# MonoSketch3D: A Dual-Channel Framework for 3D Shape Generation from Hand-Drawn Single-View Sketches

Bhavik Chandna Ganesh Bannur Isheta Bansal Jaeyoung Park
Department of Computer Science, UC San Diego
{bchandna, gbannur, i1bansal, jap063}@ucsd.edu

## Abstract

*Sketch-based 3D reconstruction remains a challenging problem due to the abstract, sparse, and highly variable nature of human-drawn sketches. Unlike natural images, sketches lack texture, shading, and depth cues, offering only contour-level information that varies significantly with drawing style, perspective, and skill. Traditional methods often treat sketches as natural images, which fails to address the intrinsic representation gap and leads to subpar reconstruction quality. To overcome these limitations, we propose MonoSketch3D, a dedicated framework for 3D voxel generation from single-view sketches. Building as an extension of the well-known Pix2Vox architecture, MonoSketch3D enhances its foundational principles by introducing a dual-channel processing design tailored for sketch inputs. Our framework processes sketches both as 2D raster images and as unordered 2D point clouds, enabling complementary extraction of pixel-wise and point-wise geometric features through the proposed SktConv and SktPoint modules. To effectively bridge these two feature modalities, we design the Cross-Feature Attention Module (CFAM), which employs multi-head self-attention to fuse semantic and geometric cues into a unified representation, thereby improving reconstruction fidelity across varying sketch styles and viewpoints. We got a strong boost as compared to Pix2Vox architecture, showing a 23% improvement. Code and data available at* https://github.com/GaneshBannur/Sketch_To_3D.

## 1. Introduction

The field of computer vision and graphics has a growing interest in three-dimensional reconstruction of scenes and objects from one or few natural images [8, 14, 16] and video [19], particularly as applications expand across physical environments [2] and virtual metaverse platforms [1, 7, 16, 17]. Among these, reconstructing 3D objects from hand-drawn sketches [6, 15, 20] emerges as a particularly intriguing and intricate challenge.

Hand-drawn sketches offer distinctive advantages for human-computer interaction and creative expression. But unlike natural photographs, sketches capture object contours [9] from specific viewpoints through simplified line representations, and they lack crucial visual information—colors, textures, and detailed structures—making semantic interpretation difficult. The problem increases if we consider viewpoint variations: a three-dimensional object's sketch appearance changes dramatically between top-down and side perspectives, creating fundamental challenges for representation learning algorithms. Also, the resulting drawings vary significantly due to differences in artistic skill, drawing style, and personal interpretation. A central challenge in sketch-based 3D reconstruction lies in effectively extracting meaningful features and capturing underlying semantic cues from input sketches. Traditionally, sketches have been treated as natural images, with conventional deep neural networks or their customized variants applied for feature extraction. However, this approach faces inherent limitations due to the fundamental differences between sketches and natural images. As a result, the quality of reconstructed 3D shapes often remains suboptimal and unsatisfactory.

To address these limitations, we propose an innovative dual-channel framework that does sketch feature modeling through two complementary pathways. Our contributions include:-

- Introduction of a novel framework designed to enhance sketch understanding by extracting both point-wise and pixel-wise features as complementary semantic clues. Rather than limiting the process to pixel-level representations, our approach also models the sketch as a 2D point cloud. This enables the framework to capture richer geometric cues, ultimately leading to higher-quality 3D reconstructions.
- a CFAM module based on the multi-head self-attention mechanism to integrate both pixel features and geometric features into a unified feature representation for

the use of 3D reconstruction. It retains both the features passed into the Pix2Vox-inspired decoder refiner module.

## 2. Related Work

### 2.1. Pixel-wise Feature based 3D Reconstruction

Pix2vox [16] is able to reconstruct high-quality single-view and multi-view 3D voxels from 2D images. Encoder uses pre-trained VGG16 [12] to extract feature map. Then, the decoder generates a coarse volume from this feature map. These coarse volumes from the decoder go over context-aware fusion to obtain high-quality volumes. Lastly, a refiner refines merged context-aware volumes to generate the final output. Pix2vox outperformed 3D-R2N2 in Intersection over Union (IoU) and forward inference time. However, Pix2vox was trained on natural images, so using pixel-wise feature that is generated from VGG16 is not enough for sketch-based 3D reconstruction. Our proposed method addresses this issue by introducing a dual-channel processing. Instead of just using pixel-wise features, we added point-wise features to address the abstract and sparse nature of sketches.

