3D Shape Reconstruction from Free-Hand Sketches

Jiayun Wang[1,2] **Jierui Lin**[1] **Qian Yu**[3,1,2] **Runtao Liu**[4,2] **Yubei Chen**[1] **Stella X. Yu**[1,2]
[1] University of California, Berkeley [2] International Computer Science Institute
[3] Beihang University [4] Peking University
{peterwg,jerrylin0928,runtao_liu,yubeic,stellayu}@berkeley.edu qianyu@buaa.edu.cn

Abstract

Sketches are the most abstract 2D representations of real-world objects. Although a sketch usually has geometrical distortion and lacks visual cues, humans can effortlessly envision a 3D object from it. This indicates that sketches encode the appropriate information to recover 3D shapes. Although great progress has been achieved in 3D reconstruction from distortion-free line drawings, such as CAD and edge maps, little effort has been made to reconstruct 3D shapes from free-hand sketches. We pioneer to study this task and aim to enhance the power of sketches in 3D-related applications such as interactive design and VR/AR games. Further, we propose an end-to-end sketch-based 3D reconstruction framework. Instead of well-used edge maps, synthesized sketches are adopted as training data. Additionally, we propose a sketch standardization module to handle different sketch styles and distortions. With extensive experiments, we demonstrate the effectiveness of our model and its strong generalizability to various free-hand sketches.

1 Introduction

Human free-hand sketches are the most abstract 2D representations of 3D visual perception. Although a sketch may consist of only a few colorless strokes and exhibit various deformation and abstractions, humans can effortlessly envision the corresponding real-world 3D object from it. A computer vision model that can replicate this ability is still missing. Although sketches and 3D representations have drawn great interest from researchers in recent years, these two modalities have been studied relatively independently. We pioneer to explore the possibility to bridge the gap between sketches and 3D representations and build a computer vision model to recover 3D shapes from sketches. Such a model will unleash many applications such as interactive CAD design and VR/AR games.

With the development of new devices and sensors, sketches and 3D shapes, as representations of real-world objects beyond natural images, become increasingly important. The popularity of touch-screen devices makes sketching not a privilege of professionals anymore and increasingly popular. Researchers have applied sketch in tasks like image retrieval [3, 4] and image synthesis [5, 6] to leverage its power in expression. Furthermore, as depth sensors, such as structure light device, LiDAR, and TOF cameras, become more ubiquitous, 3D shapes become an emerging modality in computer vision. 3D reconstruction from multi-view images has been studied for many years [7, 8, 9]. Recent works [10, 11, 12] have further explored reconstructing a 3D model from a single image.

Despite these trends and progress, works connecting 3D and sketches are quite rare. We argue that sketches are abstract 2D representations of 3D perception, and it is of great importance to study sketches in a 3D-aware perspective and build connections between two modalities. In computer graphics, researchers have explored the potential of sketching for 3D modeling, e.g. True2Form and BendSketch [13, 14] (Fig.1(**L**)). These works are based on *distortion-free* line drawings produced by professionals. Furthermore, the role of line drawings in such works is to provide geometrical information for the subsequent 3D modeling. Delanoy *et al.* [1] first employ neural networks to

Preprint. Under review.

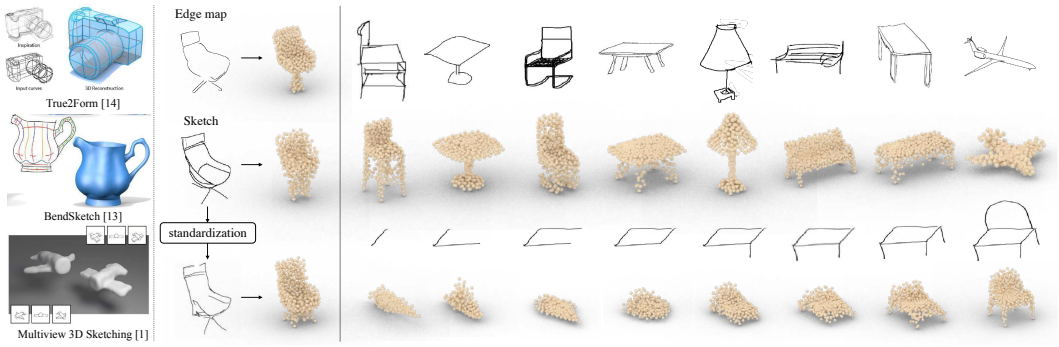

Figure 1: Sketching is a tool by *professionals* for 3D reconstruction. We relax rigid sketching constraints and reconstruct 3D shapes from single-view *free-hand* sketches (**Left**). While previous works [1, 2] employ edge-maps as proxies for sketches, we show that training with synthesized sketches as inputs generalizes better to free-hand sketches. We also show that the proposed sketch standardization module can deal with different sketching styles as well as distortions (**Middle**). Our model reconstructs 3D shapes from various free-hand sketches and may unleash many practical applications such as real-time 3D modeling with sketches (**Right**).

reconstruct 3D shapes directly from line drawings. However, the method's great results come with two major constraints: 1) it works on edge maps, i.e., a type of distortion-free line drawings; 2) it requires inputs depicting the object from multi-views to achieve satisfactory outcomes. Therefore, [1] cannot handle free-hand sketch well as we show later in the experiment. Other works such as [2, 15] tackle the task of 3D retrieval instead of 3D shape reconstruction from sketches. Overall, the research on the task of reconstructing a 3D shape from a single *free-hand* sketch is still left unexplored.

In this work, we explore free-hand sketch-based 3D reconstruction for the first time (Fig.1(**M**)). The task is challenging due to following reasons: **1**) A sketch presents the structure and the surface of an object by combinations of line strokes. However, a 3D shape has one more dimension, which encodes the depth information. Therefore, a 3D shape has richer information and to reconstruct a 3D shape from a sketch is an ill-posed problem. **2**) There is a misalignment between the two representations. A sketch depicts an object from a certain view while a 3D shape can be viewed from multiple angles due to the encoded depth information. Besides, due to the nature of hand drawing, a sketch is usually geometrically imprecise compared to the real object. Thus a sketch can only provide suggestive shape and structural information. In contrast, a 3D shape is faithful to its corresponding real-world object with no geometric deformation. **3)** Paired sketch-3D datasets are rare although there exist several large-scale sketch datasets and 3D shape datasets respectively. Furthermore, collecting sketch-3D pairs can be much more time-consuming and expensive than collecting sketch-image pairs as each 3D shape can be sketched from many different viewing angles.

