R. Sulzer[1,2] L. Landrieu[1] R. Marlet[2,3] B. Vallet[1]

# Supplementary Material for: Scalable Surface Reconstruction with Delaunay-Graph Neural Networks

In this supplementary document, we first provide additional information about our our training data in Section 6 and implementation in Section 7. Finally, we provide additional qualitative and quantitative experimental results in Section 8 for object-level reconstruction, and in Section 9 for scene-level reconstruction.

## 6. Generating Training Data

In an ideal setting, we would have trained our network on real-life, large-scale, MVS acquisitions together with associated ground-truth surfaces. However, such surfaces are difficult to produce. Two methods can be used to circumvent this issue: using laser scans or resorting to synthetic scans.

**Laser Scans.** The first option is to use a surface reconstructed from a high-precision acquisition of a scene, e.g., with a stationary LiDAR scan. In parallel, the scene can be captured by cameras to produce an MVS acquisition, typically of lower quality. This procedure has been used in several MVS benchmarks [SvHG*08, KPZK17, SSG*17, SDSS06]. However, a difficulty remains when reconstructing the ground-truth surface. We require a closed surface to derive the ground-truth occupancy. The chosen surface reconstruction method may introduce biases in the ground-truth surface, such as over-smoothing. Additionally, even with high-quality LiDAR acquisitions, parts of the scene can be missing, e.g., due to occlusions. These issues ultimately lead to inconsistencies in the training data, because the MVS acquisition locally diverges from the ground-truth surface. Thus, in practice, we found that the incompleteness of available LiDAR scans makes this source of data too unreliable to train our network.

**Synthetic Scans.** A second option for producing ground-truth data is to use synthetic scans of closed artificial shapes. To this end, we make use of the range scanning procedure from the Berger *et al.* [BLN*13] benchmark for surface reconstruction.

We modified the provided code to export the camera positions of the scanning process. We then synthetically scan artificial shapes using our modified version of the Berger *et al.* scanning software. We choose at random one of the 5 scanner settings described in Table 4 to scan each training shape. The low resolution scanner setting produces uniform point clouds, similar to those obtained by coarse voxelizations. High resolution settings produce point clouds similar to those obtained by MVS. We also add outliers to the scans in the form of randomly distributed points in the bounding box of the objects and associate these points with a random camera position. We use this method to produce training data from a small subset of 10 shapes of each of the 13 classes of the ShapeNet subset from [CXG*16]. We produce watertight meshes of the ShapeNet models using the method of Huang *et al.* [HSG18].

To obtain the ground-truth occupancy, we sample 100 points in each tetrahedron and determine the percentage of these sampled points lying inside their corresponding ground-truth models. In total, we train our network on around 10M tetrahedra. We also apply the scanning procedure with the 5 different configurations to each shape of the 5 ground-truth shapes from the Berger *et al.* [BLN*13] benchmark. See Figure 8 for the 5 ground-truth shapes and the first column of Figures 9-12 for their scans. We refer the reader to the original benchmark paper [BLN*13] for further details about the scanning process.

## 7. Implementation Details

**Multi-View Stereo.** Our implementation relies on the OpenMVS [Cer15] library for many of the MVS processing steps.

We generate dense point clouds using the provided camera poses of all scenes of the ETH3D test dataset. We use the DensifyPointCloud tool of OpenMVS with standard settings, except for the following parameters: *number-views-fuse = 2*, *optimize = 0* and *resolution-level = 4*.

**Visibility-augmented 3DT.** We use CGAL to obtain the Delaunay Triangulation and for ray tracing. For the ray tracing, we only use one camera per point. We chose the camera minimizing the angle between the line-of-sight and the point's normal (obtained by local principal component analysis). In our experiments, this allows for a significant speed-up in the ray tracing step with a negligible difference on the predicted surface. Likewise, we disregard the third tetrahedron encountered after a line of sight traverses an observed point, and beyond (see Fig. 3).

**Deep Learning.** Finally, we use PyTorch [PGM*19] and PyTorch Geometric [FL19] for implementing the graph neural network training and inference.

**Binary Weights.** We use the same surface quality term $B_{s,t}(i_s, i_t) = \mathbb{1}(i_s \neq i_t)\beta_{s,t}$ as Labatut *et al.* [LPK09] for a facet interfacing the tetrahedra $s$ and $t$. Considering the intersection of the circumspheres of $s$ and $t$ with the facet, with angles $\phi$ and $\psi$, then $\beta_{s,t}$ is defined as:

$$\beta_{s,t} = 1 - \min\{cos(\phi), cos(\psi)\} \; . \tag{13}$$

**Parameterization of Competing Methods.** We use the OpenMVS implementations of Vu *et al.* and Jancosek *et al.* through the ReconstructMesh tool with *min-point-distance = 0.0*. For Vu *et al.* we set *free-space-support = 0*, and we set it to 1 for Jancosek *et al.* .

For the reconstructions of ConvONet we use the multi-plane decoder model pretrained on ShapeNet for object-level reconstruction and the volume decoder model pretrained on the synthetic indoor scene dataset [PNM*20] for scene-level reconstruction, where we set the voxel size to 4 *cm*.

