practical novel applications, we assert a number of guiding principles to the overall design of our dataset: (P1) *convenience*: we argue for the most convenient sketching scenario where the sketch consists of sparse lines and does not require any sketching skills, prior training or time-consuming accurate drawing; (P2) *fine-grained sketching*: we require sketches to be fine-grained, capturing salient visual details of a *paired* 3D shape, that sufficiently differentiates it from other 3D sketches within the same category; (P3) *free-space sketching*: we require sketches to faithfully reflect humans' ability to sketch in free space; and (P4) *diversity*: we require sketches to contain a diversity of styles and levels of details. We capture the sketching process, recording the time-space coordinates of each stroke.

In summary, our contributions include: (i) the first large-scale dataset of paired human 3D VR sketches and 3D shapes, (ii) key insights associated with the dataset to inform future work on 3D VR sketches, (iii) the first in-depth study of a proving ground application: fine-grained 3D sketch based 3D shape retrieval, and conclude with (iv) a series of key technical insights to guide future research.

2. Related work

2.1. Sketching datasets

When collecting 2D sketches by novices it is possible to rely on crowd-sourcing and readily available hardware [15, 25, 88, 59]. Our task is more challenging since VR headsets are not yet commonly available, and that VR sketching is still a relatively novel concept with limited off-the-shelf tools available. The earlier works on sketching were targeting category-level sketch understanding, such as the TUBerlin dataset [15] which consists of 20,000 sketches in total but only 80 sketches per category. Recent work focuses on fine-grained tasks, offering datasets consisting of 1-2 categories with a larger number of sketches. *E.g.*, [82] proposed a dataset of two categories, shoes and *chairs*, with 419 and 297 sketch-photo pairs, accordingly. [86] proposed a dataset of 1,500 professional 2D sketches collected across 500 3D chair shapes. Similarly, recent dataset AmateurSketch-3DChair[2] consists of 3 2D views by novices of 1,005 3D shapes. Following this trend, and to enable the comparison with 2D sketch-based retrieval, we collect the 1,497 3D VR sketches for the same 1,005 shapes. 492 shapes are sketched by two participants each.

2.2. Retrieval

2D sketch/image to 3D shape/ 2D image Triplet loss and its variants [71, 61, 69, 82, 74, 9, 22, 83, 65] are by far the most commonly used losses for the retrieval task. A number of recent losses [45, 42, 43] were shown to improve category-level retrieval performance, but are less suitable for fine-grained retrieval. We explore here a very recently proposed contrastive loss by Khosla et al [34]. Compared to the triplet loss, their formulation allows to have multiple positive and negative samples and was shown to better learn inter- and intra-class variability.

3D sketch to 3D shape 3D sketch to 3D shape retrieval is a relatively novel field [38, 37, 80, 20, 47]. The progress is hampered by a lack of datasets of human 3D sketches. Most of the existing works [38, 37, 80] relied on the dataset collected using Microsoft Kinect, which has limited tracking accuracy. As a result, these sketches have low fidelity and exhibit less details, making it impractical for fine-grained methods. Moreover, this dataset is no longer accessible online. Luo et al. [47] addressed the category-level retrieval on sparse 3D VR sketches created by novices. They proposed a synthetic data generation method from 3D shapes and collected a small dataset of human sketches that they use as test data. They showed that the performance on human sketches drops compared to the performance on synthetic data. Given this drop, here we study how training on human sketches affects the retrieval performance, and how it compares with training on synthetic sketches. Unlike [47] we focus on the fine-grained retrieval and drive out insights on the desirable structure of 3D VR sketch datasets.

3D shape to 3D shape Closely related to our problem is 3D shape-based 3D shape retrieval. In 3D shape to 3D shape retrieval it is common to represent shapes via multi-view projections [44, 28, 27, 42]. The 3D VR sketches

[2] http://sketchx.ai/downloads/: AmateurSketch-3DChair

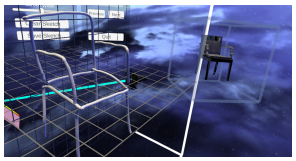
we are dealing with are sparse, and, as it was shown in [47], multi-view representations lose to point cloud based ones on a problem similar to ours. Deng et al. [13] exploit point pair features, based on the normal vector values, to align the point clouds. Such an approach can not be efficiently applied to 3D sketches, since the normal to a line is not uniquely defined in 3D, and the notion of patch is not well defined on a sparse sketch. Uy et al. [67] proposed a novel framework for deformation-aware 3D retrieval based on point cloud shape representation. It is not straightforward to extend this work to 3D sketches, since it requires ensuring the deformation of a 3D shape is consistent with a *sparse* sketch. Dahnert et al. [10] studied fine-grained retrieval from 3D scans to CAD models. Similarly to previous works, they train with the triplet loss, which we also leverage in our work.

2.3. Point cloud networks

In our work we represent sparse VR sketches as point clouds. Point cloud shape representation received a lot of attention in recent years, being a natural data format for 3D scans. Many works were proposed for representation learning, and un/conditional point clouds generation. They can be divided into those based on encoder-decoder architectures [17, 1, 79, 66], those with adversarial training [18, 1, 39, 19, 63, 31, 68, 70], and those that exploit normalizing flows [78, 53, 35]. Some works address point clouds completion and denoising [11, 12, 84, 40, 10, 5, 72, 56]. Other works target learning shape properties [24, 6, 52, 46]. The most relevant to us are the works on point clouds encoding [57, 54, 55, 4, 73, 41]. We evaluate in our work the two most commonly used encoders: *PointNet++* [55] and *DGCNN* [73].

3. Data collection

Interface & Task We base our data-collection interface on the one proposed in [47][3], where participants were sketching over the reference shape. In our work, to better capture accuracy of sketching in free space, the reference is shown in 3D in an area separate from the sketching area, where 3D shape can be freely rotated, as shown in the inset. To improve space perception we displayed guiding grids [2].



Compared to [47], we let participants choose line width, best matching the intention of the sketcher: thinner lines allow to create more detailed sketches, while thicker lines can be used to quickly get a general shape. This allows us to better capture the diversity of human styles.