As a starting point to this task, we propose a single-view sketch-to-3D shape reconstruction framework. Specifically, it is an end-to-end deep neural network, which takes a sketch image from an *arbitrary* angle as input and reconstructs a 3D shape point cloud. Our model cascades a sketch standardization module U and a reconstruction module G. U handles various drawing styles/distortions and transfers inputs to standardized sketches while G takes a standardized sketch to reconstruct the 3D shape (point cloud). Furthermore, considering the paired sketch-3D data is limited, we use cycleGAN to generate sketch-3D pairs automatically from 2D renderings of 3D shapes [5]. Together with the standardization module U, the synthesized sketches provide sufficient information to train the model. We conduct extensive experiments on a composed sketch-3D dataset, spanning 13 classes, where sketches are synthesized and 3D objects come from ShapeNet dataset [16]. Furthermore, we collect an evaluation set, which consists of 390 real sketch-3D pairs. The results show that the proposed model can reconstruct 3D shapes with fine-grained details from real sketches under different styles, stroke line-widths, and object categories. Our model also enables practical applications such as real-time 3D modelling with sketches (Fig.1(**R**)).

To summarize, our work makes following contributions: **1)** We pioneer to study the possibility of reconstructing 3D shapes from single-view free-hand sketches and the proposed model demonstrates its robust performance on real sketches; **2)** To deal with the data insufficiency issue, we provide a GAN-based method to generate synthetic sketches for 3D shapes. **3)** The sketch standardization module designed in this work largely improves the generalizability of our sketch-3D model. It provides a practical solution to handle the variation and distortion in real sketches.

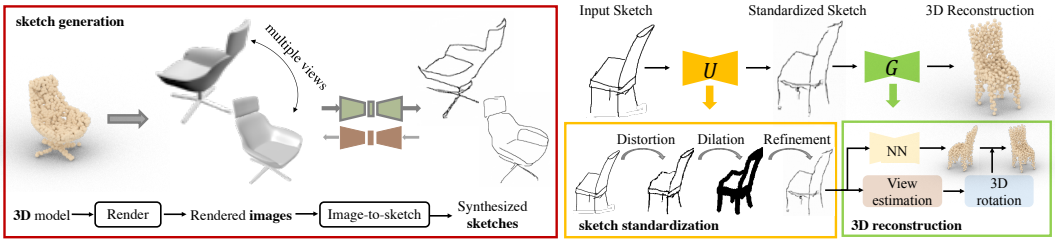

Figure 2: The pipeline of the proposed model. The model consists of three major components, sketch generation (red box), sketch standardization (orange box), and 3D reconstruction (green box). To generate sketches automatically for training the model, we first render 2D images for a 3D shape from multiple viewpoints, and then employ an off-the-shelf sketch-image translation model to generate sketches with corresponding views. The standardization module is introduced to handle sketches with different styles and distortions. In the 3D reconstruction network, a view estimation module is adopted to align the view of the output and the corresponding ground-truth 3D shape.

2 Related Works

3D Reconstruction from Images. SfM [9] and SLAM [7] achieve success in handling multi-view 3D reconstructions in various real-world scenarios. Their reconstructions can be limited by insufficient input viewpoints and 3D scanning data. Deep learning based methods have been proposed to further improve reconstructions by completing 3D shapes with occluded or hollowed-out areas [17, 8, 18].

In general, recovering the 3D shape from a single-view image is an ill-posed problem. Attempts to tackle the problem include 3D shape reconstructions from silhouettes [11], shading [10], and texture [19]. However, these methods need strong presumptions and expertise in natural images [20], which limit their usage in the real-world scenarios. Generative adversarial networks (GANs) [21] and variational autoencoders (VAEs) [22] have achieved success in image synthesis and enable [23] 3D shape reconstruction from a single-view image. Fan *et al.* [12] further adopt point clouds as 3D shape representation, which enables models to reconstruct fine-grained details from a single-view image.

3D reconstruction networks are designed differently depending on the output 3D representation. 3D voxel reconstruction networks [24, 25, 26] benefit from many image processing networks as convolutions are appropriate for voxels. They are usually constrained to low resolution due to the computational overhead. Mesh reconstruction networks [27, 28] are able to directly learn from meshes, where they suffer from topology issues and heavy computation [29]. We adopt point cloud representation as it can capture fine-grained 3D geometric details with low computational overhead. Reconstructing 3D point clouds from images has been shown to benefit from well-designed network architectures [12, 30], latent embedding matching [31], additional image supervision [32], etc.

Sketch-based 3D Retrievals/Reconstructions. Free-hand sketches are used for 3D shape retrieval [2, 15] given their power in expression. However, retrieval methods are significantly constrained by the gallery dataset. Precise sketching is also studied in the computer graphics community for 3D shape modeling or procedural modeling [13, 33]. These works are designed for professionals and require additional information for shape modeling, e.g., surface-normal, procedural model parameters.

Delanoy *et al.* [1] first employ neural networks to learn 3D voxels from line-drawings. While it is successful, this model has several limitations. 1) The model input is the edge map. 2) The model requires multiple inputs from different viewpoints for a satisfactory result. These limitations prevent the model from generalizing to real free-hand sketches. Unlike the existing works, the proposed method in this work reconstructs the 3D shape based on a single free-hand sketch. We believe our model may make 3D reconstruction and its applications more accessible to amateurs.