### 2.2. Sketch-based 3D Reconstruction

Sketch-based 3D reconstruction has been an active research topic for a long time. Ambiguities in a sketch pose a problem in determining how the sketch object is posed. Sketch2Model [20] addresses this issue by proposing a view-aware architecture. It explicitly conditioned the generation process on the choice of viewpoint to enable view-aware generation. Sketch2Mesh [6] proposed an encoder/decoder architecture to learn a latent parameterization of sketch and refined a 3D mesh by matching it with external contours. However, these approaches to generating 3D objects from the latent code of sketches are not good at preserving spatial details in the sketch. SketchSampler [5] resolved this issue by using the density map that characterizes the distribution of 2D point clouds. Firstly, it generates feature maps from the sketch translator, and these feature maps are used to predict the density map of the sketch. Then, it samples 2D point clouds from the density map, and depth values are sampled at each 2D point to generate final 3D point clouds. Alternatively, we conducted a depth-aware sampling from the input sketch to generate 2D point clouds rich in 3D information.

## 3. Dataset Curation and Preprocessing

### 3.1. Sketch Generation and Selection

To train our model effectively on the sketch-to-3D reconstruction task, we first processed the raw ShapeNet [3] data into a format suitable for our pipeline, which fuses

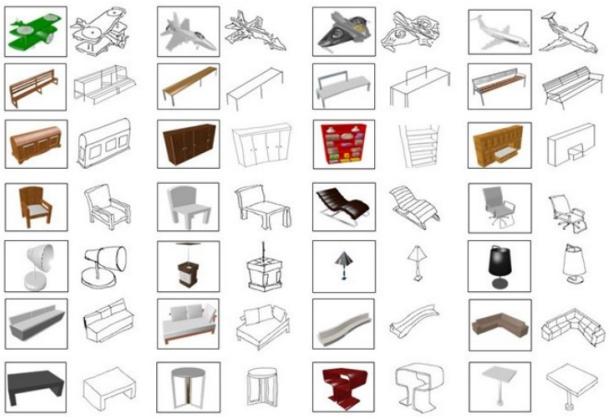

Figure 1. Samples from the Shapenet-Sketch dataset

image-based and point-based features. We selected nine object categories from ShapeNet— aeroplane, bench, cabinet, chair, display, lamp, speaker, sofa, and table—chosen for their recognizability and high intra-class variation.

For each CAD model, ShapeNet provides multiple rendered views. We selected the image closest to the canonical 3/4 front-top perspective, as this viewpoint intuitively captures most geometric details. These selected images were then transformed into sketch-like representations using the Photo-Sketching algorithm, which converts photo-realistic images into edge-based approximations. This simulates human-like freehand sketches and enables the model to learn a meaningful mapping from 2D sketch space to 3D geometry.

### 3.2. Voxel Conversion, Point Cloud Projection, and Data Organization

In parallel with sketch generation, we retrieved the corresponding 3D voxel representations in `.binvox` format from ShapeNet. These were converted into NumPy arrays (`.npy`) for efficient loading in our PyTorch-based pipeline. Each sketch was paired with its aligned 3D voxel representation.

To further enrich the geometric understanding, we generated 2D point clouds rich in 3d information from the sketch images. This involved computing depth maps based on grayscale intensity, where darker regions were interpreted as closer, We use DepthAnythingV2 [18] grayscale for getting depth maps. These depth maps were then projected into 3D using an assumed camera intrinsic matrix, yielding approximate $(x, y, z)$ coordinates.

Each processed sample includes a sketch image (`.png`), a 3D voxel grid (`<model_id>_voxel.npy`), and a corresponding point cloud (`<model_id>_points.npy`). These are stored in a unified directory with consistent naming to ensure ID alignment across modalities.

