Vis2Mesh: Efficient Mesh Reconstruction from Unstructured Point Clouds of Large Scenes with Learned Virtual View Visibility Supplementary Material

Shuang Song
The Ohio State University

Zhaopeng Cui*
Zhejiang University

Rongjun Qin†
The Ohio State University

In this supplementary material, we first explain our graph weighting scheme with an example in Section A. Then we present the details of our training dataset in Section B. In Section C, we describe the detailed network architecture. Next, we present the definition of the metrics for surface evaluation in Section D. At last, we provide additional experimental results in Section E.

A. Graph Weighting

Corresponding to Section 3 in the main paper, we present the tetrahedra graph weighting scheme of the basic method [3] by illustrating equations with their geometric meaning. As shown in Figure 1, a line of sight traverse a series of tetrahedra (where $M = 5$), from T_1 to T_5 and extended to T_6 (T_{M+1} in the main paper) which is the tetrahedron right behind the end point p. The weighting process starts from the point of view (c) marking it as the source node (Equation 2a in the main paper). Along the line of sight, accumulate the value α_v to the connectivity of facets that line of sight passes through from the front (Equation 2c in the main paper). As for T_6 behind the point, it is marked as a sink node with constant weight (Equation 2b in the main paper).

B. Training Dataset

Our training dataset is built based on textured mesh models of public Multi-View Stereo (MVS) reconstruction dataset BlendedMVS [4]. We select 5 textured models as ground truth surfaces for training and 2 models for testing. For each model, we uniformly sample points from the surface. The number of sampled points depends on the actual size and complexity of the structure, ranging from 500K to 10M points. We prefer sparse sampling since the more occluded points have been projected to the virtual views, the more contrastive samples we will have. We collect datasets by sampling virtual views (Section 4.1 in the main paper)

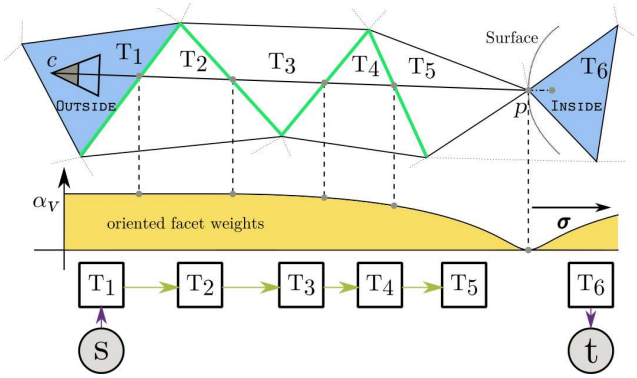


Figure 1. Visibility and graph construction [3]. From top to bottom, **Upper:** A line of sight from a reconstructed 3D point traverses a sequence of tetrahedra, the graph construction, and the assignment of weights to the tetrahedron and oriented facets. **Middle:** Soft visibility decay along the line of sight which is inversely proportional to the distance to the end point. **Lower:** Corresponding s-t graph and the cut solution.

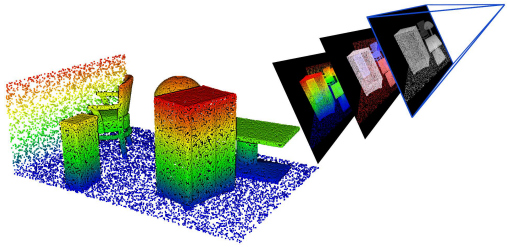
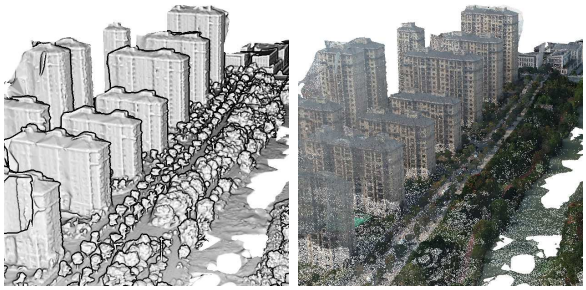


Figure 2. Record visibility information in an image by rendering: While points were rendered, the global index of the points been recorded simultaneously.