To get familiar with sketching in 3D all the participants are asked to go through the following steps: (1) get familiar with hand controllers and the functions of each button and trigger: sketch, undo last stroke, delete all strokes, menu click button; (2) practice grabbing and rotating reference space and sketching space; (3) practice adjusting line width (4) practice drawing random lines. There was no training on how to sketch in 3D.

[3]https://tinyurl.com/3DSketch3DV

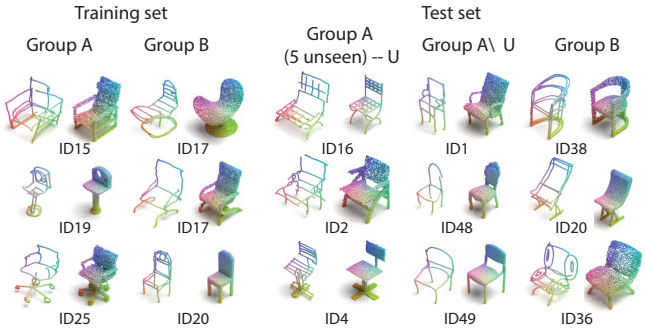


Figure 2. Example sketch-shape pairs from the training and test sets. Sketches in our dataset cover large variability of shapes and vary in amount of details and accuracy. Since the participants from group B (Sec. 3) draw more shapes than the participants from group A, their sketches tend to be more accurate and contain more details. It is interesting to observe how participants depict solid surfaces, some draw only ridge lines and some draw additional sets of parallel lines lying on the surface of a shape. Sometimes both types of depictions can be observed in the sketches of the same participant, e.g. the sketches of the participant 17 (ID17).

Shapes We collected 1,497 sketches for a subset of 1,005 3D chairs in ShapeNetCore [8]. The chairs category is chosen due to a large intra-class shapes variability. The selected subset is representative of the full set of shapes.

Participants We split participant into two groups (Fig. 2).

Group A: We hired 46 participants who sketched 10 non-overlapping shapes each. The number of shapes was limited for each participant to avoid fatigue and to limit the learning effect on sketching style.

Group B: This group consists of 4 participants, who sketched more. They sketched: 21, 72, 330 and 120 unique shapes. The last participant additionally sketched 492 shapes overlapping with the shapes from group A. Sketches from these participants allow us to study how sketching style evolved over time, and how it affects retrieval results. For instance, Fig. 3 shows, on an example of one participant, that the sketches are getting more accurate and detailed over time.

At the end of the sketching task, the participants were asked to complete a questionnaire. The age of the participants varies between 18 and 40 years old. The average sketching experience on a Likert scale is 1.84, where 1 means no sketching experience and 5 means professional. 20 of the participant reported that they usually do not sketch, 23 do sketches several times a year, 5 several times a

Figure 3. We selected the sketches by the participant 20, evenly spaced in time of creation. This participant sketched the largest amount of shapes. The figure shows shape-sketch pairs, ordered by time of creation: from first to last. It can be seen that the later sketches in overall better capture the proportions of the shapes and contain more details than the first ones.

month and 2 several times per week. 33 participants had no prior experience with VR. The average score of how comfortable the participants felt sketching in our interface was 3.44, where 1 means not comfortable and 5 means comfortable. Being above average it indicates that even inexperienced participants found sketching in 3D satisfying.

Separation into train, test and validation. In our experiments, unless specified otherwise, we use only 1,005 sketches of unique shapes. To split the collected data into three sets, we first held all the sketches of five randomly selected participants from group A as a test set to check that the compared methods generalize well to unseen human styles. For the rest of participants we split their sketches of unique shapes proportionally between the training, validation and test sets with the ratios $7:1:2$. In total, the training set consists of 702 sketches, the validation set of 101 sketches, and the test set of 202 sketches. We exploit the validation set to choose networks' hyper-parameters. See Fig. 2 for visualization of the sketches from test and training sets.

4. Retrieval

Following the analysis from [47] we select the point cloud representation for 3D sketches and shapes. To train for *fine-grained* retrieval, we compare a traditionally used triplet loss [71], and a more recent contrastive loss [29, 34].

4.1. Data preparation & augmentation

All the target shapes in the ShapeNet dataset are oriented consistently. Since the sketches were drawn in the free-space, guided only by a sparse grid, the sketches' global orientations can be arbitrary, but are aligned with the grid axes. In the collected dataset 68 sketches have vertical rotation (z-axis) inconsistent with the orientation of the 3D shapes, and 19 have x/y-axis rotations. We experiment with two versions of the dataset: (1) where we keep the original rotation of sketches and (2) where the sketches are manually rotated to match the orientation of the 3D shapes. All the shapes and sketches are centered and normalized to have a maximum dimension along x-,y-,z-axes to be equal to 1.

Since collecting human sketches is a labor-intensive and time-consuming task, we experiment with several augmentation strategies. First, we apply randomly selected rotations in the range $[0, 360]$ degrees around the vertical axis to the human sketches. Second, we apply global distortions by scaling sketches along 3 axes with 3 scale factors in a range $[0.9, 1.1]$. Both augmentations are applied to input on the fly when training. Finally, we complement humans sketches with synthetic sketches, generated as was proposed in [47] with the abstractness level 1.0. This setting was shown to be one of the optimal ones when tested on human sketches.

4.2. Evaluated losses

Triplet loss Triplet loss ensures that the anchor-negative distances are larger than the anchor-positive distances by a given margin. Let I be a set of indices of all objects in the given mini-batch. For each object we have the corresponding sketch and shape point clouds. The triplet loss is a sum over all the triplets within a mini-batch:

$$L_{TL} = \sum_{a \in I} \max\{0, ||s^a - z^a_{pos}||_2^2 - ||s^a - z^b_{neg}||_2^2 + m\}, \quad (1)$$

where $||\cdot||_2$ is the Euclidean distance, and the feature space is normalized to a unit hypersphere. We use an embedding s^a of a 3D sketch as an anchor. We use an embedding z^a_{pos} of the 3D shape, that sketch corresponds to, as a positive example. A negative example can be an embedding z^b_{neg} of any other shape in the batch, $b \in I \setminus a$. We set the margin m to 0.3, a common choice for a fine-grained retrieval task. We also found this value to be optimal on our validation set.

