PC2WF: 3D WIREFRAME RECONSTRUCTION FROM RAW POINT CLOUDS – SUPPLEMENTARY MATERIAL

Yujia Liu, Stefano D'Aronco, Konrad Schindler, Jan Dirk Wegner
EcoVision Lab, Photogrammetry and Remote Sensing, ETH Zürich
{firstname.lastname}@geod.baug.ethz.ch

In this supplementary document we provide further details regarding the network architecture, post-processing operations, training configurations, dataset details, as well as further experimental results.

1 IMPLEMENTATION DETAILS

1.1 NETWORK ARCHITECTURE

We use the FCGF feature extractor (Choy et al., 2019b) as our backbone, which has a U-Net architecture (Ronneberger et al., 2015), skip connections and ResNet blocks (He et al., 2016). It is implemented with an auto-differentiation library, the Minkowski Engine (Choy et al., 2019a), that provides sparse versions of convolutions and other deep learning layers.

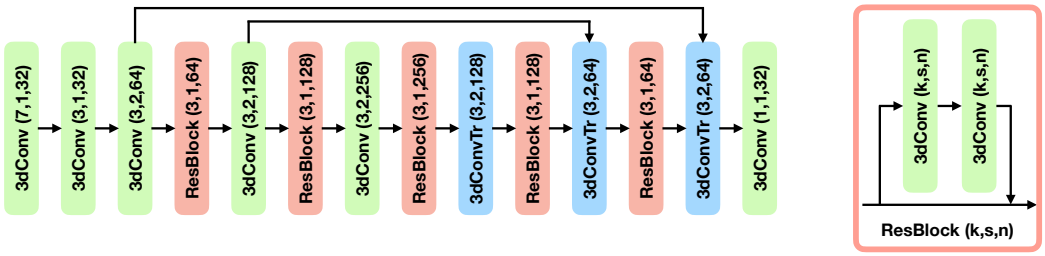

Figure 1: Network architecture of the FCGF backbone (adapted from Choy et al. (2019b)). All 3D convolution layers except for the last layer include batch normalisation and ReLU activation. Numbers are kernel size, stride, and channel dimension.

1.2 POST-PROCESSING

Since the patches generated during inference may overlap, there may be redundant vertex predictions, which would cause redundant edges. To avoid this problem, we use non-maximum suppression (NMS) and remove redundant edges: for each predicted edge $e_{\tilde{v}_i,\tilde{v}_j}$, we find all the edges $e_{\tilde{v}_k,\tilde{v}_l}$ satisfying the following conditions: $\min_{\tilde{v}_k\in\tilde{\mathcal{V}}} \|\tilde{v}_i - \tilde{v}_k\|_2 + \min_{\tilde{v}_l\in\tilde{\mathcal{V}}} \|\tilde{v}_j - \tilde{v}_l\|_2 < \eta_{\text{nms}}, e_{\tilde{v}_k,\tilde{v}_l} \in \tilde{\mathcal{E}}$, where $\tilde{\mathcal{V}}$ and $\tilde{\mathcal{E}}$ are the set of predicted vertices and detected edges, respectively. Among the edges that fulfil the above conditions, we retain the one with the maximum probability and remove all others. See Fig. 2(a) and (b) for a visual example.

Sometimes almost collinear vertices may be present in the prediction result, which results in "bending" edges or "thin triangles" shown in Fig. 2(c) and (d). We "straighten" the edges by removing vertices lying on/near detected edges and directly connecting the other two end points (Fig. 2(e)). Fig. 2(f) and (g) shows the results before and after our post processing steps. We include an ablation study regarding the different post-processing techniques (NMS and straightening) in Table. 1. As we penalise double-predicted lines when calculating sAP, the results without NMS are much worse. Both methods contribute to improving the performance.

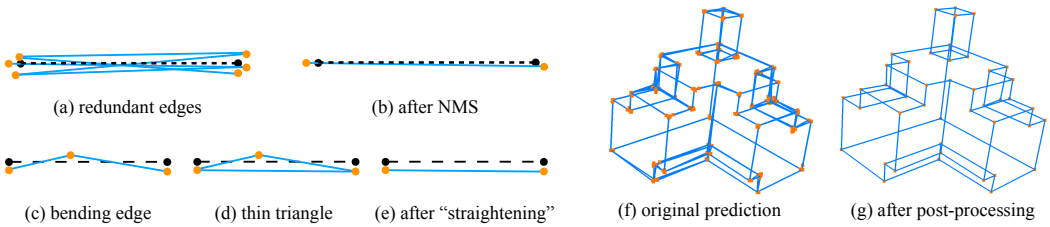

Figure 2: Illustration of post processing.