After preprocessing, we curated a dataset of **12,138** valid samples. Some samples of the dataset can be seen in Figure 1. A **80:20** train-test split was applied randomly to ensure sufficient diversity for robust model training and evaluation.

## 4. Methodology

The proposed framework consists of 4 modules, Sketch Convolution module (SktConv), Sketch Multi-scale Point module (SktPoint), Cross-modal Fusion Attention Module (CFAM), and the 3D shape decoder & refiner module (3D-Decoder). The SktConv module is tailored to extract a pixel-level feature map from an input binary sketch image. In parallel, the SktPoint module focuses on generating a point-level feature map from a 2D point cloud, which is sampled from the same sketch image. These two distinct feature representations are subsequently passed into the CFAM fusion module, which models the interdependencies between modalities and identifies salient information across both feature maps. The refined outputs are then integrated with additional feature representations to enrich contextual information. Finally, the aggregated features are fed into the 3D-Decoder module, which enhances the geometric and structural details of the reconstructed object. Figure 2 shows the overall pipeline of the framework. Each module is discussed in detail in the sections that follow.

### 4.1. SktConv

The SktConv module acts as an encoder to capture the pixel features of the sketch. We adopt a modified VGG-19 [11] network for hierarchical feature extraction from $1 \times 224 \times 224$ binary sketch images. The architecture utilizes:

- The first four convolutional blocks from VGG-19 pre-trained on ImageNet, each with two $3 \times 3$ convolution layers, batch normalization, ReLU activation, and max-pooling. These reduce feature map dimensions to $512 \times 28 \times 28$.
- Two custom convolutional blocks follow, each consisting of a $3 \times 3$ convolutional layer (with 512 and 256 output channels, respectively), batch normalization, and ReLU activation.
- A max-pooling layer with stride 3 is applied at the end to reduce spatial dimensions by one-third, resulting in a feature map $F_i \in \mathbb{R}^{256\times8\times8}$.
- Before input to the CFAM module, $F_i$ is reshaped to $256 \times 64$ by flattening the spatial dimensions.

### 4.2. SktPoint

To better capture the 3D structure of objects, we introduce the SktPoint Module, which operates directly on uniformly sampled contour points, reducing computational overhead while preserving structural fidelity. The input is a point set $S_n = \{p_1, p_2, \ldots, p_n \mid p_i \in \mathbb{R}^2\}$, with $n = 256$ points. Inspired by MCPNet [13], SktPoint employs three parallel convolutional columns with filter sizes $1 \times 2$, $3 \times 2$, and $5 \times 2$, producing output channels of 64, 64, 128, and 1024, respectively. These convolutions transform each point's coordinates into a 1024-dimensional vector, resulting in a feature map of size $n \times 1024$. A global feature vector is then obtained via max pooling across the point dimension and is concatenated with local features of size $n \times 64$ from the second column. This yields a combined representation of size $n \times 1088$, which is processed by three successive $1 \times 1$ convolutions with 512, 256, and 64 output channels to generate the final point-wise feature map $F_p \in \mathbb{R}^{n\times64}$, encoding rich geometric information.

### 4.3. CFAM

Because of the inconsistency between pixel-wise and point-wise data structures, previous methods primarily rely on pixel features to improve RGB representations. The main challenge lies in how to effectively combine and adaptively fuse complementary information from enhanced pixel features and geometric features across different levels. To address this, we propose a Cross-modal Fusion Attention Module (CFAM) that enables robust feature integration.

To enable effective multi-modal fusion, it is essential to first align the shapes of the feature maps. Each $(8 \times 8)$ feature map in the pixel-wise features $F_i$ is treated as a vector, resulting in a reshaped representation $F_i' \in \mathbb{R}^{256\times64}$, which matches the dimensions of the point-wise feature map. The transformed pixel-wise feature map $F_i'$ is then concatenated with the point-wise feature map $F_p$ to produce a combined feature matrix $F_{\text{in}} \in \mathbb{R}^{512\times64}$, which serves as the input to the CFAM fusion module.