Contrastive loss Contrastive losses proposed by [48] and [34] pursue the same goal as a triplet loss, but its formulation allows multiple negatives in the first case, and multiple positive and negative examples in the second case. These losses were shown to be beneficial over the triplet loss on certain tasks. With one positive and multiple negative examples the contrastive loss is defined as:

$$L_{CL} = -\sum_{a \in I} \log \frac{\exp\left(s^a \cdot z^a / \tau\right)}{D(a)}, \quad (2)$$

similarly to our triplet loss formulation, we use as an anchor an embedding of a 3D sketch. As a positive example we use an embedding of a matching 3D shape. τ is a temperature

parameter set to 0.1 in our experiments, empirically found to give the best results on the validation set.

We experimented with multiple choices of the denominator $D(a)$, and found the one which includes distance between the encodings of sketches and shapes, sketches, and shapes gives the best results:

$$D(a) = \sum_{b \in I} \exp\left(s^a \cdot z^b / \tau\right) + \sum_{b \in I \setminus a} \exp\left(s^a \cdot s^b / \tau\right) + \\ + \sum_{b \in I \setminus a} \exp\left(z^a \cdot z^b / \tau\right). \quad (3)$$

5. Experiments and Analysis

Our backbone architecture consists of an encoder with shared weights for sketches and shapes, trained with the triplet loss. We then study the choice of an encoder, an architecture and a loss, which allows us to achieve the best performance on our proving ground application – fine-grained retrieval. We also investigate the effect of human learning, differences in styles and level of details on the retrieval performance. Finally, we compare the synthetic sketches with human ones, and demonstrate the advantage of 3D sketches over 2D ones.

We evaluate all the methods in terms of Top-k accuracy ($A@k$), defined as a percentage of queries, which have their ground-truth shapes within k closest retrieval results. Our test set consists of 202 aligned human sketches (Sec. 3), and we perform retrieval from 5,794 chair shapes in ShapeNet dataset [8], excluding the shapes used in the training and validation sets. All the methods are run three times, and are trained for 300 epochs. We select the run and the epoch, which give the best $A@1$ on the validation set.

We use the following abbreviations in the tables: HS = human sketches, SS = synthetic sketches, CN = curve networks, TL = triplet loss, CL = contrastive loss. For all the baselines we use a mini-batch size of 6, unless specified differently. We use a bold font to highlight the best result in each column, and underline the second best. # indicates an experiment number.

5.1. Network design

Encoder choice While PointNet[54] and PointNet++[55] are a frequent choice of encoders, we also compare their performance with the more recent DGCNN [73]. On the sketches with inconsistent orientations DGCNN performs worse than PointNet++ (Table 1 #1 vs. #3). When the sketches are preliminarily aligned to have a consistent orientation (see Sec. 4.1), the gap in performance of the considered encoders is smaller (Table 1 #4 vs. #6). These experiments show that PointNet++ can be a better choice for inaccurate human sketches and can handle a small number of non-consistently oriented sketches in the training set (Table 1 #1 vs. #4).

	Training			Test set (202 → 5,794)		
#	**Size**	**Data**	**Method**	**A@1**	**A@5**	**A@10**
1	702	HS ShapeNet	PointNet++ Siam. TL	**26.2**	**43.1**	**54.5**
2	702	HS ShapeNet	PointNet++ Heter. TL	19.8	35.2	47.5
3a	702	HS ShapeNet	DGCNN Siam. TL	20.3	33.2	40.1
3b	702	HS ShapeNet	DGCNN Siam. TL	22.3	33.2	37.1
4	702	HS ShapeNet (aligned)	PointNet++ Siam. TL	<u>24.8</u>	<u>41.6</u>	48.0
5	702	HS ShapeNet (aligned)	PointNet++ Heter. TL	18.3	37.6	<u>48.5</u>
6	702	HS ShapeNet (aligned)	DGCNN Siam. TL	22.8	35.6	46.5

Table 1. The evaluation of encoders and architectures: siamese vs. heterogeneous. See Sec. 5.1 for the details. For all the baselines in this table we use a mini-batch size of 6, apart from #3b, where the mini-batch size is equal to 24.

	Training			Test set (202 → 5,794)		
#	**Size**	**Data**	**Method**	**A@1**	**A@5**	**A@10**
1	702	HS ShapeNet	PointNet++ Siam. TL	**26.2**	**43.1**	**54.5**
7	702	HS ShapeNet	PointNet++ Siam. CL	8.9	23.8	31.2
8	702	HS ShapeNet	DGCNN Siam. CL	<u>22.3</u>	<u>42.1</u>	<u>49.5</u>

Table 2. The evaluation of compared losses. See Sec. 5.1 for the details. For all the experiments we use a mini-batch size of 6, apart from the experiment #8, where the mini-batch size is set to 24.

Siamese vs. Heterogeneous We compare the Siamese architecture (the sketch and shape encoders share weights) with a heterogeneous architecture (the sketch and shape encoders do not share weights). We observed that if training from scratch, the heterogeneous architecture was not able to converge. We thus initialize the training with the weights of the encoder from the Siamese architecture at the 100th epoch. Experiments #1 vs. #2, and #4 vs. #5 in Table 1 indicate that heterogeneous training is inferior to Siamese on our training data. For Siamese training to work well, one needs to ensure that two different representations of the same instance can be described by the same embedding network. In [52], it was observed that when the PointNet encoder is trained on 3D shapes only, there is a minimal subset of points from the original 3D shape point set (a critical point set), such that the embedding of this subset is the same as the embedding of the original 3D shape point set. We observe that the critical point sets resemble abstraction of 3D shapes with sparse lines in human VR sketches. Therefore, Simaese training results in a single efficient encoder for 3D shapes and sketches, which additionally accounts for how humans represent a 3D shape with sparse lines.