**Table 4:** ***Scanning configuration for Berger et al.'s benchmark.*** *We show the five different scanner configurations used in our modified version of the Berger et al.'s scanning procedure. We use the resulting scans to evaluate object-level reconstruction with varying point-cloud defects and for training data generation. For the low resolution (LR) scans the scanning process results in* 1000 *to* 3000 *points per shape, and for the high resolution (HR), the scanning process yields around* 10 000 *to* 30 000 *points.*

| | Low res. (LR) | High res. (HR) | HR + noise (HRN) | HR + outliers (HRO) | HR + noise + outliers (HRNO) |
|---|---|---|---|---|---|
| Camera resolution x, y | 50, 50 | 100, 100 | 100, 100 | 100, 100 | 100, 100 |
| Scanner positions | 5 | 10 | 10 | 10 | 10 |
| Min/max range | 70/300 | 70/300 | 70/300 | 70/300 | 70/300 |
| Additive noise | 0 | 0 | 0.5 | 0 | 0.5 |
| Outliers (%) | 0 | 0 | 0 | 0.1 | 0.1 |

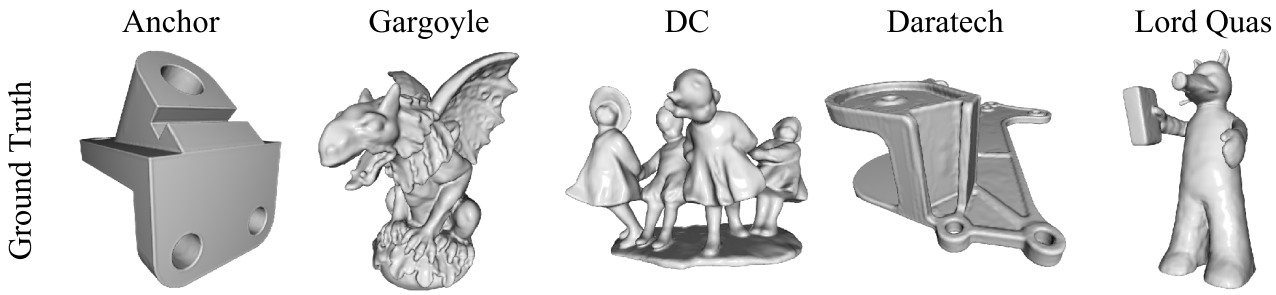



**Figure 8:** ***Ground-truth meshes for Berger et al.'s benchmark.*** *We represent the* 5 *shapes chosen from the Berger et al.'s benchmark [BLN*13] for our evaluation.*

**Cleaning of scene reconstruction.** We use default clean options in OpenMVS for the cleaning step for all scene-level mesh reconstructions.

## 8. Object-Level Reconstruction

**Metrics.** We evaluate object-level reconstruction with the volumetric IoU, the symmetric Chamfer distance, the number of connected components and the number of non-manifold edges in the reconstructed mesh.

For the Chamfer distance, we sample $n_S = 100\,000$ points on the ground-truth meshes $\mathcal{M}_\mathcal{G}$ and reconstructed meshes $\mathcal{M}_\mathcal{P}$. The distances between the resulting ground-truth point cloud $S_G$ and the reconstruction point cloud $S_P$, approximating the two-sided Chamfer distance, is then given as:

$$d_{CD}(\mathcal{M}_\mathcal{G},\mathcal{M}_\mathcal{P}) = \frac{1}{n_S}\sum_{x\in S_G}\min_{y\in S_P}||x-y||_2^2 + \frac{1}{n_S}\sum_{y\in S_P}\min_{x\in S_G}||y-x||_2^2 \tag{14}$$

The volumetric IoU is defined as:

$$\mathrm{IoU}(\mathcal{M}_\mathcal{G},\mathcal{M}_\mathcal{P}) = \frac{|\mathcal{M}_\mathcal{G}\cap\mathcal{M}_\mathcal{P}|}{|\mathcal{M}_\mathcal{G}\cup\mathcal{M}_\mathcal{P}|}, \tag{15}$$

We approximate the volumetric IoU by sampling 100 000 points in the union of the bounding boxes of the ground-truth and reconstruction meshes.

For the number of connected components, we count all components of the reconstructed meshes. The ground-truth meshes all have only one component. Additionally, they do not have any non-manifold edges.

**Additional Qualitative Results.** The main paper provides both quantitative results over the whole dataset (see Table 1) and qualitative results for one object (see Fig. 6). Figures 9-12 show the results for all the other objects.

## 9. Large-scale Scene Reconstruction

**Metrics.** For the large-scale benchmark ETH3D, we evaluate the mesh reconstruction methods at a given precision $\tau$ using the the Accuracy (precision) $P(\tau)$, the Completeness (recall) $R(\tau)$, and the F1-Score $F(\tau)$, defined as their harmonic mean:

$$F(\tau) = \frac{2P(\tau)R(\tau)}{P(\tau)+R(\tau)} \tag{16}$$

We use the ETH3D Evaluation Program [SSG*17] to compute these values from the ground-truth LiDAR scans and samplings of the meshed surfaces. In the original benchmark, the authors evaluate MVS reconstructions with threshold $\tau$ as low as 1 cm. Generating such mesh samplings implies sampling over 300 million points for some scenes. To accelerate this procedure, we only sample 900