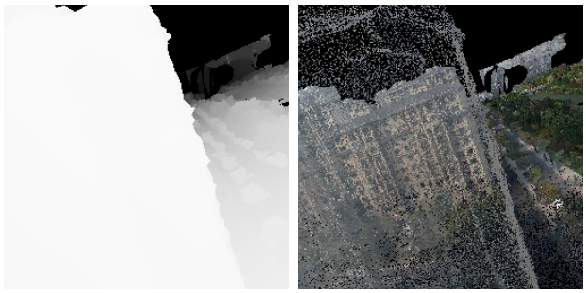
and rendering them with our renderer. Our renderer is implemented with OpenGL, which not only provides regular color images and depth images but also records the original index of each projection point, as shown in Figure 2.

Figure 3 shows an example out of 1414 virtual views. The ground truth visibility is generated by comparing the point-rendering depth map and the surface-rendering depth map. If the difference of depth is less than $\epsilon = 0.05$, it will be marked as visible, otherwise, it will be marked as occluded.

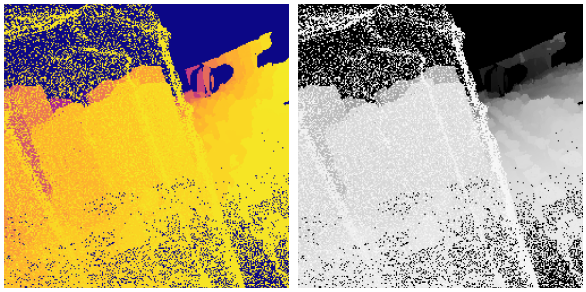
*Contribution initiated and performed at ETH Zürich.
†Corresponding author. Email: qin.324@osu.edu



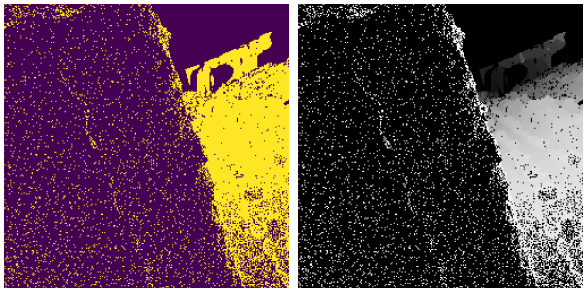
(a) Ground truth surface (b) Sampled point cloud



(c) Surface-rendering depth map (d) Point-rendering color image



(e) Point-rendering ID map (f) Point-rendering depth map



(g) Ground truth visibility (h) Ground truth cleaned depth map

Figure 3. The content of our dataset.

C. Architecture

Table 1 provides a detailed description of the input and output tensor sizes of each module in our workflow. And Table 2 presents the detailed architecture of the network, including the size of each buffer.

D. Surface Evaluation Metrics

The quantitative comparison of two mesh surfaces is a complicated problem. Alternatively, we employ highly oversampled points of triangle surfaces to approximate mesh to mesh metrics. In our experiment, we control the number of points by the density of sampling. Furthermore, to eliminate the randomness introduced by the sampling method, the ground truth mesh is sampled twice to determine the scale factors for the following metrics.

Chamfer distance (CD) [1] . For each point, we find the nearest neighbor in the other set and sums the distances up. To ensure Chamfer distance as symmetric, we sum distances of $\mathcal{P}$ to $\mathcal{Q}$ and $\mathcal{Q}$ to $\mathcal{P}$. K is a scale factor to leverage between different datasets: $K = \frac{1}{CD(\mathcal{S}_1,\mathcal{S}_2)}$.

$$\text{CD}(\mathcal{P},\mathcal{Q}) = \frac{K}{|\mathcal{P}|}\sum_{p\in\mathcal{P}}\min_{q\in\mathcal{Q}}\|p-q\|_2 + \frac{K}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}}\min_{p\in\mathcal{P}}\|p-q\|_2. \quad (1)$$

F-score [2] . As shown in Equation 2, the F-score is defined as the harmonic mean between precision and recall. $\mathcal{P} \subseteq \mathbb{R}^3$ is the point set sampled from the generated surface while $\mathcal{Q} \subseteq \mathbb{R}^3$ is the point set sampled from the ground truth surface. Threshold T is determined by computing maximum distance between two point sets ($\mathcal{S}_{1,2} \subseteq \mathbb{R}^3$) randomly sampled from the same ground truth surface: $T = \max_{s_1\in\mathcal{S}_1}(\min_{s_2\in\mathcal{S}_2}\|s_1 - s_2\|_2)$.