3 3D Reconstruction from Sketches

The proposed framework is composed of three modules (as shown in Fig.2). In order to deal with the data insufficiency issue, we first synthesize sketches for 3D shapes as the training set (Sec. 3.1). The module U transfers an input sketch to a standardized sketch (Sec. 3.2). After that, the module G takes the standardized sketch to reconstruct a 3D shape (point cloud) (Sec. 3.3). In Sec. 3.4, we present details of a new sketch-3D dataset, which is collected for evaluating the proposed model.

3.1 Synthetic Sketch Generation

As mentioned earlier, to the best of our knowledge, there exists no paired sketch-3D dataset. While it is possible to resort to edge maps as in [1], edge maps are different from sketches (as shown in the 3rd and 4th rows of Fig.3). We show that the reconstruction model trained on edge maps cannot generalize well to real free-hand sketches in Sec. 4.3. Thus it is crucial to find an efficient and reliable way to synthesize sketches for 3D shapes. Following [5], we employ a generative model to synthesize sketches from rendered images of 3D shapes. Fig.3 depicts the procedure. Specifically, we first render m images for each 3D shape, where each image corresponds to a certain view of a 3D shape. We then adopt the model introduced in [5] to synthesize sketches $\{S_i\}$ as our training data.

3.2 Sketch Standardization

Considering various sketching styles and geometric distortions of different individuals, as well as to enhance the generalizability and robustness of the proposed model, we transfer an input sketch S_i to a standardized style before 3D reconstructions. Specifically, U first applies local and structural deformation D_1 [34] to an input sketch S_i. Then, a dilation operator D_2 [35] is applied to $D_1(S_i)$, which is followed by a refinement operator R to transfer to the standardized style $\widetilde{S}_i$. R is implemented as an image translation network [36]. Together, the standardization module, $U = R \circ D_2 \circ D_1$, mimics the variation introduced in the human sketching process and standardizes input sketches $\{S_i\}$ into a standardized style $\{\widetilde{S}_i\}$. We demonstrate the standardization process in Fig.2.

3.3 Sketch-3D Reconstruction

Our 3D reconstruction network G (Fig.2) consists of several components. Given a standardized sketch $\widetilde{S}_i$, the view estimation module first estimates its viewpoint. $\widetilde{S}_i$ is then fed to the sketch-to-3D module to generate a 3D point cloud $P_{i,pre}$, which aligns with the sketch viewpoint. A 3D rotation corresponding to the viewpoint is then applied to $P_{i,pre}$ to output the canonically-posed point cloud P_i. The objective of G is to minimize the distance between P_i and the ground truth $P_{i,gt}$ pairs.

View Estimation Module. The view estimation module g_1 aims to determine the three-dimensional pose from an input sketch $\widetilde{S}$. Similar to the input transformation module of the PointNet [37], g_1 estimates a 3D rotation matrix A from a sketch $\widetilde{S}$, i.e. $A = g_1(\widetilde{S})$. A regularization loss $L_{\text{orth}} = \|I - AA^T\|_F^2$ is applied to ensure A is a rotation (orthogonal) matrix. The rotation matrix A rotates a point cloud from the viewpoint pose to a canonical pose, which matches the ground truth.

3D Reconstruction Module. The module g_2 is adapted from the Point-Set-Generation network [12]. The reconstruction network g_2 learns to reconstruct a 3D point cloud P_{pre} from a sketch $\widetilde{S}$, i.e. $P_{pre} = g_2(\widetilde{S})$. P_{pre} is further transformed by the corresponding rotation matrix A to P so that P aligns with the ground-truth 3D point cloud P_{gt}'s canonical pose. Overall, we have $P = g_1(\widetilde{S}) \cdot g_2(\widetilde{S})$. To train G, we penalize the distance between an output point cloud P and the ground-truth point cloud P_{gt}. We employ the Chamfer distance (CD) between $P, P_{gt} \subset \mathbb{R}^3$:

$$d_{CD}(P\|P_{gt}) = \sum_{\mathbf{p}\in P} \min_{\mathbf{q}\in P_{gt}} \|\mathbf{p}-\mathbf{q}\|_2^2 + \sum_{\mathbf{q}\in P_{gt}} \min_{\mathbf{p}\in P} \|\mathbf{p}-\mathbf{q}\|_2^2 \tag{1}$$

The final loss of the entire network is:

$$L = \sum_i d_{CD}\left(G \circ U(S_i)\|P_{i,gt}\right) + \lambda L_{\text{orth}} \tag{2}$$

$$= \sum_i d_{CD}\left(A_i \cdot P_{i,pre}\|P_{i,gt}\right) + \lambda L_{\text{orth}} \tag{3}$$

$$= \sum_i d_{CD}\left(g1(\widetilde{S}_i) \cdot g_2(\widetilde{S}_i)\|P_{i,gt}\right) + \lambda L_{\text{orth}} \tag{4}$$

where λ is the weight of the orthogonal regularization loss and $\widetilde{S}_i = R \circ D_2 \circ D_1(S_i)$ is the standardized sketch from S_i. Note that we employ CD rather than EMD (Section 4.1) to penalize the difference between the reconstruction and the ground-truth point clouds because CD emphasizes the geometric outline of point clouds and leads to reconstructions with better geometric details. EMD, however,