CFAM uses Multi-Head Attention (MHA) to optimize distinct feature components of each modality. In our implementation, the MHA consists of six identical Multi-Head Attention layers stacked sequentially. Each layer contains three fully connected layers that project the input into Query, Key, and Value matrices, denoted as $Q, K, V \in \mathbb{R}^{512\times64}$. These matrices are divided into $h = 4$ groups, where each group contains a corresponding Query, Key, and Value matrix. This division enables multiple Scaled Dot-Product Attention heads to operate in parallel. Each head works with matrices of dimension $d_k = d_v = \frac{64}{4} = 16$. The Scaled Dot-Product Attention computes the dot product between each Query vector and all Key vectors, scales the result by $\frac{1}{\sqrt{d_k}} = \frac{1}{\sqrt{16}} = \frac{1}{4}$, and applies a softmax to produce attention weights. These weights are used to aggregate the Value vectors, capturing relationships across the input features. The output of the MHA module is a refined feature representation that feeds into subsequent processing stages.

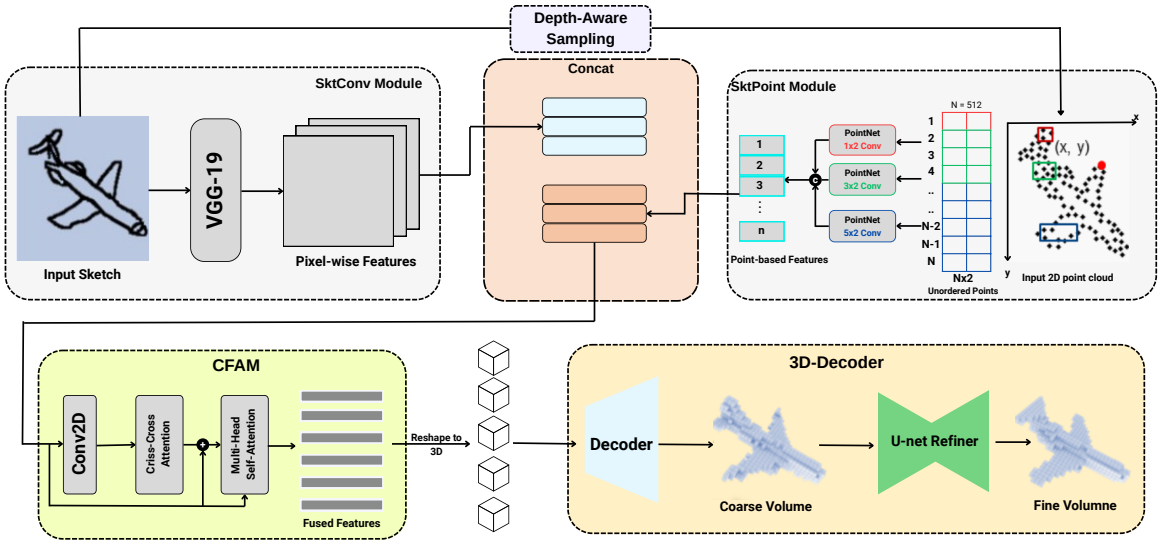


Figure 2. Pipeline of our 3D reconstruction framework

### 4.4. 3D-Decoder

The 3D Decoder transforms 2D feature maps into 3D voxel volumes, progressively increasing the resolution to $32 \times 32 \times 32$. Initially, the fused feature matrix $\mathbf{F}_{out}$ is projected into a 3D space and reshaped into a volume $\mathbf{V}_{in} \in \mathbb{R}^{512\times4\times4\times4}$. The decoder and refiner inside this module are inspired by Pix2Vox and U-Net architectures. The decoder produces a coarse voxel volume representing the 3D model and is basically five 3D transposed convolution layers, and the number of output channels is 512, 128, 32, 8, and 1, respectively. To improve reconstruction quality, a refinement module with an encoder-decoder structure and skip connections is applied, preserving local details and enhancing the final output. The coarse 3D model $\tilde{V}_{coarse}$ is imported into the refiner module to optimize the details such as imperfections on the object's surface or outline away from it and reconstruct the 3D target as $\tilde{V} \in \mathbb{R}^{32\times32\times32}$.