$$\text{Precision}(\mathcal{P},\mathcal{Q}) = \frac{\sum_{q\in\mathcal{Q}}\delta[\min_{p\in\mathcal{P}}\|p-q\|_2 < T]}{|\mathcal{Q}|}, \quad (2a)$$

$$\text{Recall}(\mathcal{P},\mathcal{Q}) = \frac{\sum_{p\in\mathcal{P}}\delta[\min_{q\in\mathcal{Q}}\|p-q\|_2 < T]}{|\mathcal{P}|}, \quad (2b)$$

$$\text{F-score}(\mathcal{P},\mathcal{Q}) = 2\cdot\frac{\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}. \quad (2c)$$

E. Additional Experimental Results

In this section, we provide additional experimental results for better evaluation of our method. Table 3 shows the data volume, source, and corresponding view generator for all the datasets we used in our experiments. We can see that different scales of data are evaluated in our experiment.

E.1. Extra Evaluation of Visibility Classifiers

In this section, we provide test scores on two more datasets in Table 4 and Table 5 to evaluate our VDVNet with datasets of different modalities, which are multi-view stereo dataset and LiDAR scanning dataset.

E.2. Performance on Extreme Sparse Data

As shown in Figure 4, We used a series of downsampled bull datasets to evaluate the performance of our method on

Module	IO	Description	Data Dimension
Virtual View Sampler	Input	Given Point Cloud	$n \times 3$
	Input	The flag indicate the pattern of view generator	1
	Output	6 DoF poses	$m \times 6 \times 1$
Renderer	Input	Given Point Cloud	$n \times 3$
	Input	6 DoF pose	6×1
	Output	Rendered sparse color image	$H \times W \times 3$
	Output	Rendered sparse depth map	$H \times W \times 1$
	Output	Rendered sparse point ID map	$H \times W \times 1$
CoarseVisNet	Input	Normalized and binary mask attached sparse depth map	$H \times W \times 2$
	Output	Predicted mask of visible pixels	$H \times W \times 1$
Multiplication	Input	Rendered sparse depth map	$H \times W \times 1$
	Input	Predicted mask of visible pixels	$H \times W \times 1$
	Output	Cleaned sparse depth map	$H \times W \times 1$
DepthCompNet	Input	Normalized and binary mask attached cleaned sparse depth map	$H \times W \times 2$
	Output	Completed dense depth map	$H \times W \times 1$
Concatenation	Input	Normalized and binary mask attached sparse depth map	$H \times W \times 2$
	Input	Normalized and binary mask attached cleaned sparse depth map	$H \times W \times 1$
	Output	Concatenated raw depth and completed depth map	$H \times W \times 3$
FineVisNet	Input	Concatenated raw depth and completed depth map	$H \times W \times 3$
	Output	Predicted mask of visible pixels	$H \times W \times 1$
Graph-cut based Surface Reconstruction	Input	Given Point Cloud	$n \times 3$
	Input	All rendered sparse point ID map	$m \times H \times W \times 1$
	Input	Predicted mask of visible pixels	$m \times H \times W \times 1$
	Output	Reconstructed triangle surface	Irregular

Table 1. Modules overview. We detail the input and output of all modules that appeared in our workflow. n is the number of input points, m is the number of generated virtual views. H and W are the customized values which are set to 256×256 in our experiments.

Part	Layer	Parameters	Output Dimension
Encoder	Double PConv1BnReLU	3x3,64,64	$H \times W \times 64$
	MaxPool + Double PConv2BnReLU	2,3x3,128,128	$H/2 \times W/2 \times 128$
	MaxPool + Double PConv3BnReLU	2,3x3,256,256	$H/4 \times W/4 \times 256$
	MaxPool + Double PConv4BnReLU	2,3x3,512,512	$H/8 \times W/8 \times 512$
	MaxPool + Double PConv5BnReLU	2,3x3,512,512	$H/16 \times W/16 \times 512$
Decoder	Bilinear Upsample1	2	$H/8 \times W/8 \times 512$
	Concat1	cat(PConv4,Unsample1)	$H/8 \times W/8 \times 1024$
	Double PConv6BnReLU	3x3,256,256	$H/8 \times W/8 \times 256$
	Bilinear Upsample2	2	$H/4 \times W/4 \times 256$
	Concat2	cat(PConv3,Unsample2)	$H/4 \times W/4 \times 512$
	Double PConv7BnReLU	3x3,128,128	$H/4 \times W/4 \times 128$
	Bilinear Upsample3	2	$H/2 \times W/2 \times 128$
	Concat3	cat(PConv2,Unsample3)	$H/2 \times W/2 \times 256$
	Double PConv8BnReLU	3x3,64,64	$H/2 \times W/2 \times 64$
	Bilinear Upsample4	2	$H \times W \times 64$
	Concat4	cat(PConv1,Unsample4)	$H \times W \times 128$
	Double PConv9BnReLU	3x3,64,64	$H \times W \times 64$
	Sigmoid	-	$H \times W \times 1$