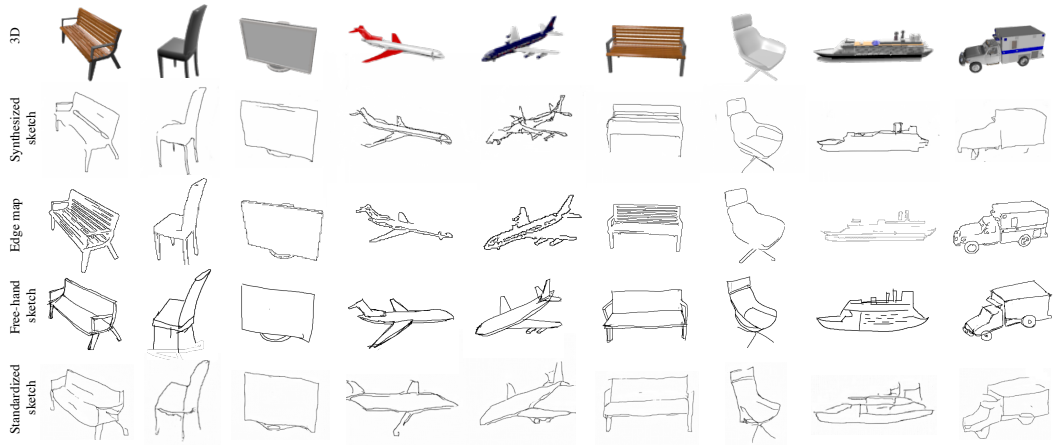

Figure 3: Comparison between the synthesized sketches (2nd row) and edge maps (3rd row). The 4th row shows the newly collected free-hand sketches, which are for evaluation. The last row shows their corresponding sketches after standardization. We observe that: **1)** The synthesized sketches are visually more similar to free-hand sketches than edge maps as they contain distortions and emphasize perceptually significant contours; **2)** After the standardization, the processed free-hand sketches have a uniform style.

emphasizes the point cloud distribution and may not preserve the geometric details well at locations with low point density.

3.4 3D Sketching Dataset

To evaluate the performance of our method, we collected a real-world evaluation set containing paired sketch-3D data. Specifically, we randomly choose ten 3D shapes from each of the 13 categories of the ShapeNet dataset [16]. Then we randomly render 3 images from different viewpoints for each 3D shape. Totally, there are 130 different 3D shapes and 390 rendered images. We recruited 11 volunteers to draw the sketches for the rendered images. Several experienced researchers reviewed the final sketches for the quality control purpose. We present several examples in Fig.3.

4 Experimental Results

We first present the datasets, evaluation metrics and implementation details, followed by qualitative and quantitative results of our model. Along the way, we provide comparisons with some state-of-the-art methods. We also conduct ablation studies to understand the benefits of different modules.

4.1 Datasets and Evaluation Metrics

The proposed model is trained on a subset of ShapeNet [16] dataset, following settings of [26]. The dataset consists of 43,783 3D shapes spanning 13 categories, including car, chair, table, etc. For each category, we randomly select 80% 3D shapes for training and 20% for evaluation. As mentioned in Section 3.1, corresponding sketches of rendered images of each 3D shape of ShapeNet are synthesized with our synthetic sketch generation module (Section 3.1).

To evaluate the proposed method's 3D reconstruction performance on real free-hand sketches, we use our proposed sketch-3D datasets (Section 3.4). To evaluate the generalizability of our model, we also evaluate on three free-hand sketch datasets, including the Sketchy dataset [4], the TU-Berlin dataset [38], and the QuickDraw dataset [39]. For these additional datasets, only sketches from categories that overlap with the ShapeNet dataset are considered.

Following the previous works on point cloud generation [12, 31, 40], we adopt two evaluation metrics to measure the similarity between the reconstructed 3D point cloud P and the ground-truth point cloud P_{gt}. The first one is the Chamfer distance (Eqn.1), and another one is the Earth Mover's distance (EMD): $d_{EMD}(P, P_{gt}) = \min_{\phi:P\mapsto P_{gt}} \sum_{x\in P} \|x - \phi(x)\|$, where P, P_{gt} has the same size $|P| = |P_{gt}|$ and $\phi : P \mapsto P_{gt}$ is a bijection. CD and EMD evaluate the similarity between two point clouds from two different perspectives (more details can be found in [12]).

4.2 Implementation Details

Sketch Generation. We utilize an off-the-shelf sketch-image translation model [5] to synthesize sketches for training. Similar to CycleGAN [41], it is trained on unpaired data [3]. The model is designed to synthesize a shoe photo given a sketch, but it can also inversely generate sketch-like contours based on a photo. We directly use the model without any fine-tuning.

Data Augmentation. During training, to further improve the generalizability and robustness of the model, we perform data augmentations for synthetic sketches before feeding them to the standardization module. Specifically, we apply image spatial translation (up to ±10 pixels) and rotation (up to ±10°) on each input sketch.

Sketch Standardization. Each input sketch S_i is first randomly deformed with moving least squares [42] both globally and locally (D_1), and then binarized and dilated 5 times iteratively (D_2) to obtain a rough sketch S_r. The rough sketch S_r is then used to train a Pix2Pix model [36], R, to reconstruct the input sketch S_i. The network is trained for 100 epochs with initial learning rate 2e-4. Adam optimizer [43] is used for the parameter optimization.

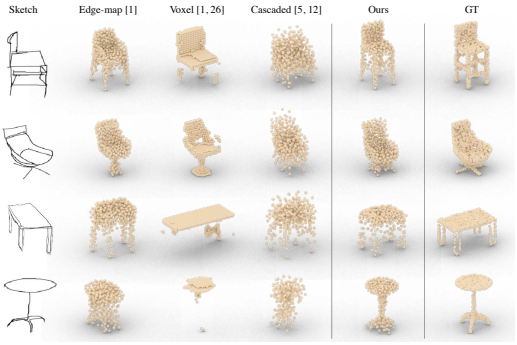

Figure 4: Performance on free-hand sketches with different design choices. We compare the performances on free-hand sketches from models based on different design choices. The design pool includes the model trained on edge maps (the 2nd column), the model whose 3D output is represented by voxel (the 3rd column), and the model with a cascaded two-stage structure (the 4th column), and our proposed model (the 5th column). Overall, the proposed method achieves better performance and keeps more fine-grained details, e.g. the legs of chairs.

3D Reconstruction. The 3D reconstruction network follows [12]'s framework with hourglass network architecture [44]. We compare several different network architectures (simple encoder-decoder architecture, two-prediction-branch architecture, etc.) and find that hourglass network architecture gives the best performance. This may be due to its ability to extract key points from images [44, 45]. We train the network for 260 epochs with an initial learning rate of 3e-5. The weight of the additional orthogonal loss is 1e-3. To enhance the performance on every category, all categories of 3D shapes are trained together. The class-aware mini-batch sampling [46] is adopted to ensure a balanced category-wise distribution for each mini-batch. We choose Adam optimizer [43] for the parameter optimization. 3D point clouds are visualized with the renderer [47].