### 4.5. Loss Function

Similar to Pix2Vox, we take out the voxel-wise binary cross-entropies between the reconstructed object and the ground truth but for the false positive and false negative cases only, and define a mean squared loss. The loss is calculated as follows: the space of the ground truth volume is divided into occupied voxels $V_p$ and unoccupied voxels $V_n$, where $P$ and $N$ are the total number of occupied and unoccupied voxels, respectively. FPCE is false positive cross-entropy defined on unoccupied voxels of a ground truth shape volume, and FNCE is false negative cross-entropy defined on occupied voxels of the same volume:

$$\text{FPCE} = -\frac{1}{N}\sum_{j=1}^{N}\left[Vn_j \times \log \widetilde{Vn}_j + (1 - Vn_j) \times \log\left(1 - \widetilde{Vn}_j\right)\right]$$

$$\text{FNCE} = -\frac{1}{P}\sum_{i=1}^{P}\left[Vp_i \times \log \widetilde{Vp}_i + (1 - Vp_i) \times \log\left(1 - \widetilde{Vp}_i\right)\right]$$

where $Vp_i$ represents the occupancy probability of the $i$-th voxel in $V_p$, and $Vn_j$ represents the occupancy probability of the $j$-th voxel in $V_n$, which means $Vp_i = 1$ and $Vn_j = 0$ on the ground truth volume. While $\widetilde{Vp}_i$ and $\widetilde{Vn}_j$ are the predicted probabilities corresponding to $Vp_i$ and $Vn_j$, respectively. Final loss is calculated as :-

$$\mathcal{L} = FPCE^2 + FNCE^2$$

## 5. Experiments

We develop the proposed model using PyTorch and perform experiments on a single NVIDIA A100 GPU. The network predicts a 3D model represented as a voxel grid sized 32 by 32 by 32. We train a single model for all the categories. The model is trained for 100 epochs with a batch size of 256. To optimize training efficiency and stability, we implement mixed precision training using the Accelerate library. We utilize the Adam optimizer with an initial learning rate of 0.001, setting $\beta_1$ and $\beta_2$ to 0.9 and 0.999, respectively. Throughout training, Weights & Biases is integrated to comprehensively track metrics, log parameter updates, and visualize performance trends in real time. At the end of each epoch, a verification process evaluates the updated parameters, providing pivotal feedback to monitor and refine training progress.

| Method | Aeroplane | Bench | Cabinet | Chair | Display | Lamp | Speaker | Sofa | Table | Overall |
|---|---|---|---|---|---|---|---|---|---|---|
| Pix2Vox [16] | 0.4547 | 0.3139 | 0.5417 | 0.2933 | 0.2528 | 0.2844 | 0.4237 | 0.5279 | 0.2652 | 0.3304 |
| 3D-R2N2 [4] | 0.4580 | 0.3067 | 0.5599 | 0.2844 | 0.2453 | 0.2464 | 0.4600 | 0.5178 | 0.2793 | 0.3305 |
| 3D-RETR [10] | 0.4969 | 0.3825 | 0.4884 | 0.3472 | **0.3893** | 0.3223 | 0.4329 | 0.5623 | 0.3360 | 0.3898 |
| Ours (MonoSketch3D) | **0.5478** | **0.4276** | **0.6313** | **0.4520** | 0.3879 | **0.4182** | **0.5511** | **0.6516** | **0.4206** | **0.4795** |

Table 1. Comparison of 3D Reconstruction Methods on Shapenet-Sketch Dataset

### Metric

We will use Voxel-IoU as a metric, where we apply the concept of Intersection over Union (IoU) to 3D space to evaluate the overlap between predicted voxel models and ground truth voxel models. It is computed as the ratio of the intersection volume to the union volume. We binarize the probabilities at a fixed threshold of 0.3. More formally,

$$\text{Voxel-IoU} = \frac{\sum_{i,j,k} \mathbb{I}(p(i,j,k) > t) \cdot \mathbb{I}(\text{gt}(i,j,k))}{\sum_{i,j,k} \mathbb{I}(p(i,j,k) > t) + \mathbb{I}(\text{gt}(i,j,k))}$$

where $p(i,j,k)$ and $\text{gt}(i,j,k)$ represent the predicted occupancy probability and the ground truth at voxel location $(i,j,k)$, respectively. $\mathbb{I}(\cdot)$ denotes the indicator function and $t = 0.3$ is the voxelization threshold. Higher IoU values indicate better reconstruction results.