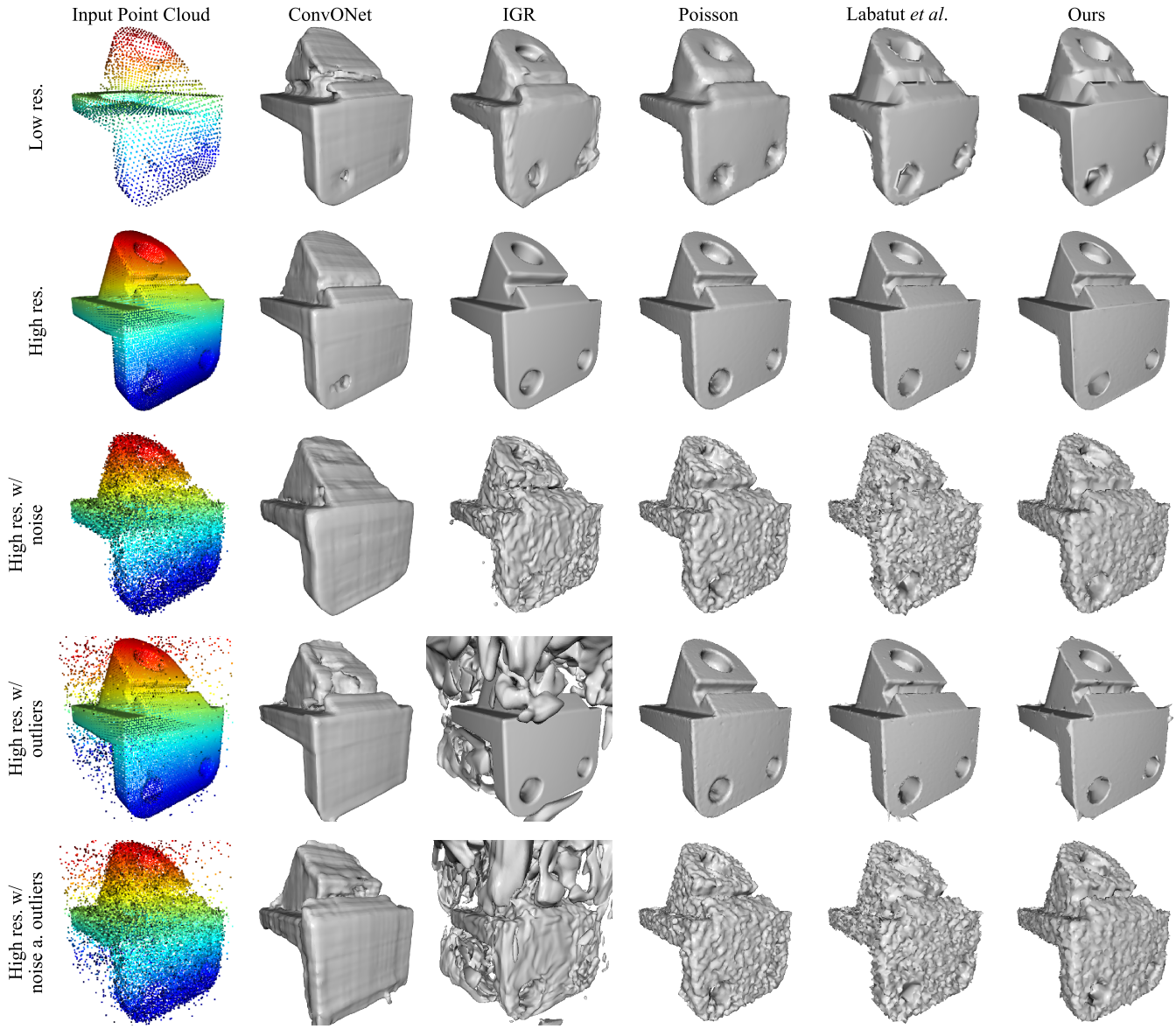



**Figure 9:** ***Reconstruction of the*** **Anchor** ***object in the surface reconstruction benchmark of Berger et al. [BLN*13].*** *We show the input point clouds in column 1. ConvONet [PNM*20] (column 2) does not generalize well to the unseen new shape. IGR [GYH*20] (column 3) works well at high resolution but fails in the other cases. The Screened Poisson [KH13] algorithm (column 4) does not reconstruct the sharp features well, but is robust against outliers, even close to the surface. The reconstructions of Labatut et al. [LPK09] (column 5) and ours (column 6) are visually similar for the easier high resolution case. Our method performs slightly better on the low resolution, and noise cases.*