4.3 Results and Comparisons

We first present our model's 3D shape reconstruction performance, along with the comparisons with various baseline methods. Then we present the results on sketches from different viewpoints and of different categories, as well the results on other free-hand sketch datasets. Note that unless specifically mentioned, all evaluations are on the free-hand sketches rather than synthesized sketches.

Baseline Methods. Our 3D reconstruction network is a one-stage model, where the input sketch is treated as an image and the output 3D shape is represented by point clouds. We compare with different variants to demonstrate the effectiveness of each design choice. **Sketch: point-based vs. image-based.** Considering a sketch is relatively sparse in pixel space and consists of colorless strokes, we can employ 2D points cloud to represent a sketch. Specifically, 512 points are randomly sampled from strokes of each binarized sketch, and we use a point-to-point network architecture (adapted from PointNet [37]) to reconstruct 3D shapes from the 2D point clouds. **Sketch: Using edge maps as proxy.** We use edge maps extracted by Canny edge detector as an alternative to synthesize sketches, following [1]. As edge maps are easier to obtain, the comparison helps us understand if our proposed synthesizing method is necessary. **3D shape: voxel vs. point cloud.** In this variant, we represent a 3D shape with voxels, following the settings in [1]. As the voxel representation is adopted from the previous method, the comparison helps to understand if representing 3D shapes with point clouds has benefits. **Model design: end-to-end vs. two-stage.** Although the task of reconstructing 3D shapes from free-hand sketches is new, sketch-to-image synthesize and 3D shape reconstruction from

Table 1: 3D shape reconstruction performance of different methods. Lower is better.

category	CD ($\times 10^{-4}$)							EMD ($\times 10^{-2}$)						
	points	edge [1]	voxel [1, 26]	cas. [5, 12]	w/o stand.	w/o v. est.	ours	points	edge [1]	voxel [1, 26]	cas. [5, 12]	w/o stand.	w/o v. est.	ours
airplane	11.4	7.8	35.1	71.7	8.1	7.6	**6.1**	8.5	7.3	10.8	12.7	10.2	8.2	**6.5**
bench	29.2	16.7	202.8	414.1	18.6	18.6	**13.0**	11.1	8.7	22.0	25.8	15.9	11.1	**7.8**
cabinet	61.7	50.4	59.1	354.5	**35.8**	43.5	39.2	17.6	17.8	17.0	29.6	19.7	17.7	**16.0**
car	20.8	13.3	173.2	114.2	15.2	**8.9**	10.4	**8.9**	20.0	25.2	20.0	13.1	15.4	18.0
chair	41.8	36.4	108.6	237.1	29.0	31.6	**26.9**	15.1	15.6	19.4	22.8	18.0	15.3	**13.0**
display	68.6	48.3	33.1	340.2	**32.8**	40.7	37.7	15.5	15.1	**13.1**	27.9	16.8	15.5	14.4
lamp	63.3	59.4	107.0	214.0	**37.9**	48.8	46.3	21.3	22.6	21.2	24.9	22.6	21.5	**20.4**
speaker	88.2	79.7	203.2	406.4	**60.3**	67.0	62.1	19.4	19.2	23.8	28.0	20.8	19.3	**17.9**
rifle	17.0	12.1	170.1	15.4	12.7	**8.8**	10.1	11.2	13.8	23.7	15.4	**9.4**	10.8	12.4
sofa	32.8	20.9	141.2	482.4	22.9	23.4	**16.3**	11.1	8.5	18.6	25.4	15.8	11.0	**7.7**
table	55.2	49.4	134.7	469.5	**36.7**	45.3	40.7	19.1	17.7	18.5	26.5	21.4	19.2	**17.3**
telephone	30.7	27.3	26.9	259.8	**18.7**	23.3	21.3	13.4	13.6	15.1	27.2	14.7	13.4	**12.3**
watercraft	32.9	26.0	129.1	53.8	**19.5**	24.2	20.3	12.5	11.1	23.1	17.8	15.1	12.6	**10.6**
avg.	42.6	34.4	117.2	264.1	**26.8**	30.1	26.9	14.2	14.7	19.3	23.4	16.4	14.7	**13.4**
f.h. sketch	87.1	89.0	162.5	334.2	92.6	86.8	**86.1**	18.6	16.4	22.9	26.1	18.2	16.2	**16.0**

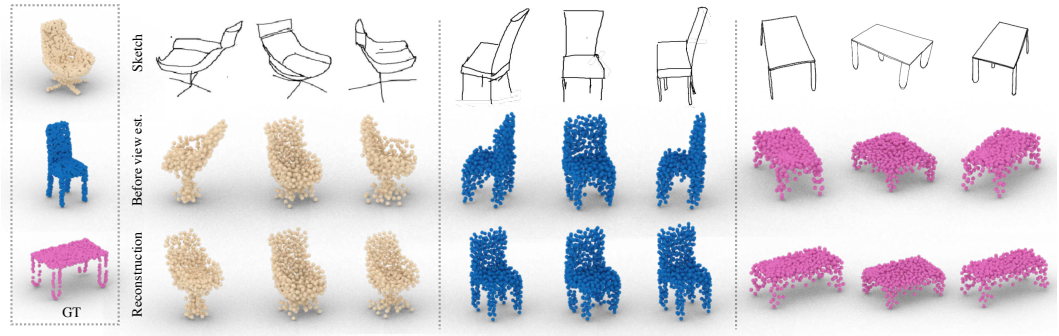

Figure 5: 3D reconstructions of sketches from different viewpoints. Before the view estimation module, the reconstructed 3D shape aligns with the input sketch's viewpoint. This module transforms the pose of the output 3D shape to align with that of the ground-truth 3D shape. The model can reconstruct a 3D object from different views when certain parts is occluded (e.g. legs of the table), with slight detail variations for different views.

images have been studied before [5, 26, 12]. Is a straight combination of the two models, instead of an end-to-end model, enough to perform well for the task? To compare the performance of these two architectures, We implement a cascaded model by composing a sketch-to-image model [5] and an image-to-3D model [12] to reconstruct 3D shapes. Further details of these baseline methods are available in the supplementary material.