A higher Voxel-IoU indicates a greater similarity between the two point clouds.

## 5.1. Results

### Quantitative Results

Table 1 shows the Voxel-IoU scores on the Shapenet-Sketch test set across all nine object categories. We compare our method with three prior deep learning-based baselines that represent distinct approaches to single-view 3D reconstruction from 2D inputs.

Pix2Vox [16] employs a coarse-to-fine voxel-based generation pipeline, where a shared encoder extracts image features that are then refined through a context-aware fusion module. 3D-R2N2 [4] uses a recurrent neural network to aggregate features over a sequence of images and regress voxel occupancy grids. It was originally designed for multi-view settings but can be adapted for single-view inputs. 3D-RETR [10] introduces a retrieval-augmented reconstruction paradigm that incorporates shape priors via learned latent embeddings, offering a hybrid between retrieval and generation. It performs better than previous baselines on sketch inputs by leveraging prior knowledge, yet struggles with geometric precision in fine-grained structures.

Our method surpasses all previous approaches in all categories except display. By jointly modeling point-wise and pixel-wise features from sketches as complementary semantic clues, it achieves significant gains in reconstruction quality, culminating in an overall IoU of 0.4795, notably outperforming 3D-RETR's 0.3898.

### Qualitative Results

Figure 3 shows the best improvement over the previous methods.

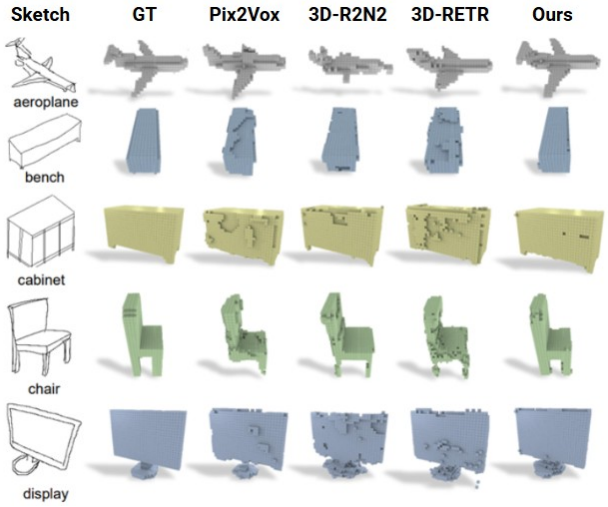


Figure 3. Qualitative Results - Comparison of the output voxel volume for different sketches from different categories.

## 6. Limitations

Our approach effectively captures the fine details present in hand-drawn sketches. However, it is not based on a probabilistic generative model, meaning it does not explicitly learn the distribution of 3D object shapes. As a result, the generated point clouds and meshes may not always align with the true underlying structure. This can lead to reduced quality in the output, particularly when handling complex inputs. Incorporating probabilistic generative models in future work could help overcome this limitation and improve the visual realism of the generated 3D objects.

## 7. Conclusion

In this paper, we propose a deep learning framework for 3D reconstruction from a single, monocular sketch. Our approach uses a self-attention mechanism to fuse multi-modal

sketch data. The loss function is designed to balance multiple objectives, leading to improved reconstruction quality. We show results on the ShapeNet-Sketch dataset, highlighting the effectiveness of our method in capturing 3D structure from sketches.

For future work, we plan to integrate a module similar to SketchSampler to generate more informative 3D point cloud priors, moving beyond 2D inputs and enabling better depth estimation that reveals hidden parts of the object. Additionally, we aim to conduct human studies and test our framework on other sketch datasets to evaluate generalization and perceptual quality. We are also interested in exploring video diffusion models to incorporate temporal information, which could help improve consistency and realism in the reconstructed shapes.

