Supplementary Material for 'LASA: Instance Reconstruction from Real Scans using A Large-scale Aligned Shape Annotation Dataset'

Haolin Liu[1,2*], Chongjie Ye[1,2*], Yinyu Nie[3], Yingfan He[1,2], Xiaoguang Han[2,1†]
[*]equal contribution [†]corresponding author
[1]FNii, CUHKSZ [2]SSE, CUHKSZ [3]Technical University of Munich
gap-lab-cuhk-sz.github.io/LASA

1. Overview

The supplementary material includes a video and this pdf document. Please refer to the homepage to access the video for more dataset visualization and DisCo's in-the-wild reconstruction results.

1.1. Scan Registration and Alignment

We conduct a two-stage alignment method to match the laser scan and RGB-D sequence coordinates for annotations. First, we use Pointsect[3] to render pseudo images from the comprehensive point cloud and align them with the RGB-D footage using COLMAP[8]. This gives an initial global transformation matrix. Next, we refine the transformation matrix further with Generalized ICP (GICP)[9]. Doing a coarse global register through structure-from-motion first, then dialing it in with GICP, delivers an accurate and sturdy alignment between data types.

1.2. dataset class statistics

Fig. 1 presents the distribution of annotated objects across different classes. LASA is far superior to Scan2CAD in terms of scale. LASA contains 3 times more unique CAD models - 10k vs 3k in Scan2CAD. This allows representing a greater variety of object shapes and forms. While LASA has 17 object classes, focusing on furniture items, versus 35 classes in Scan2CAD, it makes up for this with order-of-magnitude more annotations per class.

1.3. Detail implementation of DisCo

In DisCo, we implement our triplane VAE model on the resolution of 128×128. During the training of VAE, the latent triplane is randomly downsampled to 64×64 for learning a robust decoder with low-resolution input. When training the VAE, several augmentations are used to train a robust model, which is randomly shifting -0.1m to 0.1m from the centers (already normalized from -1m to 1m), rotating between -15 to 15 degrees, and scaling from 0.7 to 1.0. the kl weight for the $\mathcal{L}_{kl}$ is set to 0.025. The triplane diffusion is conducted on 64×64 triplane.

During the training, for each object, 1-5 images are randomly sampled. For evaluation, we typically chose up to 10 images as inputs. There are six super categories that are trained, which are Chair, Table, Bed, Cabinet, Shelf, Sofa. Among them, Chair contains sub-categories of stool and chair, Cabinet contains cabinet, oven, refrigerator, and dishwasher.

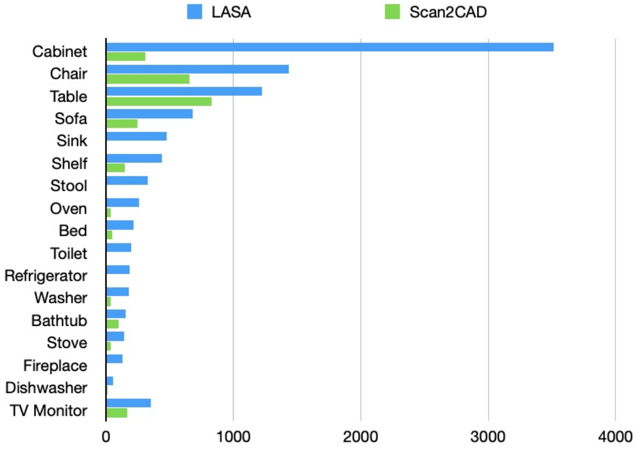

Figure 1. Dataset statistic of LASA and Scan2CAD dataset.

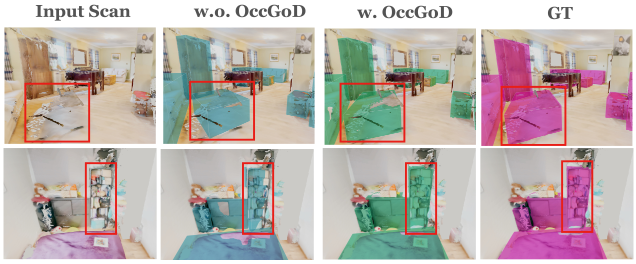

Figure 2. Comparison of the object detection results between without OccGOD and with OccGOD.