Table 2. Detailed encoder-decoder network architecture used for our **CoarseVisNet**, **DepthCompNet**, and **FineVisNet**. **PConv**: partial convolution layer. **Double PConvBnReLU**: PConv+BatchNorm+ReLU+PConv+BatchNorm+ReLU. Parameter of **Double PConvBnReLU**: kernel size, number of filters for the first PConv, and the second PConv.

Name	# Points	Scale	Source	Generator
Triceratops	25K	Object	P2M	Spherical
Tiki	5K	Object	P2M	Spherical
Giraffe	25K	Object	COSEG	Spherical
Bull	25K	Object	COSEG	Spherical
DSLR	25K	Object	BlendedMVS	Spherical
Birdcage	25K	Object	Thingi10k	Spherical
Room0	100K	Indoor	CONet	User
Room1	100K	Indoor	CONet	User
MPT:SingleFloor	400K	Indoor	Matterport	User
MPT:MultiFloor	500K	Indoor	Matterport	User
Toronto Downtown #1	177K	Outdoor	ISPRS	Nadir
Toronto Downtown #2	300K	Outdoor	ISPRS	Nadir
Columbus City	925K	Outdoor	OGRIP	Nadir
Church	500K	Outdoor	BlendedMVS	Oblique
Archway	500K	Outdoor	BlendedMVS	Oblique
Dragon Park	500K	Outdoor	BlendedMVS	Oblique
Dragon Park (Teaser)	5M	Outdoor	BlendedMVS	Oblique
Eco Park	500K	Outdoor	BlendedMVS	Oblique
Pedestrian street	500K	Outdoor	BlendedMVS	Oblique
YParc	2M	Outdoor	senseFly	Nadir
UThammasat	5M	Outdoor	senseFly	Nadir
Hotel	5M	Outdoor	senseFly	Oblique
GSM Tower	3.3M	Outdoor	senseFly	User
Street View	5M	Outdoor	MAI	User
Crossroad	2.6M	Outdoor	KITTI	User

Table 3. Details of data of our experiments, including their source, volume of the data, as well as virtual view generator.

Visibility Estimator	%P	%R	%F1	%AUC
HPR	90.12	79.58	86.2	75.81
UNet	89.54	84.21	86.6	75.89
UNet + PConv	**90.23**	84.7	87.18	78.09
VISIBNET	89.13	86.74	87.74	75.8
Ours w/o DepthComp or PConv	89.49	86.43	87.75	77.37
Ours w/o DepthComp	89.65	90.25	89.83	78.73
Ours w/o PConv	89.6	89.28	89.33	78.17
Ours	90.09	**93.3**	**91.52**	**81.59**

Table 4. Quantitative analysis of methods on binary visibility classification task on MVS dataset.

Visibility Estimator	%P	%R	%F1	%AUC
HPR	80.6	85.48	82.14	81.29
UNet	78.71	75.58	76.6	79.84
UNet + PConv	77.98	82.49	79.67	81.17
VISIBNET	76.98	85.59	80.52	80.76
Ours w/o DepthComp or PConv	77.17	85.18	80.26	79.11
Ours w/o DepthComp	78.00	87.23	81.81	81.69
Ours w/o PConv	79.29	91.29	84.31	83.57
Ours	**80.4**	**93.71**	**85.27**	**83.96**

Table 5. Quantitative analysis of methods on binary visibility classification task on Simulated LiDAR dataset.

very low-density points. The reconstruction quality of our method drops obviously as long as the number of points decreasing. It is because the initial mesh created by Delaunay is a bottom-up approach, and the graph-cut only applies a smooth constraint. The core idea of the proposed method is to recover details in the mesh reconstruction process based on correct visibility recovery, where these original point measurements are available. Therefore having sufficient points will show this advantage.