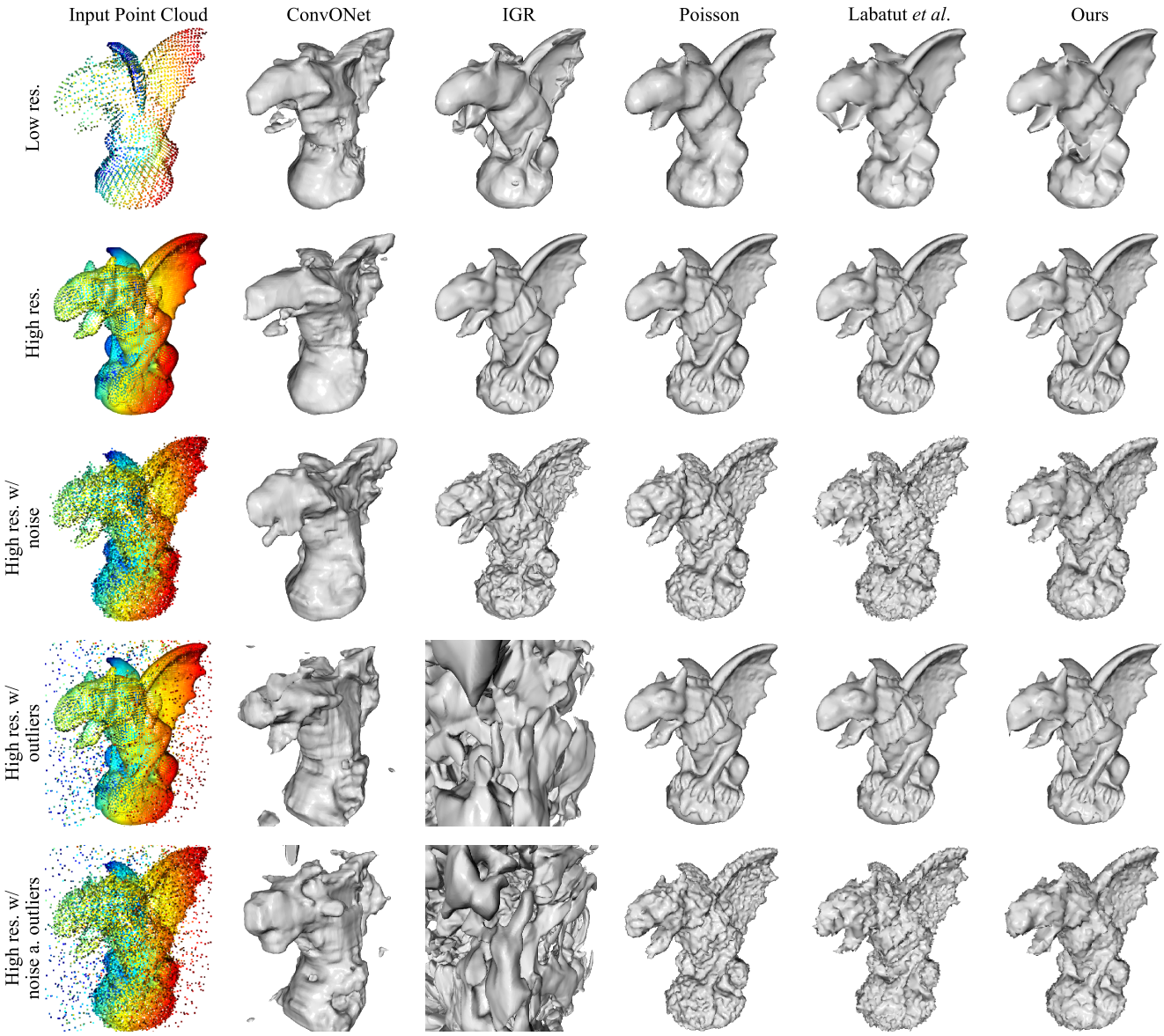



**Figure 10:** ***Reconstruction of the*** **Gargoyle** ***object in the surface reconstruction benchmark of Berger et al. [BLN*13].*** *We show the input point clouds in column 1. ConvONet [PNM*20] (column 2) does not generalize well to the unseen new shape. IGR [GYH*20] (column 3) generates many surface components from outliers. The Screened Poisson [KH13] algorithm (column 4) does not reconstruct the sharp features well, but is robust against outliers, even close to the surface. The reconstructions of Labatut et al. [LPK09] (column 5) and ours (column 6) are visually similar for the easier high resolution case. While both methods are very robust against outliers, our method performs slightly better on the low resolution, outlier and noise cases.*

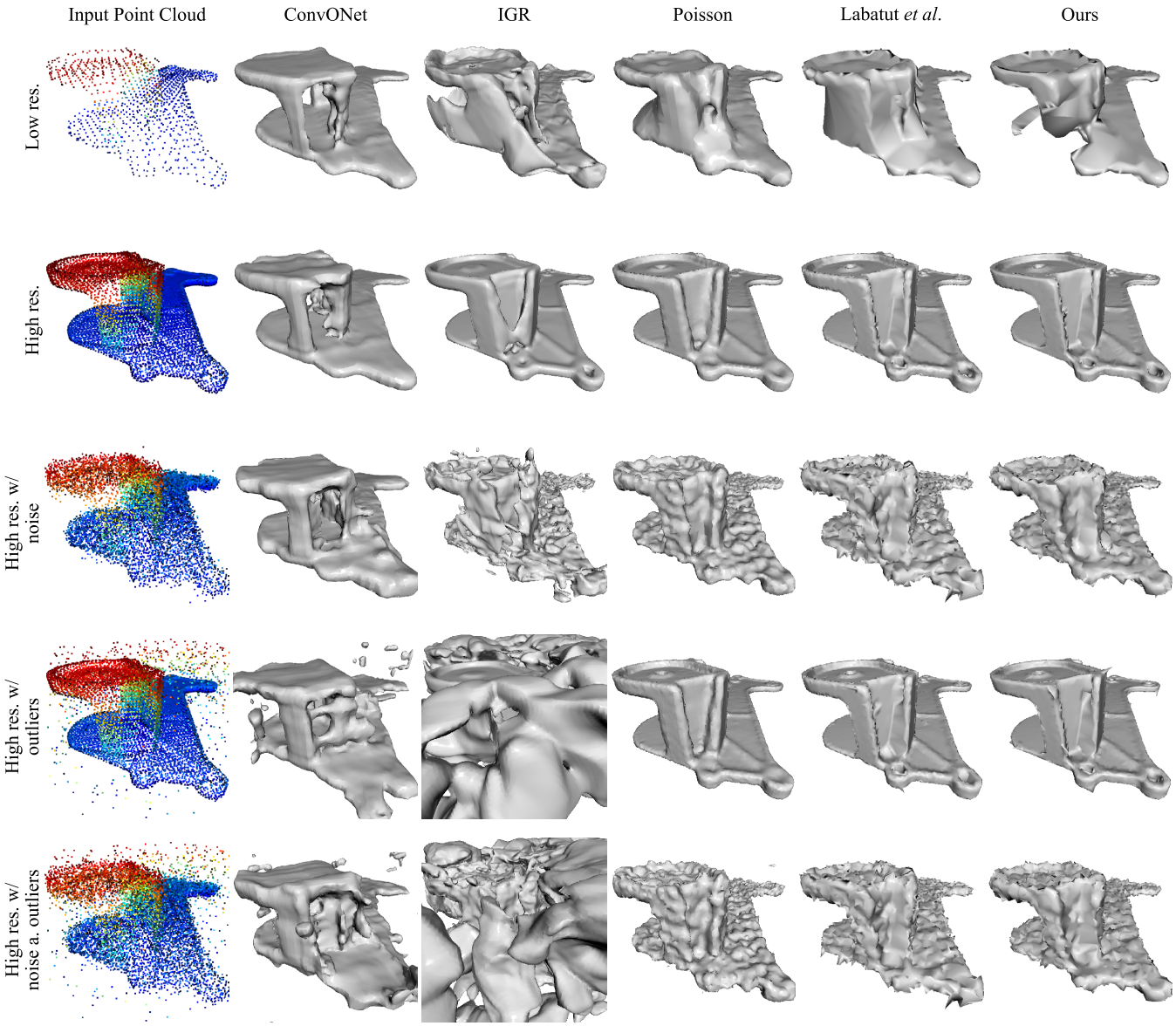



**Figure 11:** ***Reconstruction of the*** **Daratech** ***object in the surface reconstruction benchmark of Berger et al. [BLN*13].*** *We show the input point clouds in column 1. ConvONet [PNM*20] (column 2) does not generalize well to the unseen new shape. As with other shapes, IGR [GYH*20] (column 3) works well at high resolution but generates artefacts or fails in other settings. The Screened Poisson [KH13] algorithm (column 4) does not reconstruct the sharp features well, but is robust against outliers, even close to the surface. In the low resolution setting, our algorithm is incomplete where Labatut creates unwanted surface parts.*