Train On chair of	(a)LASA	(b)Syn	(c)LASA's syn	Syn	LASA's syn
Test On chair of	LASA			Syn	LASA's syn
IFNet	26.4 / 24.6 / 22.5	21.6 / 18.0 / 20.1	21.0 / 15.9 / 21.0	57.2 / 4.35 / 49.8	41.0 / 7.26 / 43.9
3DShape2VecSet	27.1 / 7.32 / 21.7	23.8 / 10.7 / 18.1	25.5 / 8.23 / 20.9	51.7 / 3.21 / 31.8	33.8 / 5.60 / 29.2
LAS-pts	- / 11.0 / 21.7	- / 19.5 / 17.3	- / 17.1 / 20.2	- / 5.8 / 45.3	- / 7.50 / 25.7
Ours-pts	30.1 / 6.69 / 23.9	26.3 / 10.5 / 20.1	28.5 / 8.39 /23.5	63.0 / 2.35 / 47.4	47.7 / 2.93 / 44.6
Ours-pts+img	35.6 / 4.35 / 27.4	26.9 / 10.0 / 22.6	32.0 / 7.01 / 24.9	69.3 / 1.13 / 55.2	50.2 / 2.56 / 47.5

Table 1. Domain gap analysis on chair category. The evaluation metrics are mIoU / chamfer L2 / F-score respectively.

LASA dataset follows train/test/val splits with 70%/20%/10% random partition for training and evaluation. Notice that the splits are conducted at the scene level.

1.4. Detail implementation of OccGOD

In OccGOD, the occupancy prediction head consists of two components: a feature pyramid neck and a regression head. The feature pyramid neck takes in multi-level features from the backbone of the model. To enhance the resolution and density of the output feature grid, we employ 3D generative convolution blocks for upsampling. After the feature pyramid neck, a single-layer 3D convolution is applied for voxel occupancy prediction. To demonstrate the effectiveness of OccGOD, we provide visualization results in Fig. 2.

1.5. Domain gap analysis

We claim training solely on synthetic data will cause domain gap problems when inference on real data. We conduct further experiments on the chair category to verify the existence and the sources of domain gaps in the table 1

Broadly, there are two types of domain gaps: **Input gap** is between real and synthesized scans: synthesized scan is produced by fusing depth renderings of CAD annotations, whereas real scans are captured using RGBD sensors; The **output gap** is between annotation's distribution of different dataset: synthetic dataset such as ShapeNet is collected from the web while LASA is from real scenarios. To verify these domain gaps, we compare training on three datasets: (a) **LASA**, (b) **Syn**, and (c) **LASA's syn**. **LASA's syn** is modified from LASA with synthesized input point clouds. It introduces an input domain gap compared with training on LASA. **Syn** refers to synthetic dataset comprising ShapeNet, ABO and 3D-Future, which includes both types of gaps. To eliminate the effect of dataset size, we use the same number of training samples across all types of training datasets. The performance gap between (a) and (c) indicates the existence of the input gap. The performance gap between (b) and (c) indicates the existence of the output gap. We observe that both gaps exist, and LASA bridges the domain gap ensuring real-world generalization.

1.6. Compared with training on all categories

We further compare the performance of training on each category individually and train on all categories. The comparison is shown in Table 2. Both have similar performance, except on Shelf. Training on all categories has significant improvement for the shelf category. I speculate the reason for it would be the small number of training samples while solely training the shelf category.