Losses comparison Contrastive loss in [34] is shown to be dependent on the mini-batch size. Due to a GPU memory limitation our mini-batch size is limited to 6 shape-sketch

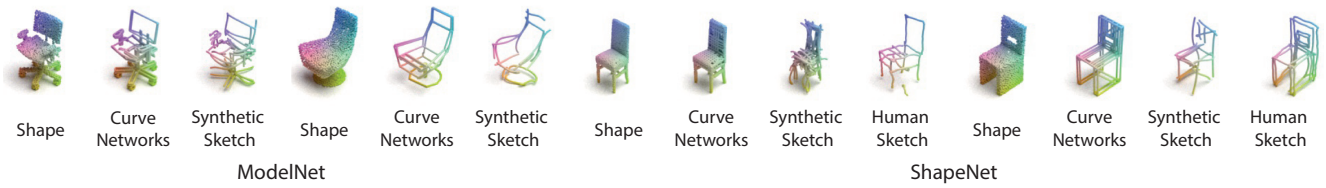


Figure 4. Visual comparison of human sketches, with synthetic sketches and curve networks on two shapes datasets: ModelNet and ShapeNet. The curve network extraction step is very sensitive to the quality of the input mesh, naturally it expects a mesh to be a manifold curvature aligned quad mesh, what requires heavy time-consuming mesh processing, where most of readily available remeshing tools either break the shape manifoldness or are unable to process complicated input.

pairs in case of PointNet++. DGCNN has less parameters and we are able to run the training with 24 sketch-shape pairs per mini-batch. In this case it can be seen in Table 2 #7 vs. #8 that training with DGCNN is advantageous over training with PointNet++ when using the contrastive loss. Nevertheless, all the results with the contrastive loss lose in performance to the triplet loss, due to a limited batch size.

5.2. Training data

5.2.1 Training data size

We analyze how the retrieval performance varies depending on the number of available human sketches. For this experiment, we randomly select $x\%$ sketch-shape pairs of the full training set. For each $x\%$ we run an experiment three times with different random selection of sketches. In these experiments we used PointNet++ encoder and TL. We select the checkpoint which gives the best results on the validation set, which is fixed for all the experiments in this paper. It can be seen in Fig. 5 that the most rapid improvement due to adding more sketches happens up to 60% of all human sketches in our training set. When we use between 80% to all the sketches the retrieval accuracy starts to stabilize, indicating that simply adding more data might be not enough to boost the performance.

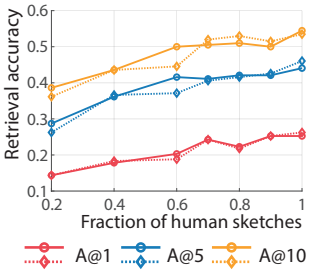


Figure 5. Retrieval accuracy vs. percentage of human sketches. The checkpoint was chosen based on best A@1 (solid lines) or best A@5 (dashed lines) on the validation set.

5.2.2 Human sketches vs Synthetic sketches

We compare the training on *human sketches (HS)*, with training on *curve networks (CN)* extracted from 3D shapes with [21] and on *synthetic sketches (SS)*, obtained from curve networks using the code from [47]. We generate curve networks and synthetic sketches for the set of shapes for which we collected human sketches. Since the method [21] expects the mesh to be an edge-manifold mesh, we process all the meshes using the algorithm by Huang et al. [30], followed by a decimation step to reduce meshes size. We additionally use randomly picked 702 SSs for a subset of ModelNet shapes, provided by [47]. Fig. 4 shows a visual comparison between HSs, CNs and SSs. It can be seen in Table 3 #1 vs. #9-11 that training with human sketches marginally outperforms training with either of SSs or CNs.

	Training		Test set (202 → 5,794)		
#	Size	Data	A@1	A@5	A@10
1	702	HS ShapeNet	**26.2**	**43.1**	**54.5**
9	702	SS ModelNet	11.9	25.7	32.7
10	702	CN ShapeNet	7.4	21.3	26.2
11	702	SS ShapeNet	13.4	25.3	30.7

Table 3. Human sketches vs. synthetic sketches. All the networks in this table use PointNet++ [55] as an encoder, and are trained with the Triplet Loss and Siamese architecture.

5.2.3 Augmentation on human sketches

The previous section concludes that the retrieval accuracy on human sketches when training only on synthetic sketches falls short of the retrieval accuracy when the training is done on human sketches, However, we observe that synthetic sketches can be used in addition to human sketches to boost the performance of the fine-grained retrieval: Table 4 (#12 vs. #l, and #13 vs. #4) shows that $A@5/10$ are consistently improved.

	Training data size and type		Test set (202 → 5,794)		
#	Sketches/ shapes	Data	A@1	A@5	A@10
1	702 / 702	HS ShapeNet	26.2	43.1	54.5
4	702 /702	HS ShapeNet (aligned)	24.8	41.6	48.0
12	702+702 / 702+702	HS ShapeNet + SS ModelNet	24.3	**48.0**	**55.9**
13	702+702 / 702+702	HS ShapeNet (aligned) +SS ModelNet	24.3	44.1	53.0
14	702+Aug./ 702	HS ShapeNet +Aug. distortions	**28.2**	44.1	55.0
15	702+Aug./ 702	HS ShapeNet +Aug. rotations	19.8	35.6	45.1
16	702+363 / 702	HS ShapeNet + Aug. style	24.3	45.5	54.0
17	702+363 / 702	HS ShapeNet + SS ShapeNet	24.3	41.6	49.0

Table 4. The evaluation of augmentation strategies. All the networks in this table use PointNet++ [55] as an encoder, and are trained with the Triplet Loss and Siamese architecture.

Table 4 (#14, vs. #12 and vs. #1) shows that a simple data augmentation of human sketches by applying random distortions along x,y,z-axes can lead to a higher top-1 accuracy than training on the combination of human sketches and synthetic sketches. Despite that PointNet++ performs better on a non-aligned dataset (Table 1 #1 vs. #4), when we apply rotations on sketches as an augmentation strategy the performance drops (Table 4 #15). This implies that in the practical retrieval application either users should provide the orientation of a sketch or the retrieval system should address this problem explicitly.

Additionally, we evaluate how the fine-grained retrieval performance changes if we provide for each shape more than one sketch. In the experiment 16, we train on 702 sketches from our training set augmented with 363 sketches by one of the participants from group B, providing a second sketch representation for 363 3D shapes. In the experiment 17, we instead augment with 363 synthetic sketches of the same shapes. It can be seen that having several shapes drawn by difference participants does not bring much (Table 4 #16 vs. #1). Using synthetic sketches gives a negative effect on the retrieval performance (Table 4 #17 vs. #1). This allows to drive out an important conclusion: *When collecting a dataset one should aim at maximizing the diversity of shapes and styles rather than aiming at having several shapes drawn by several different participants.*

To further understand the effect of data augmentation with synthetic sketches, we study augmentation when only 40% of our training set of human sketches is used (Table 5), and vary the number of synthetic sketches. We observe that augmenting with synthetic sketches improves the $A@1/5/10$ till the number of synthetic sketches is 1.5 to 2.0 times higher than the number of human sketches. We thus conclude that a certain number of synthetic sketches can help increase the retrieval accuracy, but the ratio between human and synthetic sketches should be taken into account. Adding more synthetic sketches results in performance degradation. Augmentation by sketches distortions still achieves the highest $A@1$ (last line in Table 5).