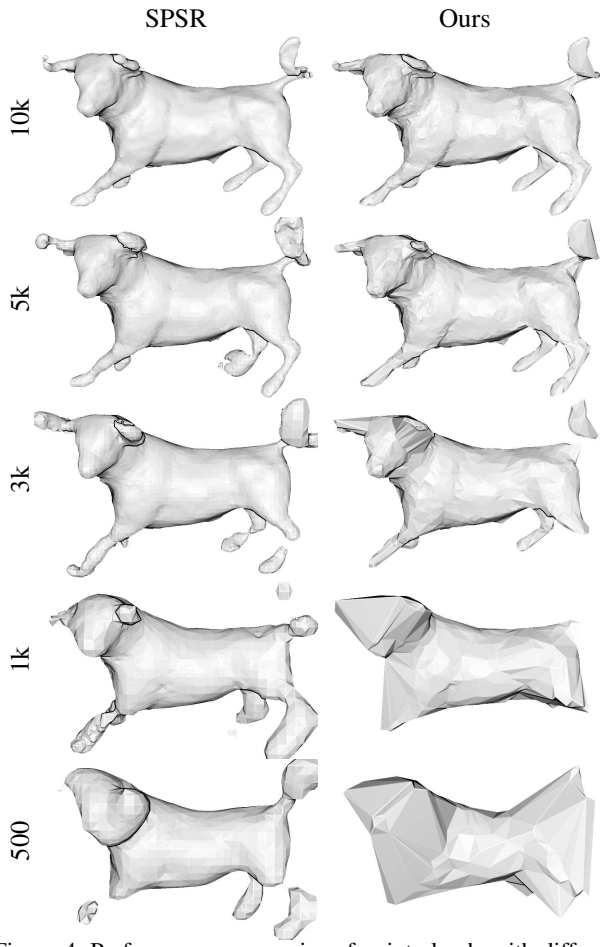


Figure 4. Performance on a series of point clouds with different densities.

E.3. Efficiency Statistics

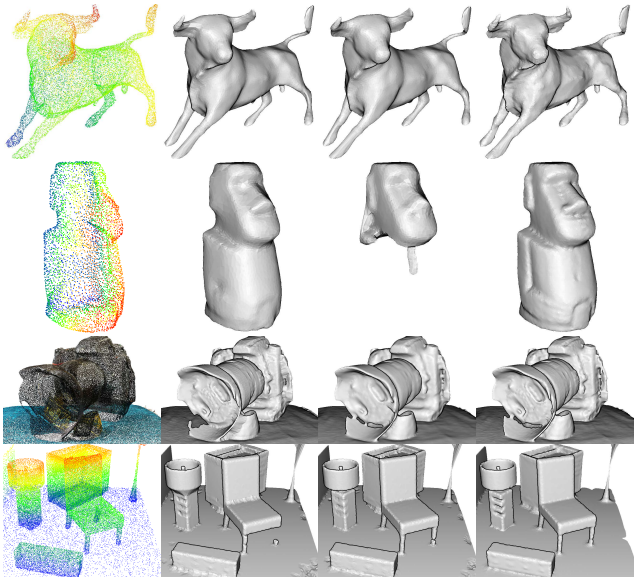
We report the processing time for each component of our method in Table 6 and compare it with the SPSR method. Efficiency performance is evaluated on the Laptop with Intel-4700MQ, 16 GB RAM, and Nvidia 970M (6GB GRAM), which is a challenge on such a resource-limited platform for reconstruction approaches. The processing time of our method increases with the number of points and virtual views, and the SPSR is related to the depth of the octree. In our experiment, we decide the number of views by the complexity of the scene, and we set the depth to 10 levels for SPSR. In addition, our VDVNet takes 993MB GPU RAM for visibility estimation (image size: 256x256).

E.4. Qualitative Evaluation of Visibility Classifiers

Figure 5 shows the qualitative evaluation of different visibility estimators and visually shows the result in Table 2 in our main paper. The improvement in classification accuracy is reflected in the sharpness of the reconstructed surface details.