Strategy	Chair	Sofa	Table
Train individually	**38.6** / 3.57/ 31.0	70.7/ 2.88/ 31.6	**41.5** / **6.52** / **36.1**
Train on all	37.2 / **3.56** / **31.4**	**71.1** / **2.56** / **32.6**	39.1 / 6.73 / 35.4
	Cabinet	Bed	Shelf
Train individually	**75.1** / 3.10 / **37.0**	**62.5** / **2.62** / **35.4**	24.5 / 3.45 / 37.5
Train on all	74.4 / **3.07** / 34.5	62.3 / 2.72 / 34.1	**27.1** / **2.68** / **38.0**

Table 2. Comparison between training on each category individually and training on all categories together. The evaluation metrics are mIoU / chamfer L2 / F-score respectively.

1.7. Compared with image-to-3D method.

We claim that supported by both LASA and DisCo, our method produces better results than current image-to-3D methods such as OpenLRM[5, 6] and one-2-3-45[7] as shown in Figure 3. The reasons are two-fold, first, these methods or their backbones are trained on synthetic data, introducing domain gap problems when applying to real-world images. This necessitates LASA, which may potentially alleviate the domain gap problem for these image-to-3D methods. Second, due to lacking input spatial information, current single image-to-3D methods are not able to output mesh that is spatially aligned with the input scene.

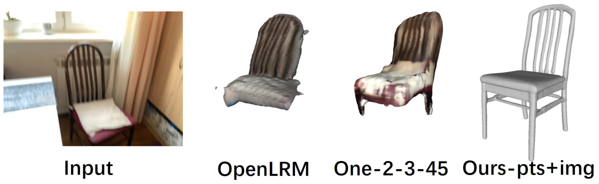

Figure 3. Compare our method with image-to-3D methods.

1.8. Dataset Comparison under the same scene

We further compare Scan2CAD[1] with LASA under the same scene. Since Scan2CAD is annotated on ScanNet[4] while LASA is on ArkitScene[2], they cannot compare directly. We annotate one sample on ArkitScene using methods in Scan2CAD, as shown in the below figure. LASA dataset has better quality than Scan2CAD.

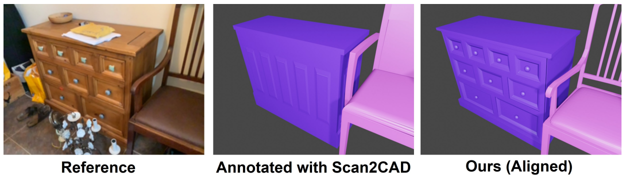

Figure 4. Comparison between LASA and Scan2CAD under the same scene.

1.9. More dataset visualization

We provide examples of both object-level and full scene-level visualizations in Fig. 5 and Fig. 6. The left-most image shows the point cloud from laser scans. The middle image displays a mesh model fused from the data captured by an iPhone ToF camera. The right-most image is the render of the annotated CAD models. For video visualization, please refer to the supplementary video.

1.10. More in-the-wild results

In-the-wild results are included in the video (from 1:04 to 7:25). We first capture RGB-D video using the iPhone's ToF sensor or use the existing RGB-D sequence from ArkitScene. Then, the RGB-D frames are fused into scene scans. OccGoD is then employed for detection using the scene scans as inputs. Then, for each detected object, we compute its visibility in every frame. More specifically, we crop the points inside the detected 3d bounding box, then project to each frame, and compare with the depth map followed by counting the number of points that are in front of the depth map, as the number of visible points. Afterwards, up to 10 frames are chosen from those with more than 1,024 visible points. These frames and the cropped point cloud will be the inputs of DisCo to produce the reconstruction results. Finally, the reconstructed mesh will be put back to the scene according to the detection result.

References

[1] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 2614–2623, 2019. 3

[2] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021. 3

[3] Jen-Hao Rick Chang, Wei-Yu Chen, Anurag Ranjan, Kwang Moo Yi, and Oncel Tuzel. Pointersect: Neural rendering with cloud-ray intersection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8359–8369, 2023. 1

[4] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 3

[5] Zexin He and Tengfei Wang. Openlrm: Open-source large reconstruction models. https://github.com/3DTopia/OpenLRM, 2023. 2

[6] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023. 2

[7] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Zexiang Xu, Hao Su, et al. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *arXiv preprint arXiv:2306.16928*, 2023. 2