Comparison and Results. Table 1 and Fig.4 present the quantitative and qualitative results of our method and different design variants. Specifically for quantitative comparisons (Table 1), we report 3D shape reconstruction performance on both synthesized (evaluation set) and free-hand sketches. This is due to that the collected free-hand sketch dataset is relatively small and together they provide a more comprehensive evaluation. We have the following observations: **1)** Representing sketches as images outperforms representing them as 2D point clouds (points vs. ours). **2)** The model trained on synthesized sketches performs better on real free-hand sketches than the model trained on edge maps (89.0 vs. 86.1 on CD, 16.4 vs. 16.0 on EMD). **3)** In terms of the model design, the end-to-end model outperforms the two-stage model by a large margin (cas. vs. ours). **4)** In terms of the 3D shape representation, while the voxel representation can reconstruct the general shape well, the fine-grained details are frequently missing due to its low resolution ($32 \times 32 \times 32$). Thus point clouds outperform voxels. Note that the resolution can hardly improve much due to the complexity and computational overhead. Voxels are converted to point clouds by balanced sampling points on its surface.

Reconstruction with Different Views and Categories. Fig.5 depicts 3D reconstructions with sketches from different viewpoints. Our model can reconstruct 3D shapes from different views even certain part is occluded (e.g. legs of the table). Slight variations in details exist for different views.

Fig.6 shows 3D reconstruction results with sketches from different object categories. Our model can reconstruct 3D shapes of multiple categories unconditionally. There are some failure cases that the model may not handle detailed structures well, recognized the wrong category (display as lamp) due to the ambiguity of the sketch, or not able to generate accurate 3D shapes from sketches with little geometric information.

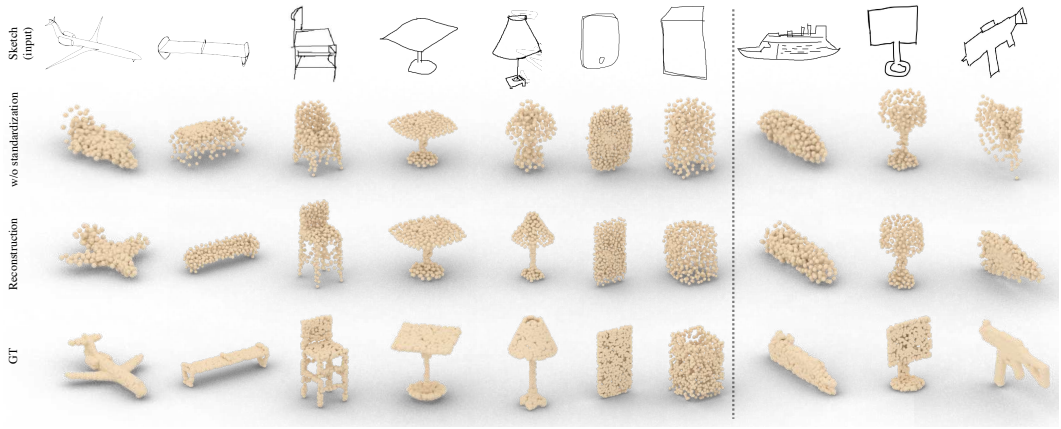

Figure 6: 3D reconstructions on our proposed evaluation dataset. **Left:** Examples of some good reconstruction results. **Right:** Examples of failure cases. It seems that our model may not handle detailed structures well (e.g., *watercraft*), recognize the wrong category (e.g., *display* as *lamp*) due to ambiguity of the sketch, as well as not able to generate 3D shape from very abstract sketches where few geometric information is available (e.g., rifle).

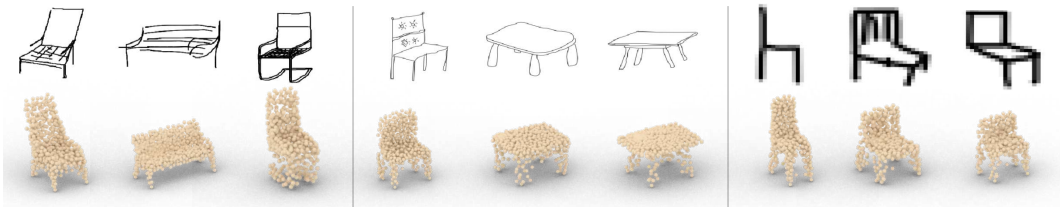

Figure 7: Evaluation on other sketch datasets. **Left**: Sketchy dataset [4]; **Middle**: TU-Berlin dataset [38]; **Right**: QuickDraw dataset [39]. Our model is able to reconstruct 3D shapes from sketches with different styles and line-widths, and even low-resolution data.

Evaluation on other Sketch Datasets. To evaluate the generalizability of the proposed method, we also evaluate on three other free-hand sketch datasets [4, 38, 39]. We only present some qualitative results (Fig.7) as the ground-truth 3D shapes are not available. Our model is able to reconstruct 3D shapes from sketches with different styles, line-widths, levels of distortions even at low resolution.

4.4 Ablation Studies

Sketch Standardization Module. Quantitative (Table 1) and qualitative results (Fig.6) are presented. While removing sketch standardization does not affect the 3D reconstruction performance from the synthesized sketches much, the performance on real sketches has a significant drop without the standardization module. The module can be considered as a domain adaption module designed for sketches, and empirically we find it important for 3D shape reconstruction from real sketches.

View Estimation Module. In terms of the quantitative results (Table 1), removing the view estimation module leads to a performance drop of both CD and EMD. This indicates an explicit view estimation is helpful for reconstructing the 3D shape more faithfully. We also demonstrate the function of the view estimation module in Fig.5. Before the 3D rotation, the reconstructed 3D shape has the pose aligned with the input sketch. After the 3D rotation based on the estimated viewpoint, the 3D shape is aligned to the ground truth's canonical pose.