Data	# of points	# of virtual views	Visibility estimation	Graph-based reconstruction	Overall (ours)	Normal estimation	SPSR	Peak RAM (ours)	Peak RAM (SPSR)
Object (Quantitative)	25K	∼30	∼3s	∼1s	∼4s	∼1s	∼3s	∼0.1GB	∼0.2GB
Indoor (Quantitative)	100K	∼50	∼8s	∼4s	∼12s	∼3s	∼6s	∼0.2GB	∼0.3GB
Outdoor (Quantitative)	500K	∼100	∼14s	∼13s	∼27s	∼4s	∼22s	∼0.8GB	∼1.8GB
YParc	2M	203	24s	152s	179s	15s	192s	3.4GB	5.2GB
UThammasat	5M	300	30s	278s	311s	38s	204s	4.3GB	5.9GB
Hotel	5M	340	38s	229s	272s	40s	208s	4.1GB	5.9GB
GSM Tower	3.3M	87	10s	92s	103s	23s	195s	4.2GB	5.1GB
Crossroad	2.6M	114	15s	71s	86s	12s	105s	3.2GB	4.6GB

Table 6. Computational efficiency comparison with SPSR. To note that our overall processing time including image rendering, visibility estimation, and graph-cut based reconstruction.



(a) Input (b) HPR (c) UNet (d) Ours

Figure 5. Qualitative comparison of visibility estimators. From top to bottom, the data is *Bull*, *Tiki*, *DSLR*, and *Room1*.

E.5. Extra Robust Evaluation

In Figure 6 we present extra results for robustness evaluation of noises (Figure 9 in the main paper).

E.6. Extra Qualitative Evaluation

We present extra results qualitatively in three scales, for each, small objects (Figure 7), indoor scenes (Figure 8), and large-scale outdoor datasets (Figure 9).

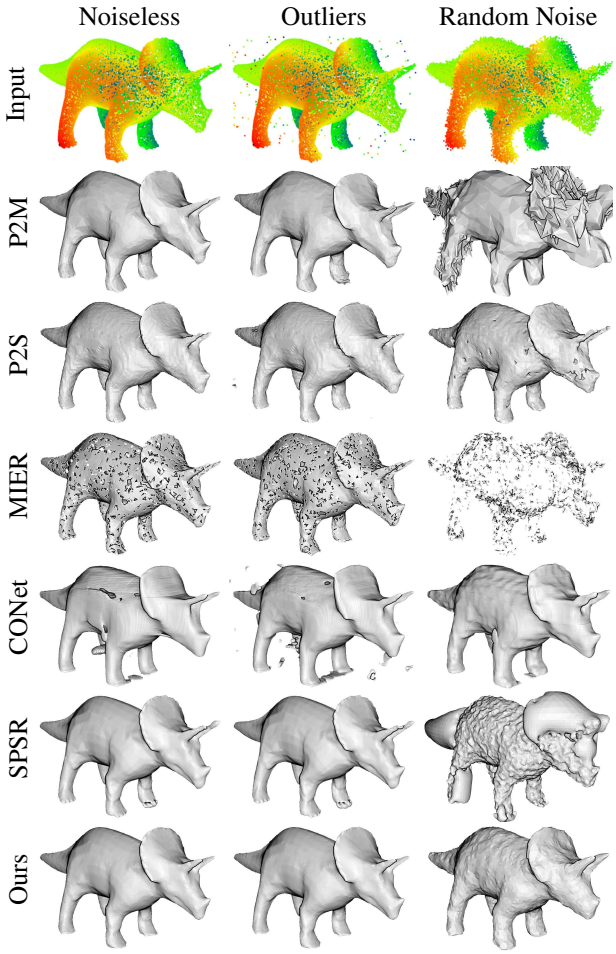


Figure 6. Evaluation of the methods subject to additional noises on the point clouds. The data is *Triceratops*.