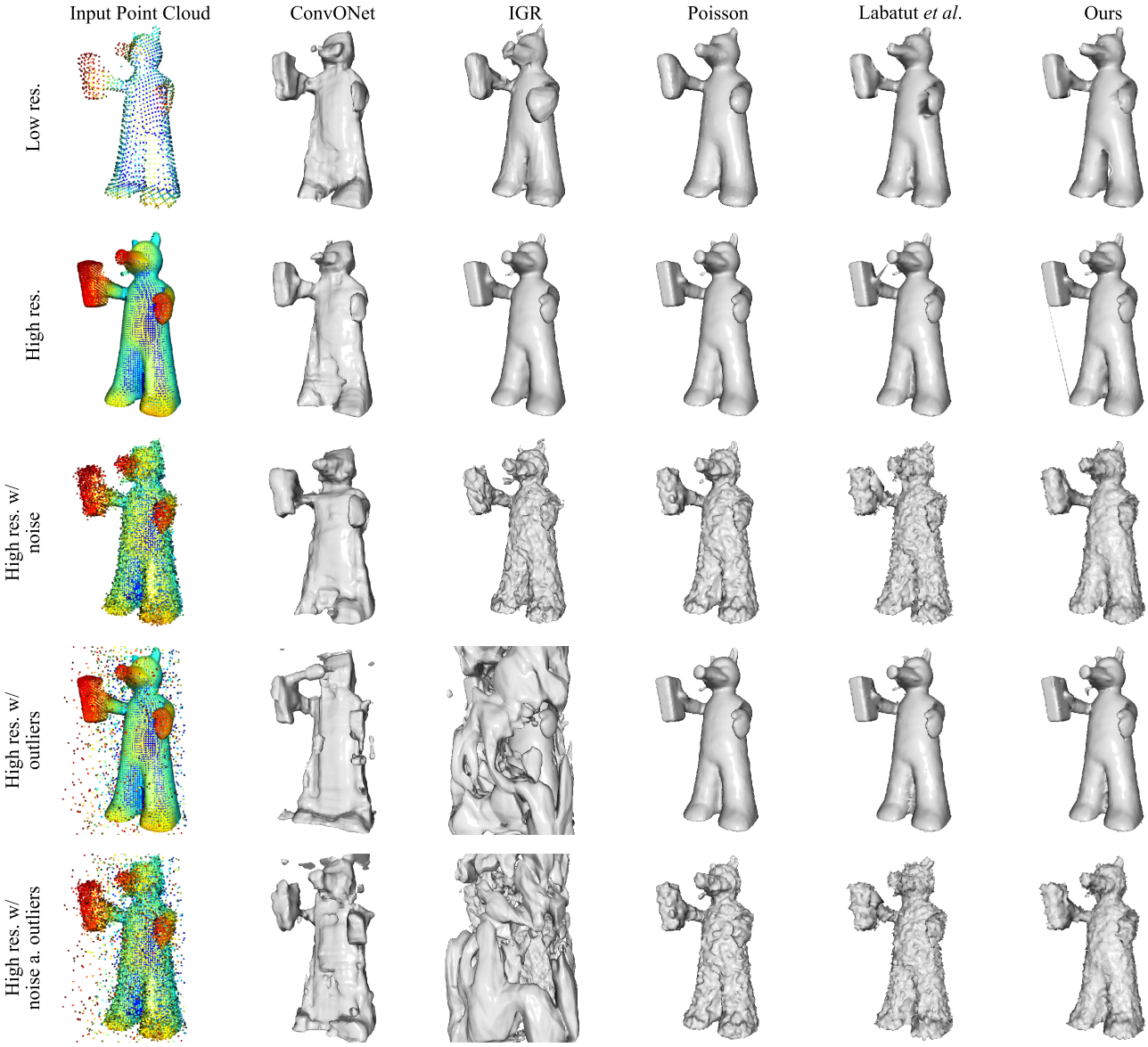



**Figure 12:** ***Reconstruction of the*** **Quasimoto** ***object in the surface reconstruction benchmark of Berger et al. [BLN*13].*** *We show the input point clouds in column 1. ConvONet [PNM*20] (column 2) does not generalize well to the unseen new shape. IGR [GYH*20] (column 3) is not able to filter outliers in the scan. The Screened Poisson [KH13] algorithm (column 4) does not reconstruct the sharp features well. The reconstructions of Labatut et al. [LPK09] (column 5) and ours (column 6) are visually similar for the defect-free cases. Both methods produce small artifacts in the high resolution case: between the book and nose for Labatut et al. [LPK09] and between the book and left foot for ours. Both methods are very robust against outliers.*