Table 1: Ablation study on post-processing

	$\text{sAP}^{0.03}$	$\text{sAP}^{0.05}$	$\text{sAP}^{0.07}$	msAP
no NMS	0.489	0.495	0.498	0.494
no straighten	0.776	0.810	0.821	0.802
NMS+straighten	0.868	0.898	0.907	0.891

1.3 Training Details

We fine-tune our PC2WF network starting from a pre-trained FCGF and optimise the parameters using ADAM Kingma & Ba (2014) with initial learning rate 0.001. The network is trained for 10 epochs and the learning rate is then decreased by a factor of two 2. We normalise each input point cloud to $[0, 1]$ and set the weights in the loss function to $\alpha = 10$ and $\beta = 1$.

2 Datasets

As mentioned in the main manuscript, we collect our own furniture dataset from the web. For the vertex and edge annotations, we use the Google SketchUp software and change the face style to wireframe mode to get an object's edges and their intersections. The detailed statistics of the dataset are presented in Tab. 2, some examples of original 3D objects are shown in Fig. 3.

Table 2: Statistics of furniture dataset

	bed	chair/sofa	table	monitor	stairs	**average**
#models	57	59	58	38	38	250 in total
#points	190035.5	185966.3	150666.1	178835.7	152399.1	172509.2
#vertices	32.9	38.8	34.3	29.3	62.7	38.6
#edges	50.0	59.5	52.9	45.8	96.5	59.3

3 Choice of Positive and Negative Edge Samples

In order to better understand and motivate the choice for the positive and negative examples sets used to train of the edge detector, we conduct additional experiments with different combinations, see Tab. 3. In each ablation experiment, we removed some subsets from the full set combination, and trained the whole network.

We observe that msAP drops to 0.760 if only the ground truth samples are used ($\mathcal{E}^{\text{gt}+}$ and $\mathcal{E}^{\text{gt}-}$). This means that the removal of training samples obtained from *predicted* vertices lowers the performance of the network (msAP for the full sets combination is 0.891). This is due to the fact that, by using only ground truth vertices, we create a shift in the distribution of input vertices between the training and the inference phase. The model trained only on ground truth vertices apparently does not generalise

Figure 3: Examples of objects used for virtual scanning in the furniture dataset.

Table 3: Ablation study of PC2WF. The first column represents that what sets are used for edge detector.

	$\text{sAP}^{0.03}$	$\text{sAP}^{0.05}$	$\text{sAP}^{0.07}$	msAP
$\mathcal{E}^{\text{gt+}}$, $\mathcal{E}^{\text{gt-}}$	0.715	0.777	0.788	0.760
$\mathcal{E}^{\text{pred+}}$, $\mathcal{E}^{\text{pred-}}$	0.651	0.773	0.795	0.740
$\mathcal{E}^{\text{gt+}}$, $\mathcal{E}^{\text{pred+}}$	0.525	0.611	0.622	0.586
$\mathcal{E}^{\text{gt+}}$, $\mathcal{E}^{\text{gt-}}$, $\mathcal{E}^{\text{pred+}}$, $\mathcal{E}^{\text{pred-}}$	**0.868**	**0.898**	**0.907**	**0.891**

well enough, and fails during inference. In the next experiment we remove the vertices based on ground truth and use *only* the sets created from the predicted vertices ($\mathcal{E}^{\text{pred}+}$ and $\mathcal{E}^{\text{pred}-}$). The results show that in this case the performance drop is even larger, msAP decreases to 0.740. We speculate that, during the initial training stage, the vertex detector/localiser finds very few vertices (positive samples), making a training of the edge detection module very difficult. The samples based on ground truth ensure there are enough valid vertex examples and mitigate this problem, thus improving the learning of the network parameters.

We also observe that training the network only on the positive set ($\mathcal{E}^{\text{gt}+}$ and $\mathcal{E}^{\text{pred}+}$) results in even worse performance (msAP: 0.586), which indicates that including well-selected negative edge samples is helpful for the training.