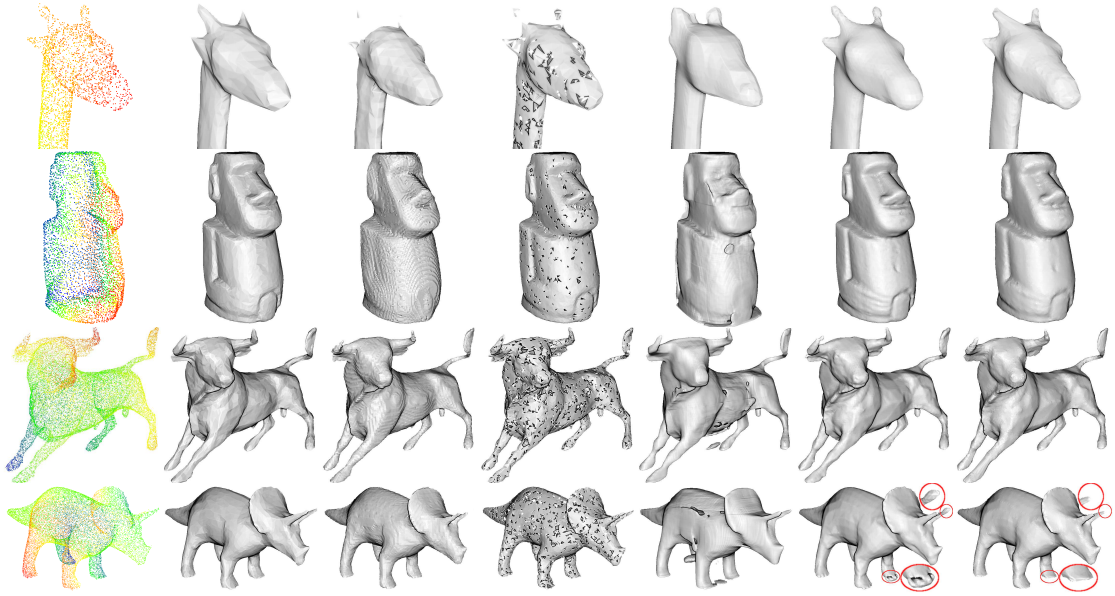
References

[1] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'77, page 659–663, San Francisco, CA, USA, 1977. Morgan Kaufmann Publishers Inc. 2

[2] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 2

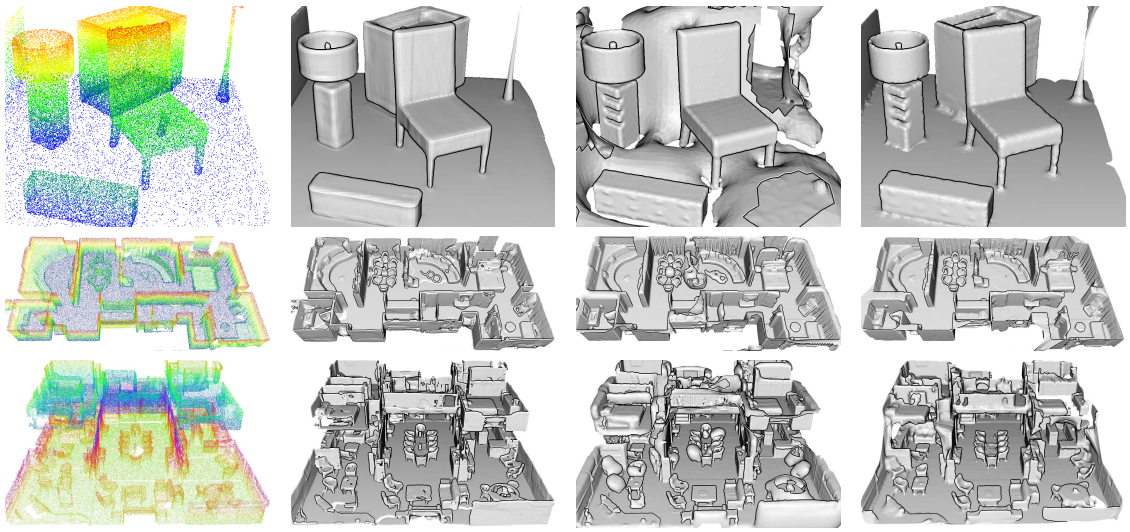
[3] Patrick Labatut, J-P Pons, and Renaud Keriven. Robust and efficient surface reconstruction from range data. In *Computer graphics forum*, volume 28, pages 2275–2290. Wiley Online Library, 2009. 1

[4] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. *Computer Vision and Pattern Recognition (CVPR)*, 2020. 1