	Test set (202 → 5,794)		
Data	**A@1**	**A@5**	**A@10**
281 HS ShapeNet	17.82	36.14	43.56
281 HS ShapeNet + 140 SS ModelNet	19.31	36.14	43.56
281 HS ShapeNet + 280 SS ModelNet	20.79	36.14	45.54
281 HS ShapeNet + 421 SS ModelNet	20.30	**43.07**	49.50
281 HS ShapeNet + 561 SS ModelNet	20.30	39.60	**50.00**
281 HS ShapeNet + 702 SS ModelNet	19.80	36.63	48.02
281 HS ShapeNet + Aug. distortions	**21.78**	38.61	44.06

Table 5. The detailed evaluation of augmentation strategies.

5.3. Generalization across users

As described in Sec. 3 we have two groups of participants: group A – those who each sketched 10 shapes and group B – those who sketched more. It can be seen in Fig. 2 that sketches in group B are more accurate and have more details, and the retrieval performance on such sketches is higher than average, as shown in Table 6. The sketches of unseen participants contain moderate amount of details (Fig. 2), and the performance on those sketches is slightly below the average performance. This indicates that the methods generalize sufficiently well to unseen styles. Following these results, the retrieval accuracy for the sketches of overlapping 129 shapes by one of the participants (ID = 38) from group B is higher than for the sketches of participants from group A: $31.78\%/58.14\%/68.99\%$ versus $13.95\%/33.33\%/44.96\%$ $A@1/5/10$. These 129 sketches are not a part of a test set of 202 sketches.

5.4. 2D sketch vs. 3D sketch

We are not aware of any work on 2D sketch-based *instance-level* 3D shape retrieval. We thus adopt *category-level* (sketch/image)-based shape retrieval methods, such as [36, 28], for a fine-grained scenario. Due to a large domain gap between a 2D sketch and a 3D shape, we limit in this section the gallery size from 5794 to 202 shapes from the test set. We use the dataset of 2D human sketches done by participants without any art experience[4] on the same set of shapes as our dataset. We compare two architectures, where the 2D branch is VGG11 [64][5], and the 3D shape branch either employs (1) NGVNN [27] with multi-view shape representation (#A), or (2) PointNet++ with point cloud shape representation (#B). The training is performed with a Triplet Loss. In $\#C$ we combine information from 3 2D sketch views of the same shape with azimuth angles of 0,30 and 75^o. Finally, we compare with a 2D sketch to a 2D image fine-grained retrieval baseline [82]. In $\#D$ we use as a target a shape view (Phong shading) from the same viewpoint as a sketch. In $\#E$ we use as a target a 2D NPR rendering from the same viewpoint as a sketch. $\#C, D, E$ use encoders pretrained on ImageNet. The numerical evaluation in Table 7 demonstrates that *3D sketches can bring a new level of accuracy to instance-level retrieval.*

5.5. Discussion

Fig. 6 shows the visualization of the retrieval results. We observe that the retrieved results frequently match well the input sketches, but do not necessary match the ground-truth – due to the sketch being a distorted version of the shape. This analysis highlights the need for new structure-aware metrics, more tolerant towards distortions. Similarly, the retrieval method should be structure-aware to account for potential distortions. Nevertheless, all the 3D retrieval solutions manage to grasp the shape from a few sparse strokes,

[4] http://sketchx.ai/downloads/: AmateurSketch-3DChair

[5] VGG11 gives better results than ResNet [26] in our experiments

#	Training			Test set (202 → 5,794)			Unseen 5 participants (50 → 5,794) (U)			Group A \U (67 → 5,794)			Group B (85 → 5,794)		
	Size	Data	Method	A@1	A@5	A@10	A@1	A@5	A@10	A@1	A@5	A@10	A@1	A@5	A@10
1	702	HS ShapeNet	PointNet++ Siam. TL	26.2	43.1	54.5	24.0	38.0	44.0	11.9	22.4	35.8	38.8	62.4	75.3

Table 6. Results analysis per group.