5 Summary

To the best of our knowledge, we are the first to study reconstructing 3D shapes from single-view free-hand sketches. We propose using synthesized sketches for training and introduce a standardization module to handle the data insufficiency problem and style variations of sketches. Our proposed model proves to be able to successfully reconstruct 3D shapes from free-hand sketches of different views and categories. We hope this work unleashes more potentials of the sketch in applications such as sketch-based 3D design/games, making them more accessible to the general public.

Broader Impact

The proposed model can reconstruct 3D shapes from free-hand sketches instead of distortion-free line-drawings, which require high proficiency and accuracy. The relaxing of the input makes 3D shape modeling more accessible to the general public and may unleash many practical applications such as product design, 3D printing, and VR/AR games. The research direction in this work may also help making traditional CAD tools more intuitive and interactive, reducing the professionals' burden in their design process.

Further, this work help to bridge the gap between abstract sketch and 3D shape representations. As we mentioned in the introduction that humans could effortlessly recover a real-world 3D shape from a abstractly drawn free-hand sketch. A scientific understanding of this process is still missing. We believe that building a practical model is an important step towards a deeper understanding. In this work, We develop an end-to-end neural network which enables machines to perform the task. Although there is a significant difference between human brains and a deep neural network, understanding how a model achieves the goal may inspire research on unveiling the mechanism of humans' 3D reconstruction ability.

Acknowledgement

We thank Charles R. Qi and Kaichun Mo for the help with the point cloud renderer, as well as Yifei Xing, Haoran Liao, Suifang Mao, Mengyang Zhang, Jiazheng Zhao, Ziyun Zhao, Ruiqing Xiang, Yu Wang, Yan He and Mengyao Lu for providing free-hand sketches of 3D objects.

References

[1] Johanna Delanoy, Mathieu Aubry, Phillip Isola, Alexei A Efros, and Adrien Bousseau. 3d sketching using multi-view deep volumetric prediction. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 1(1):1–22, 2018.

[2] Fang Wang, Le Kang, and Yi Li. Sketch-based 3d shape retrieval using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[3] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen-Change Loy. Sketch me that shoe. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

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

[5] Runtao Liu, Qian Yu, and Stella Yu. An unpaired sketch-to-photo translation model. *arXiv preprint arXiv:1909.08313*, 2019.

[6] Arnab Ghosh, Richard Zhang, Puneet K Dokania, Oliver Wang, Alexei A Efros, Philip HS Torr, and Eli Shechtman. Interactive sketch & fill: Multiclass sketch-to-image translation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[7] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial intelligence review*, 43(1):55–81, 2015.

[8] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*. Springer, 2016.

[9] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion*. *Acta Numerica*, 26:305–364, 2017.

[10] Stephan R Richter and Stefan Roth. Discriminative shape from shading in uncalibrated illumination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[11] Endri Dibra, Himanshu Jain, Cengiz Oztireli, Remo Ziegler, and Markus Gross. Human shape from silhouettes using generative hks descriptors and cross-modal neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[12] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[13] Changjian Li, Hao Pan, Yang Liu, Xin Tong, Alla Sheffer, and Wenping Wang. Bendsketch: modeling freeform surfaces through 2d sketching. *ACM Transactions on Graphics*, 36(4):1–14, 2017.

[14] Baoxuan Xu, William Chang, Alla Sheffer, Adrien Bousseau, James McCrae, and Karan Singh. True2form: 3d curve networks from 2d sketches via selective regularization. *Transactions on Graphics*, 33(4), 2014.

[15] Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. Triplet-center loss for multi-view 3d object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[16] 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.

[17] Bo Yang, Stefano Rosa, Andrew Markham, Niki Trigoni, and Hongkai Wen. Dense 3d object reconstruction from a single depth view. *IEEE transactions on pattern analysis and machine intelligence*, 41(12):2820–2834, 2018.

[18] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Advances in neural information processing systems*, 2017.

[19] Andrew P Witkin. Recovering surface shape and orientation from texture. *Artificial intelligence*, 17(1-3):17–45, 1981.

[20] Yang Zhang, Zhen Liu, Tianpeng Liu, Bo Peng, and Xiang Li. Realpoint3d: An efficient generation network for 3d object reconstruction from a single image. *IEEE Access*, 7:57539–57549, 2019.

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, 2014.

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[23] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in neural information processing systems*, 2016.

[24] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[25] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. In *2017 International Conference on 3D Vision*. IEEE, 2017.

[26] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[27] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision*, 2018.

[28] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[29] Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single rgb images via topology modification networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[30] Priyanka Mandikal and Venkatesh Babu Radhakrishnan. Dense 3d point cloud reconstruction using a deep pyramid network. In *2019 IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2019.

[31] Priyanka Mandikal, KL Navaneet, Mayank Agarwal, and R Venkatesh Babu. 3d-lmnet: Latent embedding matching for accurate and diverse 3d point cloud reconstruction from a single image. *arXiv preprint arXiv:1807.07796*, 2018.

[32] KL Navaneet, Priyanka Mandikal, Mayank Agarwal, and R Venkatesh Babu. Capnet: Continuous approximation projection for 3d point cloud reconstruction using 2d supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8819–8826, 2019.

[33] Haibin Huang, Evangelos Kalogerakis, Ersin Yumer, and Radomir Mech. Shape synthesis from sketches via procedural models and convolutional networks. *IEEE transactions on visualization and computer graphics*, 23(8):2003–2013, 2016.

[34] 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. *International journal of computer vision*, 122(3):411–425, 2017.

[35] Shuai Yang, Zhangyang Wang, Jiaying Liu, and Zongming Guo. Deep plastic surgery: Robust and controllable image editing with human-drawn sketches. *arXiv preprint arXiv:2001.02890*, 2020.

[36] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[37] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[38] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Transactions on Graphics*, 31(4):44:1–44:10, 2012.

[39] Google Inc. *The Quick, Draw! Dataset*, 2017 (accessed May 30, 2020). `https://quickdraw.withgoogle.com/data`.