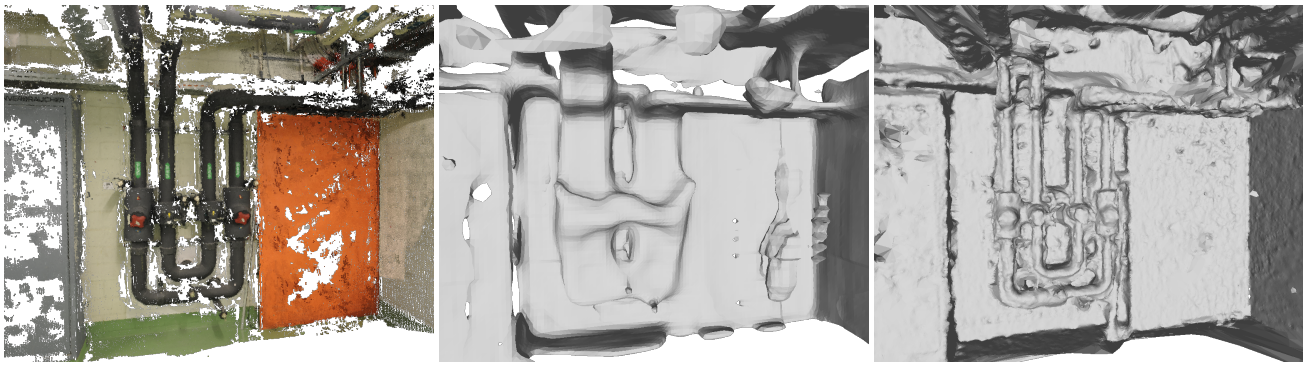
| | F1-score - uncleaned mesh | | | | F1-score - cleaned mesh | | | |
|---|---|---|---|---|---|---|---|---|
| scene | Poisson | Vu et al. | Jan. et al. | Ours | Poisson | Vu et al. | Jan. et al. | Ours |
| kicker | 0.75 | **0.79** | 0.75 | 0.76 | 0.75 | **0.81** | 0.78 | 0.78 |
| pipes | 0.77 | **0.79** | 0.77 | 0.76 | 0.77 | **0.78** | 0.77 | 0.75 |
| delivery_area | 0.69 | 0.70 | 0.66 | **0.71** | 0.69 | 0.70 | 0.68 | **0.71** |
| meadow | 0.45 | 0.52 | 0.51 | **0.58** | 0.40 | 0.50 | 0.50 | **0.60** |
| office | 0.60 | **0.65** | 0.59 | 0.59 | 0.60 | **0.64** | 0.62 | 0.58 |
| playground | 0.61 | **0.70** | 0.63 | **0.70** | 0.60 | 0.69 | 0.66 | **0.73** |
| terrains | 0.73 | **0.78** | 0.76 | 0.75 | 0.74 | **0.78** | 0.77 | 0.76 |
| terrace | 0.79 | 0.76 | 0.74 | **0.83** | 0.79 | 0.79 | 0.78 | **0.85** |
| relief | 0.72 | 0.67 | 0.64 | **0.80** | 0.73 | 0.69 | 0.67 | **0.80** |
| relief_2 | 0.70 | 0.68 | 0.67 | **0.79** | 0.71 | 0.70 | 0.70 | **0.78** |
| electro | 0.65 | 0.64 | 0.60 | **0.68** | 0.65 | 0.65 | 0.64 | **0.69** |
| courtyard | 0.76 | 0.75 | 0.72 | **0.77** | 0.75 | 0.75 | 0.74 | **0.77** |
| facade | 0.50 | 0.52 | 0.50 | **0.53** | 0.51 | **0.55** | 0.54 | 0.50 |
| mean | 0.67 | 0.69 | 0.66 | **0.71** | 0.67 | 0.69 | 0.68 | **0.71** |

**Table 5:** ***Detailed quantitative results on ETH3D.*** *F1-score of all scenes of the train dataset of ETH3D [SSG*17] for uncleaned and cleaned mesh reconstructions at distance $\tau = 5$ cm. The best (highest) values per scene are in bold. We perform better than all competing methods on 8 scenes out of 13. On average, our method performs between 2 and 5% better than the competing methods, and improve the F1-score for 8 out of 13 scenes. The mesh cleaning only improves the F1-score of the reconstruction of Jancosek et al. [JP14].*

points per $m^2$ on the reconstructed meshes. This allows us to compute accuracy and completeness with a threshold of 5 cm and up.

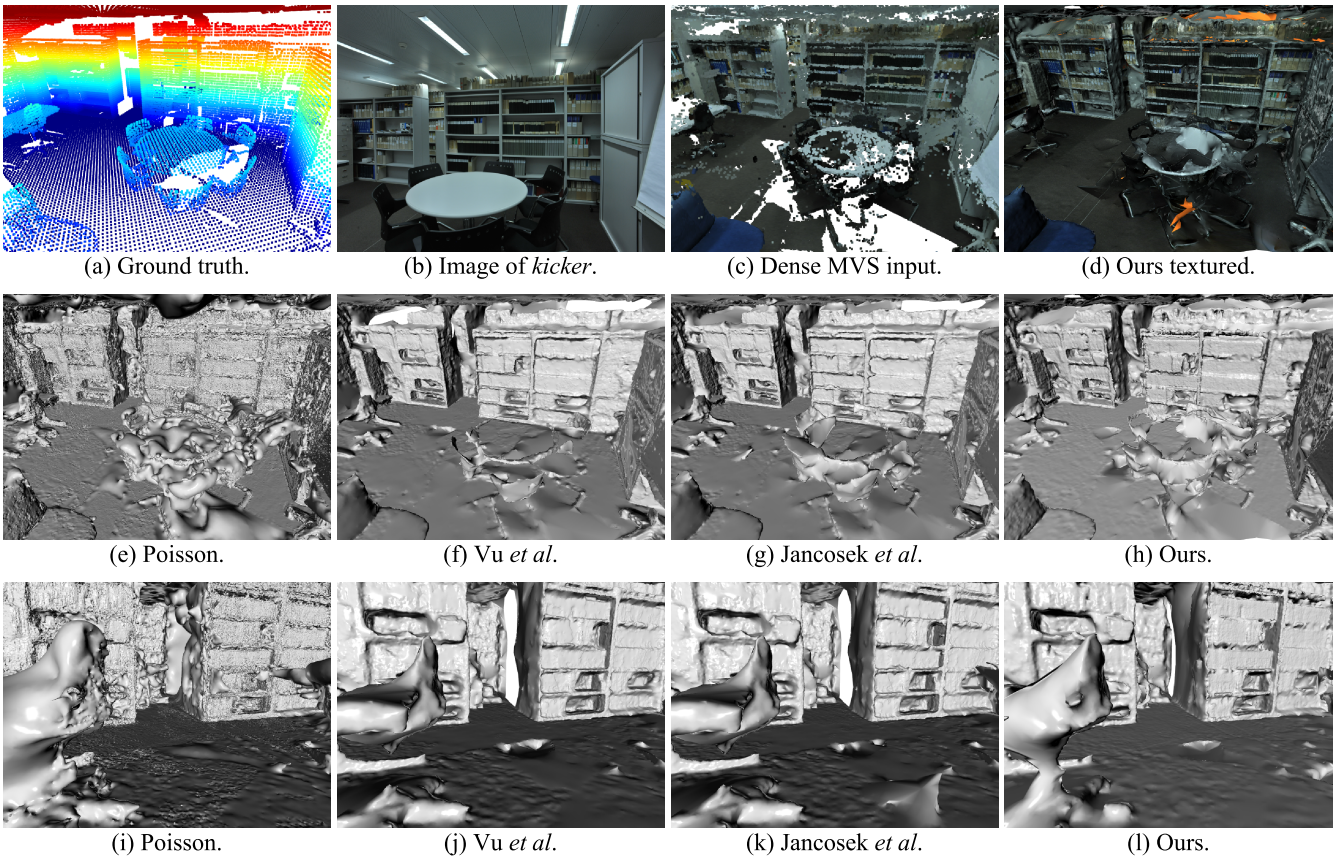
**Detailed quantitative Results** In Table 5, we show the F1-Score at $\tau = 5$ cm of all 13 scenes of the ETH3D dataset for both uncleaned and cleaned mesh reconstructions. Our method produces the best reconstruction scores for 9 out of 13 scenes. Mesh cleaning did not significantly alter the scores as it resulted in less complete but more accurate reconstructions.