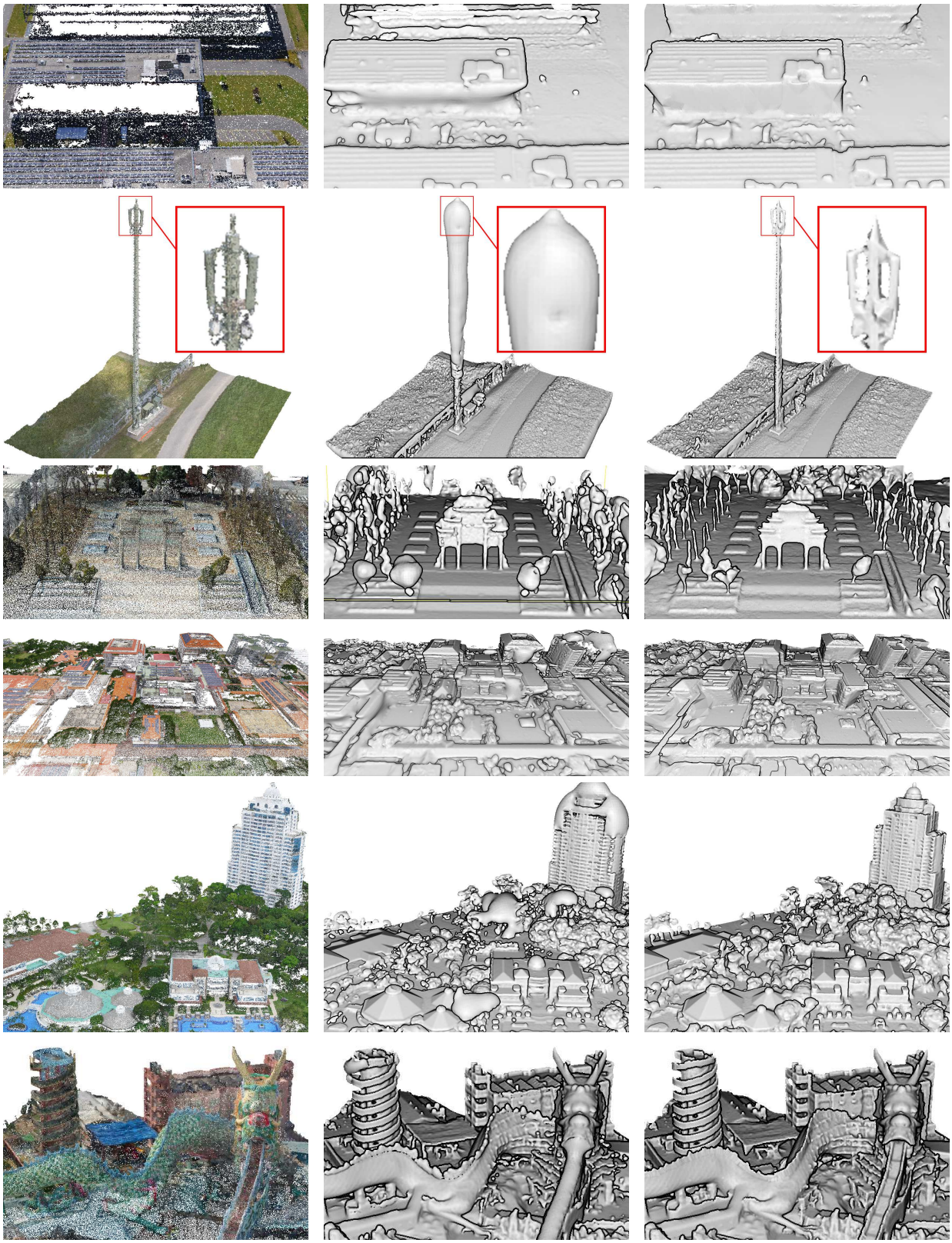
(a) Input (b) P2M (c) P2S (d) MIER (e) CONet (f) SPSR (g) Ours

Figure 7. Qualitative comparison on additional small object level datasets. From top to bottom, the data is *Giraffe*, *Tiki*, *Bull*, and *Triceratops*.



(a) Input Points (b) CONet (c) SPSR (d) Ours

Figure 8. Qualitative comparison on additional indoor datasets. From top to bottom, the data is *Room1*, *MPT:SingleFloor*, and *MPT:MultiFloor*.



(a) Input Points (b) SPSR (c) Ours

Figure 9. Qualitative comparison on additional large scale datasets. From top to bottom, the data is *YParc*, *GSM Tower*, *Archway*, *UThammasat*, *Hotel*, and *Dragon Park*.