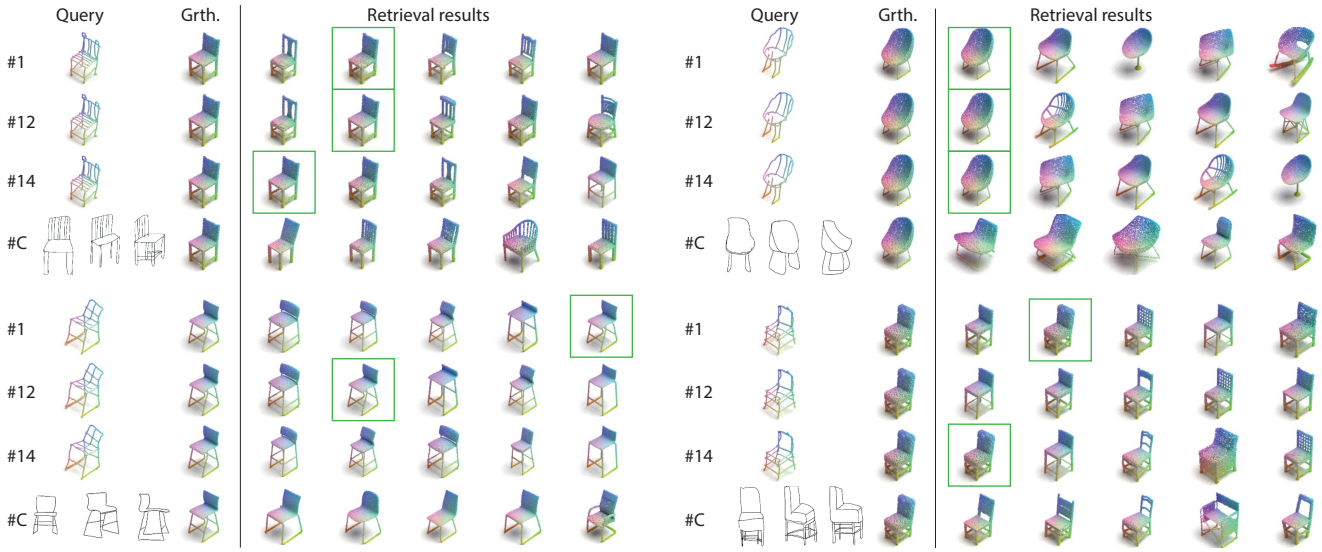


Figure 6. The ranked retrieval results from 5,794 shapes when training on 702 human sketches with the triplet loss and Siamese architecture (#1), with augmentation by synthetic sketches of additional 702 shapes (#12), with augmentation with random distortions (#14). The 3D retrieval results are compared with the retrieval results given 3 2D viewpoints (#C). For more results please refer to the supplemental. The green box shows the position of the ground-truth.

#	Method	Data	A@1	A@5	A@10
A	2d→3d: Multi-view, Heter. TL	2D HS + 3D shapes	19.8	45.05	59.41
B	2d→3d: PointNet++ TL	2D HS + 3D shapes	11.39	34.65	54.95
C	(3 views) 2d→3d: Multi-view, Heter. TL	2D HS (3 views) + 3D shapes	32.67	68.81	80.69
D	2d→2d: [82]	2D HS + 3D views	25.25	60.4	77.23
E	2d→2d: [82]	2D HS + 2D NPR	14.36	45.54	60.89
1	3d→3d: PointNet++, Siam. TL	3D HS + 3D shapes	**61.39**	**83.67**	**90.59**

Table 7. The comparison of 2D sketch-based and 3D sketch-based retrieval methods. See Sec. 5.4 for the details.

achieving notably higher accuracy than the retrieval results from the comparable 2D sketches.

6. Conclusion

We propose the first large-scale dataset of human VR sketches to fulfill a vision on the synergy between sketching and VR. With the dataset, we (i) demonstrate the advantage of 3D sketches over 2D sketches for navigating large 3D shapes collections, and (ii) analyze the role of training set size and alternative augmentation strategies.

Our experiments suggest that to better address the remaining domain gap between 3D VR sketch and 3D shapes, future work should focus on how to better account for structural information, adapting to the distortions inherent to human sketches. Such algorithms will further require deformation-aware losses. Our experiments with the contrastive loss indicate the need for light-weight encoders to provide more options for novel losses designs. Nevertheless, carefully leveraging existing tools, we demonstrate notable improvement in instance-level retrieval accuracy over the previous work [47]. Our best method in terms of Top-1 accuracy: PointNet++ encoder trained with a triplet loss on human sketches with augmentation by distortions, achieves $A@1 = 28.2$, $A@5 = 44.1$, and $A@10 = 55.0$, while the best model from [47], trained as was proposed in their work, on our data reaches just $A@1 = 2.5$, $A@5 = 5.5$, $A@10 = 8.4$. Through our experiments we conclude that (i) when collecting a fine-grained dataset of VR sketches one should aim at maximizing the diversity of shapes and styles rather than aiming at having several shapes drawn by several different participants, (ii) the synthetic sketches can be used as data augmentation when they provide additional shapes diversity and their number does not exceed the number of human sketches, and (iii) a training set of 600-700 sketches provides a good estimate of the achievable performance of the fine-grained retrieval method. Our dataset is the first step towards adoption of 3D VR tools by an average consumer. We believe it will foster research on multiple topics, such as design of novel encoder and decoder architectures, and loss designs for retrieval and reconstruction problems. Our dataset and the custom-build VR sketching interface are available at tinyurl.com/VRSketch3DV21.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In *ICML*, 2018. 3

[2] Rahul Arora, Rubaiat Habib Kazi, Fraser Anderson, Tovi Grossman, Karan Singh, and George W Fitzmaurice. Experimental evaluation of sketching on surfaces in VR. In *CHI*, 2017. 3

[3] Rahul Arora and Karan Singh. Mid-air drawing of curves on 3d surfaces in virtual reality. *ACM Trans. Graph.*, 40(3):1–17, 2021. 1

[4] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *ACM Trans. Graph.*, 37(4):1–12, 2018. 3

[5] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2CAD: learning CAD model alignment in RGB-D scans. In *Proc. of IEEE/CVF CVPR*, 2019. 3

[6] Yizhak Ben-Shabat, Michael Lindenbaum, and Anath Fischer. Nesti-Net: Normal estimation for unstructured 3D point clouds using convolutional neural networks. In *Proc. of IEEE/CVF CVPR*, 2019. 3

[7] Ayan Kumar Bhunia, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Sketch less for more: On-the-fly fine-grained sketch-based image retrieval. In *Proc. of IEEE/CVF CVPR*, 2020. 1

[8] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 3, 5

[9] John Collomosse, Tu Bui, Michael J Wilber, Chen Fang, and Hailin Jin. Sketching with style: Visual search with sketches and aesthetic context. In *Proc. of IEEE ICCV*, 2017. 2

[10] Manuel Dahnert, Angela Dai, Leonidas J Guibas, and Matthias Nießner. Joint embedding of 3D scan and CAD objects. In *Proc. of IEEE ICCV*, 2019. 3

[11] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape Completion using 3D-Encoder-Predictor CNNs and Shape Synthesis. In *Proc. of IEEE/CVF CVPR*, 2017. 3

[12] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. ScanComplete: Large-scale scene completion and semantic segmentation for 3D scans. In *Proc. of IEEE/CVF CVPR*, 2018. 3

[13] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPFNet: Global context aware local features for robust 3D point matching. In *Proc. of IEEE/CVF CVPR*, 2018. 3

[14] Anjan Dutta and Zeynep Akata. Semantically tied paired cycle consistency for any-shot sketch-based image retrieval. *IJCV*, 128(10):2684–2703, 2020. 1

[15] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Trans. Graph.*, 31(4), 2012. 2

[16] Mathias Eitz, Ronald Richter, Tamy Boubekeur, Kristian Hildebrand, and Marc Alexa. Sketch-based shape retrieval. *ACM Trans. Graph.*, 31(4):1–10, 2012. 1