4 ADDITIONAL RESULTS

4.1 QUANTITATIVE RESULTS ON FURNITURE DATASET

We provide additional results for vertex (Fig. 4) and edge (Fig. 5) detection on the furniture dataset for different patch sizes. Our method performs similarly well on the furniture dataset and on the ABC dataset. Moderate patch sizes, 20 ~50, achieve the best performance on both datasets.

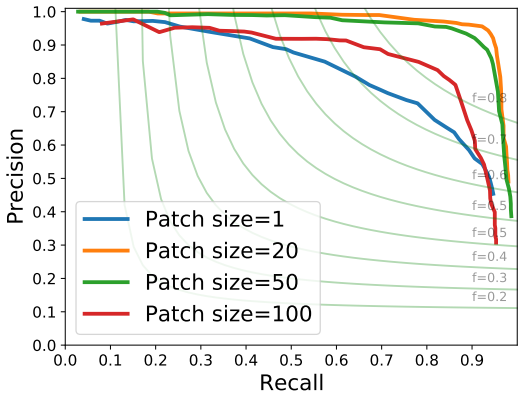

Figure 4: Vertex prediction accuracy on furniture dataset.

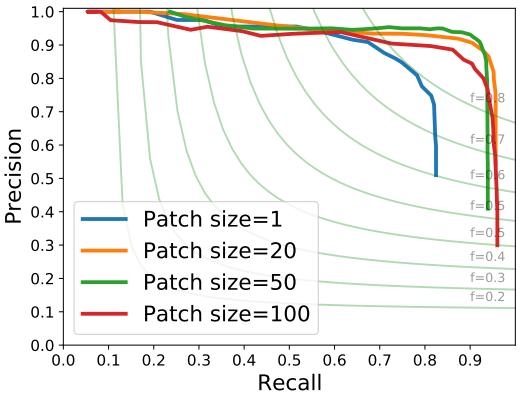

Figure 5: Edge detection accuracy on furniture dataset.

4.2 POINT CLOUD NOISE LEVELS

In order to investigate how the noise level affects the algorithm performance we train and test our method on point clouds with different amounts of noise. To do this we perturb the synthetic point cloud with zero mean Gaussian noise with different values of standard deviation ($\sigma_{\text{noise}} = 0, 0.01, 0.02, 0.05$). Table. 4 summarises the results of the experiment for different noise levels, as we can observe the noise level does not seem to affect the performance substantially. The model trained with a higher noise level is more sensitive to the threshold η_w for the sAP (defined in Section 4.1). In the case of $\sigma_{\text{noise}} = 0.05$, a relatively small η_w value (0.03) leads to a less accurate result. Visual results are shown in Fig. 9. As expected the quality of the wireframe prediction tends to decrease as the noise level increases. This effect, however, strongly depends on the size of the object details: the upper part of the object, which has no small structural details, is reconstructed correctly for all the noise levels, whereas the smaller details on the lower part are not predicted as accurately when the noise gets larger.

Table 4: PC2WF behavior with respect to noise levels

σ_{noise}	$\text{sAP}^{0.03}$	$\text{sAP}^{0.05}$	$\text{sAP}^{0.07}$	msAP
0	0.937	0.951	0.954	0.947
0.01	0.868	0.898	0.907	0.891
0.02	0.744	0.807	0.815	0.789
0.05	0.357	0.648	0.705	0.570

4.3 VISUALISATION

We show additional qualitative results, for both the ABC dataset (Fig. 6, 7, 8) and the furniture dataset (Fig. 10), with vertices in orange. Our method can successfully reconstruct wireframes from noisy point clouds. There are of course slight deviations between the predicted vertex positions and the ground truth, but they are imperceptibly small.