[40] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[41] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[42] Scott Schaefer, Travis McPhail, and Joe Warren. Image deformation using moving least squares. In *ACM SIGGRAPH 2006 Papers*, pages 533–540. ACM New York, NY, USA, 2006.

[43] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[44] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*. Springer, 2016.

[45] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[46] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *European conference on computer vision*. Springer, 2016.

[47] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *ACM Transactions on Graphics*, 38(6), 2019.

[48] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015.

Supplementary Material

In the supplementary material, we detail baseline method implementations (Section A), demonstrate the standardized sketches from different sketching styles (Section B), and show a real-time 3D modeling with sketches demo (Section C).

A Implementation Details of the Baseline Methods

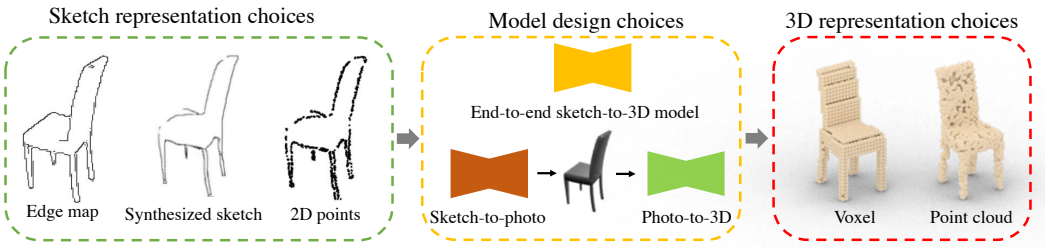

Figure 8: The comparison between different baseline methods and design choices. For the sketch training data, we use either edge maps (as a surrogate), synthesized sketches, or 2D points sampled from synthesized sketches. We either reconstruct 3D shapes from sketches with an end-to-end model or employ a cascaded model for the model design. The cascaded model first synthesizes photos from sketches and then reconstruct 3D shapes from the synthesized photos. For the output 3D representation, we can represent 3D shapes with either voxels or point clouds. The proposed method takes synthesized sketches as the training inputs and reconstructs 3D point clouds with an end-to-end model. To understand each component's benefit, we only change one variant at a time.

We present more details of different baseline methods mentioned in Section 4.3. Figure 8 depicts different baseline methods.

Input: Use Edge Maps Instead of Synthesized Sketches. For each 3D object, we first render 24 photos from different viewing angles and then use the Canny edge detector to extract edge maps. All of the other settings are the same.

Input: Use 2D Point Clouds Instead of the Sketch Images to Represent Sketches. We represent synthesized sketches with 2D point clouds. Specifically, we first binarize a given sketch, and randomly and uniformly sample 512 points from it. The sketch-to-3D model is different since the input is a point cloud, making 2D convolution no longer applicable. To build a strong baseline, we search different depths of networks, different numbers of channels, and different skip connection strategies. Our experimental results show that the following structure gives the best result. The network is adapted from the PointNet [37] and consists of three modules: Multi-Layer Perceptrons (MLP) shared among all points, a max-pooling layer as a symmetric function to extract global features (which is later concatenated with local features of points), and skip-connections of different levels of features, similar to UNet [48]. The MLP applied on points consists of 7 hidden layers with neuron sizes 32, 64, 64, 128, 128, 96, and 80 respectively. Three fully connected layers have dimensions of 12,288, 8,192 and 3,072 respectively are applied to the output of the last MLP layer. The output vector of length 3,072 is then reshaped to 1024×3, which is of the same size of the reconstructed 3D point cloud. We use Adam optimizer [43] with initial learning rate 1e-3. The network is trained for 300,000 iterations with mini-batch size 32.

Model Design: Use the Cascaded Model Instead of the End-to-End Model. We first use CycleGAN [41] to generate synthesized photos from sketches. The network is trained for 100 epochs with an initial learning rate of 0.0002. We further feed the synthesized photo to the photo-to-3D network for 3D shape reconstructions. The network architecture is the same as our sketch-to-3D model.

Output: To Use Voxels Instead of Point Clouds. In this case, we represent 3D objects with voxels. While all other settings are kept the same, the sketch-to-3D model is different since the output modality has changed. We try voxel reconstruction model from several different state-of-the-art methods [1, 26] and find [26] gives the best reconstruction results. All the settings are the same as [26]. Specifically, we use one input view per 3D object and train for 250 epochs with Adam optimizer, and the encoder/decoder initial learning rate is 1e-3.

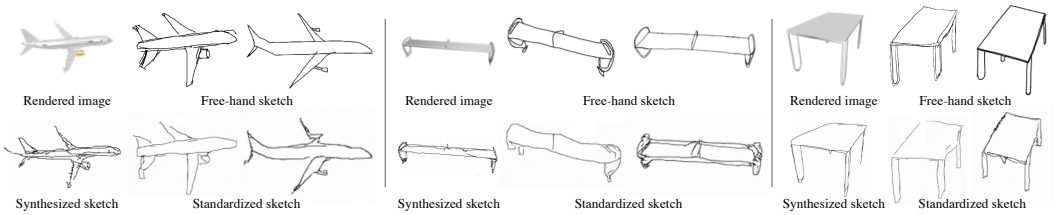

Figure 9: Standardized sketches converted from different styles (by different volunteers). For each rendered image of a 3D object, we show free-hand sketches from two volunteers and the standardized sketches from these free-hand sketches. While the contents are preserved after the standardization process, we can see that the standardized sketches share the style similar to the synthesized ones.

B Sketch Standardization with Different Styles

Fig.9 shows the free-hand sketches of the same objects at a particular viewing angle by different volunteers. Together, we show the standardized sketches of these free-hand sketches and compare them to the synthesized ones. After the standardization module, the sketches share a style similar to synthesized sketches. The standardization module helps domain adaption of sketches with various styles and enhances the generalization of the proposed method.

C Real-time 3D Modeling with Sketches Demo

Along with the paper, we provide a demo of the real-time 3D modeling from free-hand sketches. Please click here to watch the demo.