**Qualitative Results.** We show an example of a locally more accurate reconstruction of our method compared to our competitors in Figure 13 and Figure 14. We show in Figure 15 the effect of the cleaning step on a hard problem due to a large amount of noise and outliers. Finally, we also show an example of our method producing a less complete reconstruction in Figure 16.



**(a)** *Dense MVS input.* **(b)** *ConvONet.* **(c)** *Ours.*

**Figure 13:** ***Indoor ETH3D reconstruction.*** *Reconstruction of the* pipes *scene of the ETH3D benchmark [SSG*17]. We show the dense MVS point cloud in (a), the mesh reconstructions obtained by ConvONet [PNM*20] in (b) and our proposed reconstruction in (c). Similar to object-level reconstruction, ConvONet does not generalize well to the unseen new shapes in this scene. Our learning algorithm, operating purely locally, is able to reconstruct the pipes and fill all holes in the point cloud acquistion.*



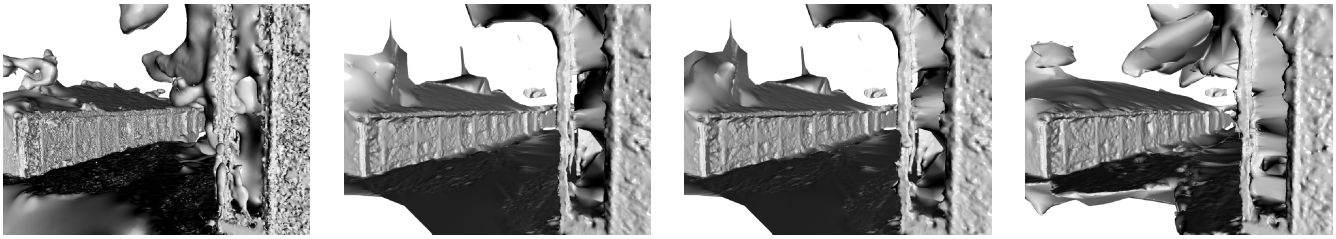
(a) Ground truth. (b) Image of *kicker*. (c) Dense MVS input. (d) Ours textured.

(e) Poisson. (f) Vu *et al.* (g) Jancosek *et al.* (h) Ours.

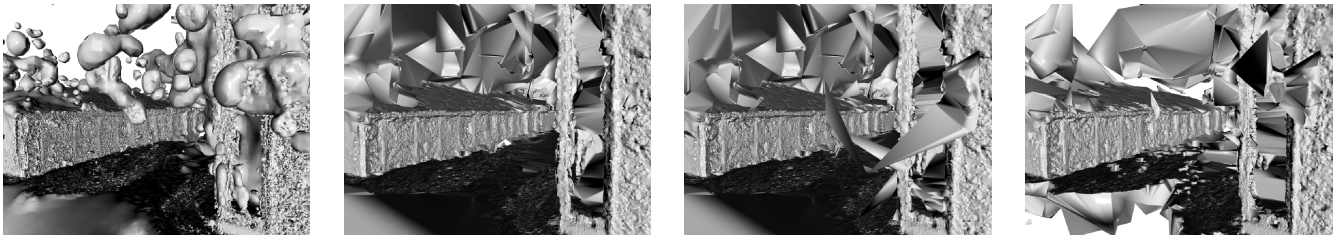
(i) Poisson. (j) Vu *et al.* (k) Jancosek *et al.* (l) Ours.

**Figure 14:** ***Indoor ETH3D reconstruction.*** *Reconstruction of the* kicker *scene of the ETH3D benchmark [SSG*17]. We show the ground truth that is used for evaluation in (a). A set of images, such as the one represented in (b), is transformed into a dense MVS point cloud (c), from which a mesh can be reconstructed and textured [WMG14], as shown in (d) with our proposed mesh reconstruction. We show the untextured mesh reconstructions obtained by the screened Poisson algorithm in (e,i), the algorithms of Vu et al. [VLPK12] in (f,j) and of Jancosek et al. [JP14] in (g,k), and finally our proposed reconstruction in (h,l). All methods struggle to reconstruct the table and the chairs, that have little data support.*

(a) Ground truth. (b) Image of *meadow*. (c) Dense MVS input. (d) Ours textured.