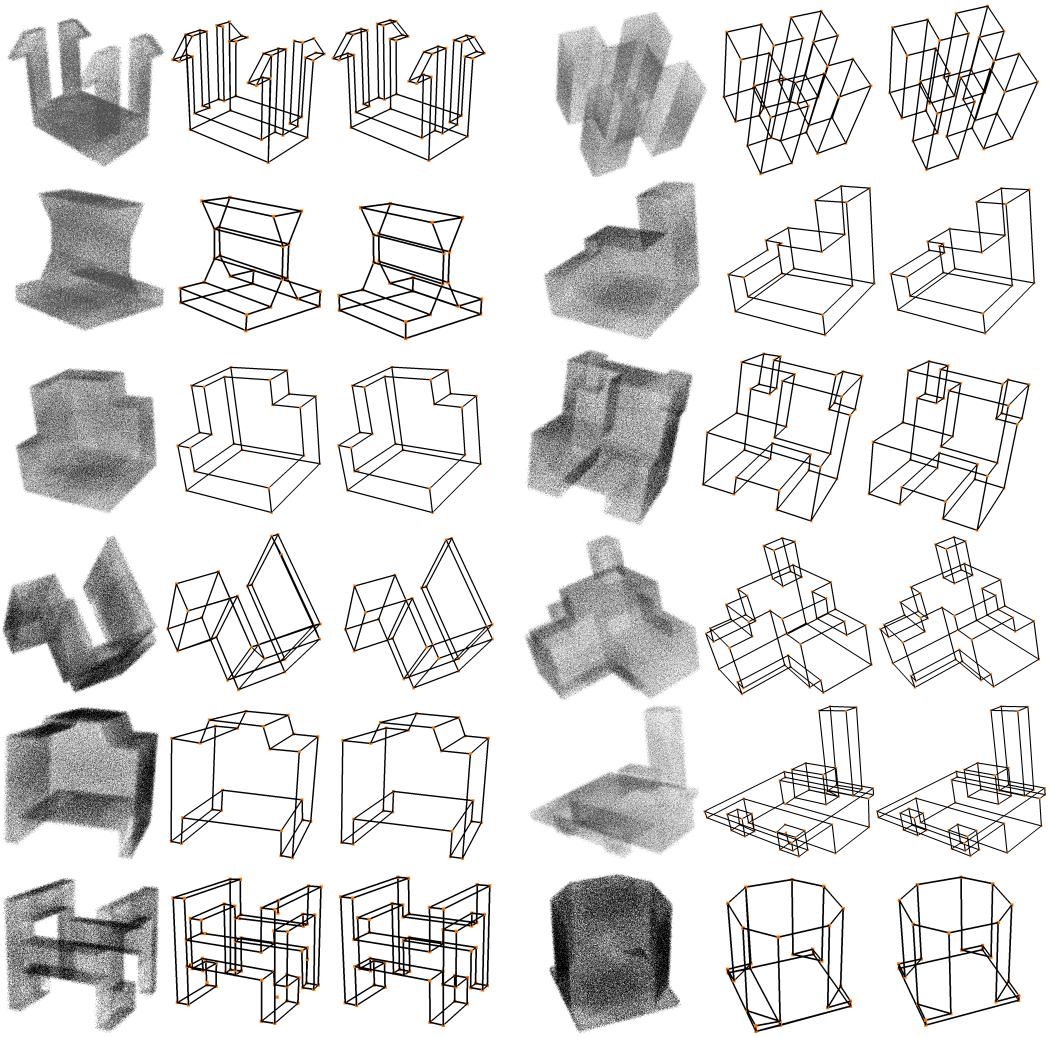

Figure 6: Wireframe reconstruction results on ABC dataset. (a)(d) input raw point clouds; (b)(e) predicted wireframes; (c)(f) ground truth