## References


[1] Yara Jamil Alkhatib, Anna Forte, Gabriele Bitelli, Roberto Pierdicca, and Eva Malinverni. Bringing back lost heritage into life by 3d reconstruction in metaverse and virtual environments: The case study of palmyra, syria. In *International Conference on Extended Reality*, pages 91–106. Springer, 2023. 1

[2] Yonge Bai, LikHang Wong, and TszYin Twan. Survey on fundamental deep learning 3d reconstruction techniques. *arXiv preprint arXiv:2407.08137*, 2024. 1

[3] 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. 2

[4] 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 *Computer vision–ECCV 2016: 14th European conference, amsterdam, the netherlands, October 11-14, 2016, proceedings, part VIII 14*, pages 628–644. Springer, 2016. 5

[5] Chenjian Gao, Qian Yu, Lu Sheng, Yi-Zhe Song, and Dong Xu. Sketchsampler: Sketch-based 3d reconstruction via view-dependent depth sampling. In *ECCV*, 2022. 2

[6] B. Guillard, E. Remelli, P. Yvernay, and P. Fua. Sketch2mesh: Reconstructing and editing 3d shapes from sketches. In *ICCV*, 2021. 1, 2

[7] Weipeng Jing, Shijie Wang, Wenjun Zhang, and Chao Li. Reconstruction of neural radiance fields with vivid scenes in the metaverse. *IEEE Transactions on Consumer Electronics*, 70(1):3222–3231, 2023. 1

[8] Kebin Peng, Rifatul Islam, John Quarles, and Kevin Desai. Tmvnet: Using transformers for multi-view voxel-based 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 222–230, 2022. 1

[9] Jhony K Pontes, Chen Kong, Sridha Sridharan, Simon Lucey, Anders Eriksson, and Clinton Fookes. Image2mesh: A learning framework for single image 3d reconstruction. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part I 14*, pages 365–381. Springer, 2019. 1

[10] Zai Shi, Zhao Meng, Yiran Xing, Yunpu Ma, and Roger Wattenhofer. 3d-retr: End-to-end single and multi-view 3d reconstruction with transformers. *arXiv preprint arXiv:2110.08861*, 2021. 5

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[12] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[13] Bor-Shiun Wang, Chien-Yi Wang, and Wei-Chen Chiu. Mcpnet: An interpretable classifier via multi-level concept prototypes. In *CVPR*, 2024. 3

[14] Fei Wang, Shujin Lin, Hefeng Wu, Hanhui Li, Ruomei Wang, Xiaonan Luo, and Xiangjian He. Spfusionnet: Sketch segmentation using multi-modal data fusion. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1654–1659. IEEE, 2019. 1

[15] Jiayun Wang, Jierui Lin, Qian Yu, Runtao Liu, Yubei Chen, and Stella X Yu. 3d shape reconstruction from free-hand sketches. In *European Conference on Computer Vision*, pages 184–202. Springer, 2022. 1

[16] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *ICCV*, 2019. 1, 2, 5

[17] Haozhe Xie, Hongxun Yao, Shengping Zhang, Shangchen Zhou, and Wenxiu Sun. Pix2vox++: Multi-scale context-aware 3d object reconstruction from single and multiple images. *International Journal of Computer Vision*, 128(12):2919–2935, 2020. 1

[18] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024. 2

[19] Xingbin Yang, Liyang Zhou, Hanqing Jiang, Zhongliang Tang, Yuanbo Wang, Hujun Bao, and Guofeng Zhang. Mobile3drecon: Real-time monocular 3d reconstruction on a mobile phone. *IEEE Transactions on Visualization and Computer Graphics*, 26(12):3446–3456, 2020. 1

[20] S.H. Zhang, Y.C. Guo, and Q.W. Gu. Sketch2model: View-aware 3d modeling from single free-hand sketches. In *CVPR*, 2021. 1, 2