[17] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3D object reconstruction from a single image. In *Proc. of IEEE/CVF CVPR*, 2017. 3

[18] Matheus Gadelha, Subhransu Maji, and Rui Wang. Shape generation using spatially partitioned point clouds. *arXiv preprint arXiv:1707.06267*, 2017. 3

[19] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3D point cloud processing. In *ECCV*, 2018. 3

[20] Daniele Giunchi, Stuart James, and Anthony Steed. 3D sketching for interactive model retrieval in virtual reality. In *Proc. of Expressive*, 2018. 1, 2

[21] Giorgio Gori, Alla Sheffer, Nicholas Vining, Enrique Rosales, Nathan Carr, and Tao Ju. FlowRep: Descriptive curve networks for free-form design shapes. *ACM Trans. Graph.*, 36(4), 2017. 6

[22] Alexander Grabner, Peter M Roth, and Vincent Lepetit. 3D pose estimation and 3D model retrieval for objects in the wild. In *Proc. of IEEE/CVF CVPR*, 2018. 2

[23] Yulia Gryaditskaya, Mark Sypesteyn, Jan Willem Hoftijzer, Sylvia Pont, Fredo Durand, and Adrien Bousseau. OpenSketch: A richly-annotated dataset of product design sketches. *ACM Trans. Graph. (Proc. of SIGGRAPH Asia)*, 38(6), 2019. 2

[24] Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J Mitra. PCPNet learning local shape properties from raw point clouds. In *CGF*, volume 37, pages 75–85, 2018. 3

[25] David Ha and Douglas Eck. A neural representation of sketch drawings. In *ICLR*, 2018. 2

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of IEEE/CVF CVPR*, 2016. 7

[27] Xinwei He, Tengteng Huang, Song Bai, and Xiang Bai. View n-gram network for 3D object retrieval. In *Proc. of IEEE ICCV*, 2019. 2, 7

[28] Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. Triplet-center loss for multi-view 3D object retrieval. In *Proc. of IEEE/CVF CVPR*, 2018. 2, 7

[29] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR, 2020. 4

[30] Jingwei Huang, Yichao Zhou, and Leonidas Guibas. ManifoldPlus: A Robust and Scalable Watertight Manifold Surface Generation Method for Triangle Soups. *arXiv preprint arXiv:2005.11621*, 2020. 6

[31] Li Jiang, Shaoshuai Shi, Xiaojuan Qi, and Jiaya Jia. GAL: Geometric adversarial loss for single-view 3D-object reconstruction. In *ECCV*, 2018. 3

[32] Shichao Jiao, Xie Han, Fengguang Xiong, Fusheng Sun, Rong Zhao, and Liqun Kuang. Cross-domain correspondence for sketch-based 3d model retrieval using convolutional neural network and manifold ranking. *IEEE Access*, 8, 2020. 1

[33] Jamil Joundi, Yannick Christiaens, Jelle Saldien, Peter Conradie, and Lien De Marez. An explorative study towards using vr sketching as a tool for ideation and prototyping in

product design. In *Proceedings of the Design Society: DESIGN Conference*, volume 1. Cambridge University Press, 2020. 1

[34] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *NIPS*, 33, 2020. 1, 2, 4, 5

[35] Hyeongju Kim, Hyeonseung Lee, Woo Hyun Kang, Joun Yeop Lee, and Nam Soo Kim. SoftFlow: Probabilistic framework for normalizing flow on manifolds. *arXiv preprint arXiv:2006.04604*, 2020. 3

[36] Tang Lee, Yen-Liang Lin, HungYueh Chiang, Ming-Wei Chiu, Winston Hsu, and Polly Huang. Cross-Domain Image-Based 3D Shape Retrieval by View Sequence Learning. In *3DV*, 2018. 7

[37] Bo Li, Yijuan Lu, Fuqing Duan, Shuilong Dong, Yachun Fan, Lu Qian, Hamid Laga, Haisheng Li, Yuxiang Li, P Lui, et al. SHREC'16 track: 3D sketch-based 3D shape retrieval. 2016. 1, 2

[38] Bo Li, Yijuan Lu, Azeem Ghumman, Bradley Strylowski, Mario Gutierrez, Safiyah Sadiq, Scott Forster, Natacha Feola, and Travis Bugerin. 3D sketch-based 3D model retrieval. In *ICMR*, 2015. 2

[39] Chun-Liang Li, Manzil Zaheer, Yang Zhang, Barnabas Poczos, and Ruslan Salakhutdinov. Point cloud GAN. *arXiv preprint arXiv:1810.05795*, 2018. 3

[40] Ruihui Li, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. PU-GAN: a point cloud upsampling adversarial network. In *Proc. of IEEE ICCV*, 2019. 3

[41] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution on x-transformed points. In *NIPS*, 2018. 3

[42] Zhaoqun Li, Cheng Xu, and Biao Leng. Angular triplet-center loss for multi-view 3D shape retrieval. In *Proc. of AAAI*, 2019. 2

[43] Zhaoqun Li, Cheng Xu, and Biao Leng. Rethinking loss design for large-scale 3D shape retrieval. In *IJCAI*, 2019. 2

[44] Anan Liu, Shu Xiang, Wenhui Li, Weizhi Nie, and Yuting Su. Cross-domain 3D model retrieval via visual domain adaptation. In *IJCAI*, 2018. 2

[45] Yu Liu, Hongyang Li, and Xiaogang Wang. Rethinking feature discrimination and polymerization for large-scale recognition. *arXiv preprint arXiv:1710.00870*, 2017. 2

[46] Dening Lu, Xuequan Lu, Yangxing Sun, and Jun Wang. Deep feature-preserving normal estimation for point cloud filtering. *Comput. Aided Des.*, page 102860, 2020. 3

[47] Ling Luo, Yulia Gryaditskaya, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. Towards 3D VR-sketch to 3D shape retrieval. In *Proc. of 3DV*, 2020. 1, 2, 3, 4, 6, 8

[48] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 4

[49] Alfred Oti and Nathan Crilly. Immersive 3d sketching tools: Implications for visual thinking and communication. *Computers & Graphics*, 94:111–123, 2020. 1

[50] Alfred Oti and Nathan Crilly. Immersive 3d sketching tools: Implications for visual thinking and communication. *Computers & Graphics*, 94, 2021. 1