[8] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1

[9] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, page 435. Seattle, WA, 2009. 1

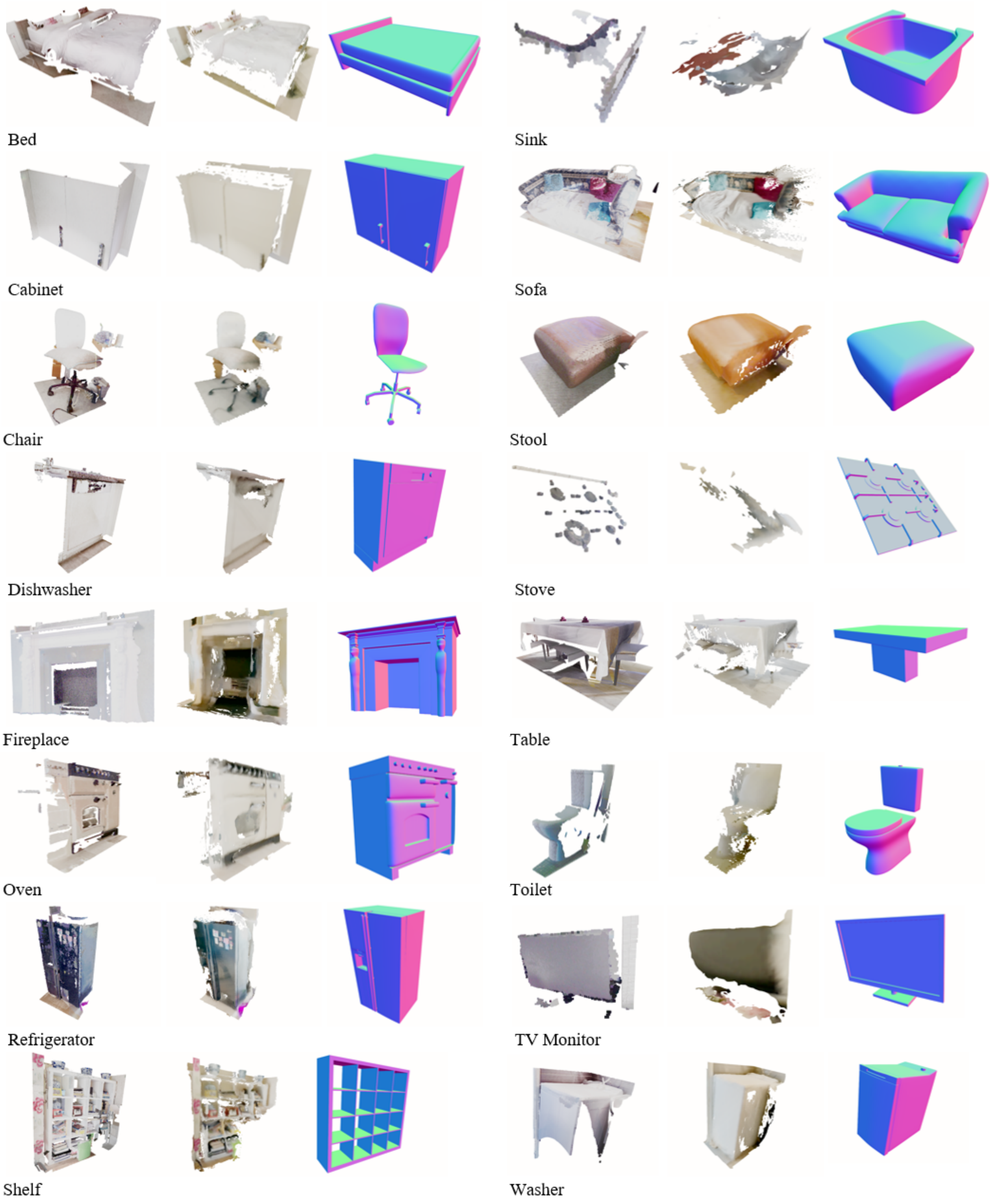

Figure 5. Examples of visualizations for different object types.