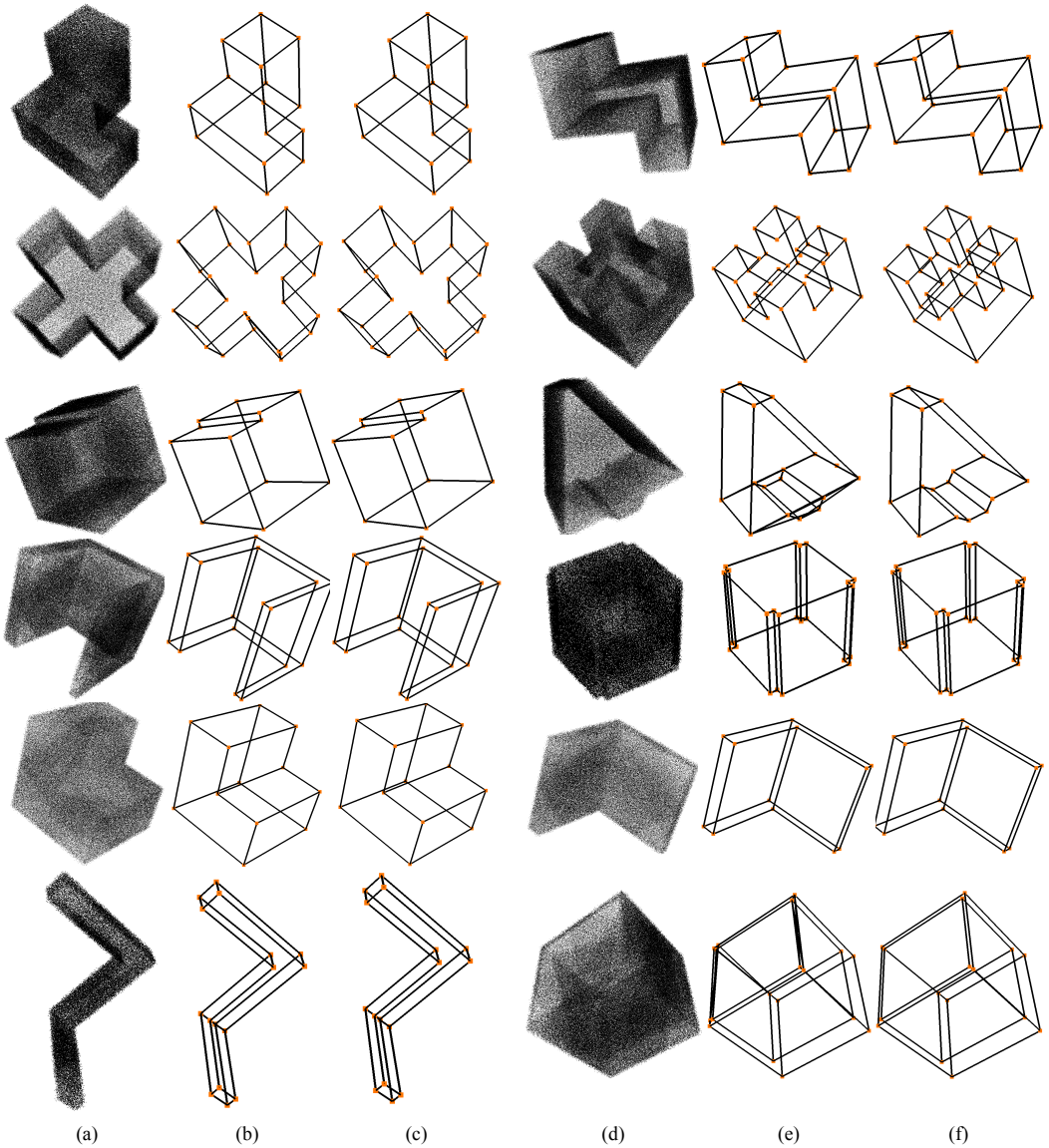

Figure 7: Wireframe reconstruction results on ABC dataset. (a)(d) input raw point clouds; (b)(e) predicted wireframes; (c)(f) ground truth