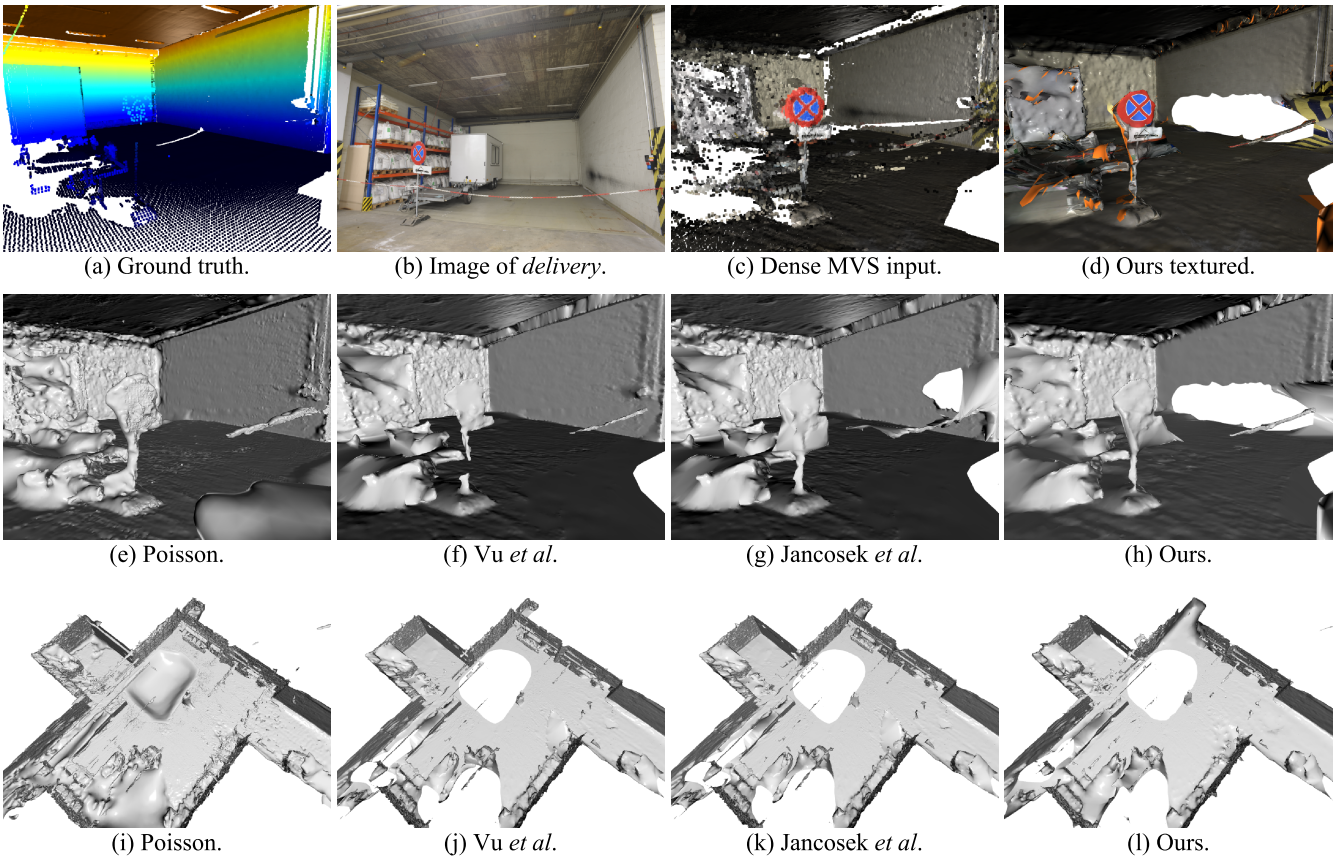


(e) Poisson. (f) Vu *et al*. (g) Jancosek *et al*. (h) Ours.



(i) Poisson uncleaned. (j) Vu *et al* uncleaned. (k) Jancosek *et al* uncleaned. (l) Ours uncleaned.

**Figure 15:** ***Outdoor ETH3D reconstruction.*** *Reconstruction of the* meadow *scene of the ETH3D benchmark [SSG*17]. We show the ground truth that is used for evaluation in (a). A set of images, such as the one represented in (b), is transformed into a dense MVS point cloud (c), from which a mesh can be reconstructed and textured [WMG14], as shown in (d) with our proposed mesh reconstruction. We show the untextured mesh reconstructions obtained by the screened Poisson algorithm in (e,i), the algorithms of Vu et al. [VLPK12] in (f,j) and of Jancosek et al. [JP14] in (g,k), and finally our proposed reconstruction in (h,l). Trees and outliers in the sky lead to a large number of isolated components in all mesh reconstructions. Most of these small components can be removed with the heurestic mesh cleaning step that we apply as post-processing.*



(a) Ground truth. (b) Image of *delivery*. (c) Dense MVS input. (d) Ours textured.

(e) Poisson. (f) Vu *et al*. (g) Jancosek *et al*. (h) Ours.

(i) Poisson. (j) Vu *et al*. (k) Jancosek *et al*. (l) Ours.

**Figure 16:** ***Failure case on ETH3D.*** *Reconstruction of the* delivery area *scene of the ETH3D benchmark [SSG*17]. We show the ground truth that is used for evaluation in (a). A set of images, such as the one represented in (b), is transformed into a dense MVS point cloud (c), from which a mesh can be reconstructed and textured [WMG14], as shown in (d) with our proposed mesh reconstruction. We show the untextured mesh reconstructions obtained by the screened Poisson algorithm in (e,i), the algorithms of Vu et al. [VLPK12] in (f,j) and of Jancosek et al. [JP14] in (g,k), and finally our proposed reconstruction in (h,l). Our method does not close the wall on the right, but performs slightly better on reconstructing the no-parking sign. Yet, considering the whole scene, the holes we create do not cover a larger area than other methods.*