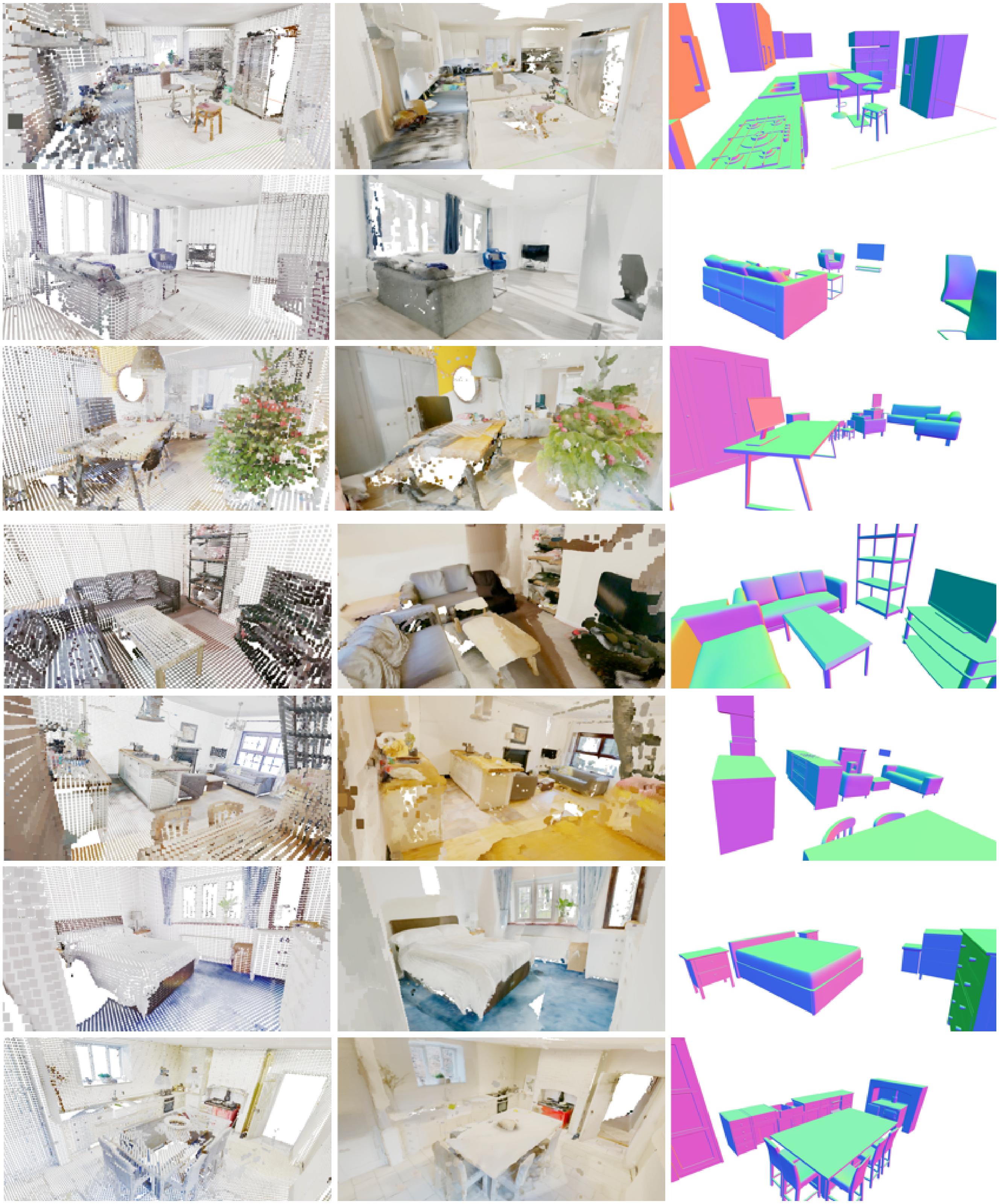

Figure 6. Examples of visualizations for full scenes.