[51] Kaiyue Pang, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Solving mixed-modal jigsaw puzzle for fine-grained sketch-based image retrieval. In *Proc. of IEEE/CVF CVPR*, 2020. 1

[52] Francesca Pistilli, Giulia Fracastoro, Diego Valsesia, and Enrico Magli. Point cloud normal estimation with graph-convolutional neural networks. In *ICMEW*, 2020. 3

[53] Albert Pumarola, Stefan Popov, Francesc Moreno-Noguer, and Vittorio Ferrari. C-Flow: Conditional generative flow models for images and 3D point clouds. In *Proc. of IEEE/CVF CVPR*, 2020. 3

[54] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3d classification and segmentation. In *Proc. of IEEE/CVF CVPR*, 2017. 3, 5

[55] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 1, 3, 5, 6

[56] Marie-Julie Rakotosaona, Vittorio La Barbera, Paul Guerrero, Niloy J Mitra, and Maks Ovsjanikov. PointCleanNet: Learning to denoise and remove outliers from dense point clouds. In *CGF*, volume 39, 2020. 3

[57] Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. Deep learning with sets and point clouds. *arXiv*, 2016. 3

[58] Enrique Rosales, Jafet Rodriguez, and Alla Sheffer. SurfaceBrush: from virtual reality drawings to manifold surfaces. *ACM Trans. Graph.*, 38(4), 2019. 1

[59] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Trans. Graph.*, 35(4):1–12, 2016. 2

[60] Ryan Schmidt, Azam Khan, Gord Kurtenbach, and Karan Singh. On expert performance in 3d curve-drawing tasks. In *Proc. of SBIM*, 2009. 1

[61] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proc. of IEEE/CVF CVPR*, 2015. 2

[62] Tianjia Shao, Weiwei Xu, Kangkang Yin, Jingdong Wang, Kun Zhou, and Baining Guo. Discriminative sketch-based 3D model retrieval via robust shape matching. In *CGF*, volume 30, 2011. 1

[63] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3D point cloud generative adversarial network based on tree structured graph convolutions. In *Proc. of IEEE ICCV*, 2019. 3

[64] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[65] Jifei Song, Qian Yu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In *Proc. of IEEE ICCV*, 2017. 2

[66] Yongbin Sun, Yue Wang, Ziwei Liu, Joshua Siegel, and Sanjay Sarma. PointGrow: Autoregressively learned point cloud generation with self-attention. In *WACV*, 2020. 3

[67] Mikaela Angelina Uy, Jingwei Huang, Minhyuk Sung, Tolga Birdal, and Leonidas Guibas. Deformation-aware 3d model embedding and retrieval. In *ECCV*. Springer, 2020. 3

[68] Diego Valsesia, Giulia Fracastoro, and Enrico Magli. Learning localized generative models for 3D point clouds via graph convolution. In *ICLR*, 2018. 3

[69] Fang Wang, Le Kang, and Yi Li. Sketch-based 3D shape retrieval using convolutional neural networks. In *Proc. of IEEE/CVF CVPR*, 2015. 1, 2

[70] He Wang, Zetian Jiang, Li Yi, Kaichun Mo, Hao Su, and Leonidas J Guibas. Rethinking sampling in 3D point cloud generative adversarial networks. *arXiv preprint arXiv:2006.07029*, 2020. 3

[71] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proc. of IEEE/CVF CVPR*, 2014. 1, 2, 4

[72] Xiaogang Wang, Marcelo H Ang Jr, and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *Proc. of IEEE/CVF CVPR*, 2020. 3

[73] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.*, 38(5), 2019. 1, 3, 5

[74] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016. 2

[75] Baoxuan Xu, William Chang, Alla Sheffer, Adrien Bousseau, James McCrae, and Karan Singh. True2form: 3d curve networks from 2d sketches via selective regularization. *ACM Trans. Graph.*, 2014. 1

[76] Yongzhe Xu, Jiangchuan Hu, Kun Zeng, and Yongyi Gong. Sketch-based shape retrieval via multi-view attention and generalized similarity. In *ICDH*. IEEE, 2018. 1

[77] Eun Kyoung Yang and Jee Hyun Lee. Cognitive impact of virtual reality sketching on designers' concept generation. *Digital Creativity*, 2020. 1

[78] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. PointFlow: 3D point cloud generation with continuous normalizing flows. In *Proc. of IEEE ICCV*, 2019. 3

[79] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. FoldingNet: Point cloud auto-encoder via deep grid deformation. In *Proc. of IEEE/CVF CVPR*, 2018. 3

[80] Yuxiang Ye, Bo Li, and Yijuan Lu. 3D sketch-based 3D model retrieval with convolutional neural network. In *ICPR*, 2016. 1, 2

[81] Emilie Yu, Rahul Arora, Tibor Stanko, J Andreas Bærentzen, Karan Singh, and Adrien Bousseau. Cassie: Curve and surface sketching in immersive environments. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2021. 1

[82] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen-Change Loy. Sketch me that shoe. In *Proc. of IEEE/CVF CVPR*, 2016. 1, 2, 7, 8

[83] Qian Yu, Jifei Song, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Fine-grained instance-level sketch-based image retrieval. *IJCV*, pages 1–17, 2020. 1, 2

[84] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. PCN: Point Completion Network. In *Proc. of 3DV*, 2018. 3

[85] Yue Zhong, Yulia Gryaditskaya, Honggang Zhang, and Yi-Zhe Song. Deep sketch-based modeling: Tips and tricks. In *Proc. of 3DV*, 2020. 1

[86] Yue Zhong, Yonggang Qi, Yulia Gryaditskaya, Honggang Zhang, and Yi-Zhe Song. Towards practical sketch-based 3d shape generation: The role of professional sketches. *IEEE TCSVT*, 2020. 1, 2

[87] Fan Zhu, Jin Xie, and Yi Fang. Learning cross-domain neural networks for sketch-based 3d shape retrieval. In *Proc. of AAAI*, 2016. 1

[88] Changqing Zou, Qian Yu, Ruofei Du, Haoran Mo, Yi-Zhe Song, Tao Xiang, Chengying Gao, Baoquan Chen, and Hao Zhang. SketchyScene: Richly-annotated scene sketches. In *ECCV*, 2018. 2