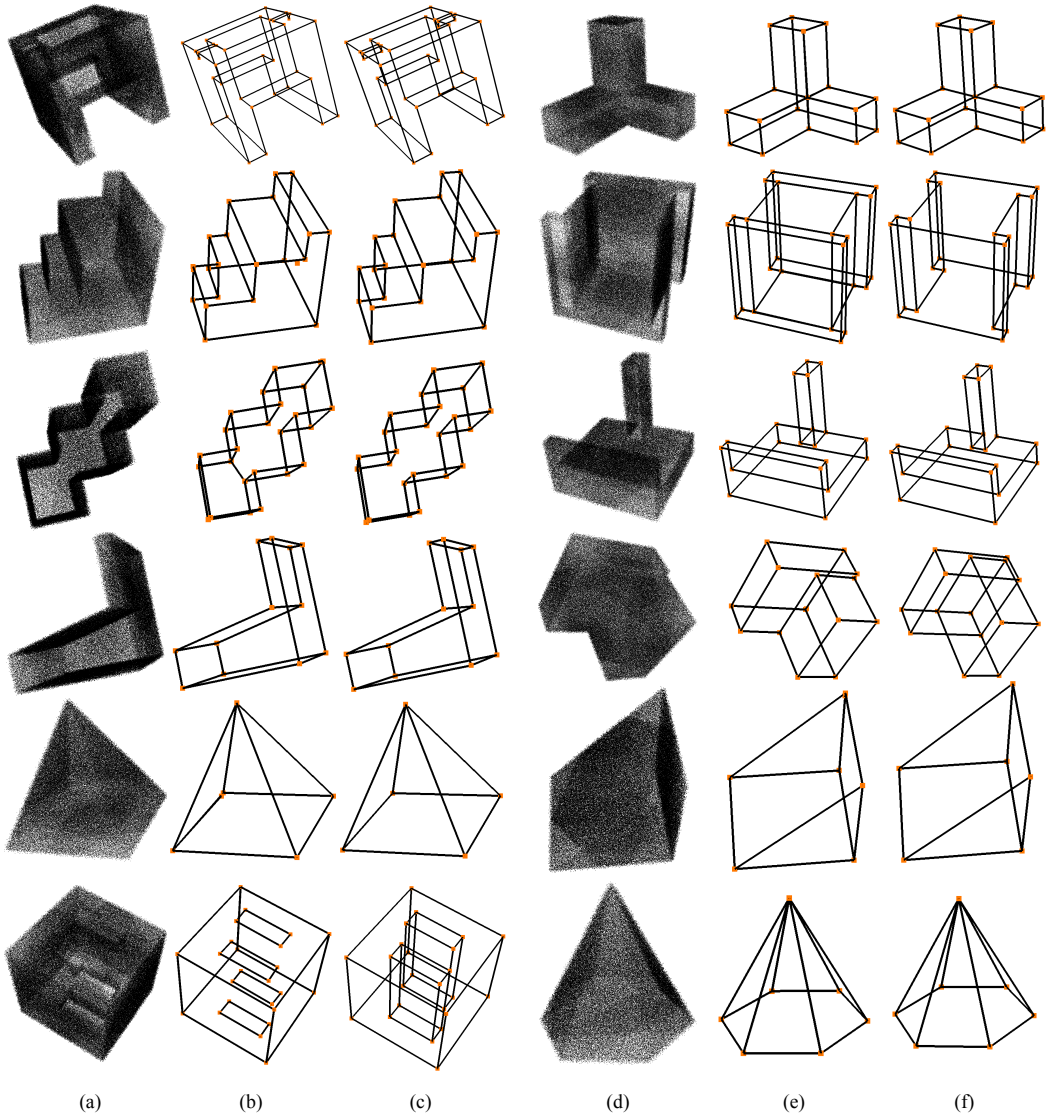

Figure 8: Wireframe reconstruction results on ABC dataset. (a)(d) input raw point clouds; (b)(e) predicted wireframes; (c)(f) ground truth

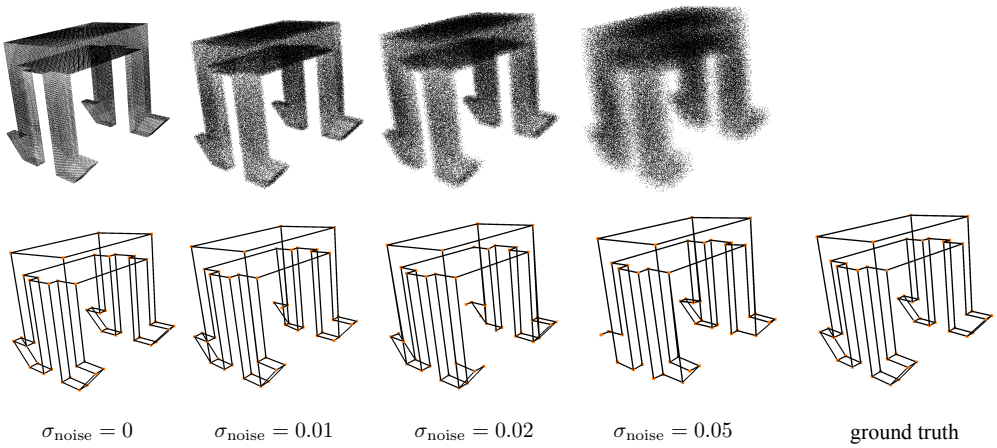

Figure 9: Qualitative results of input point clouds with different noise levels.

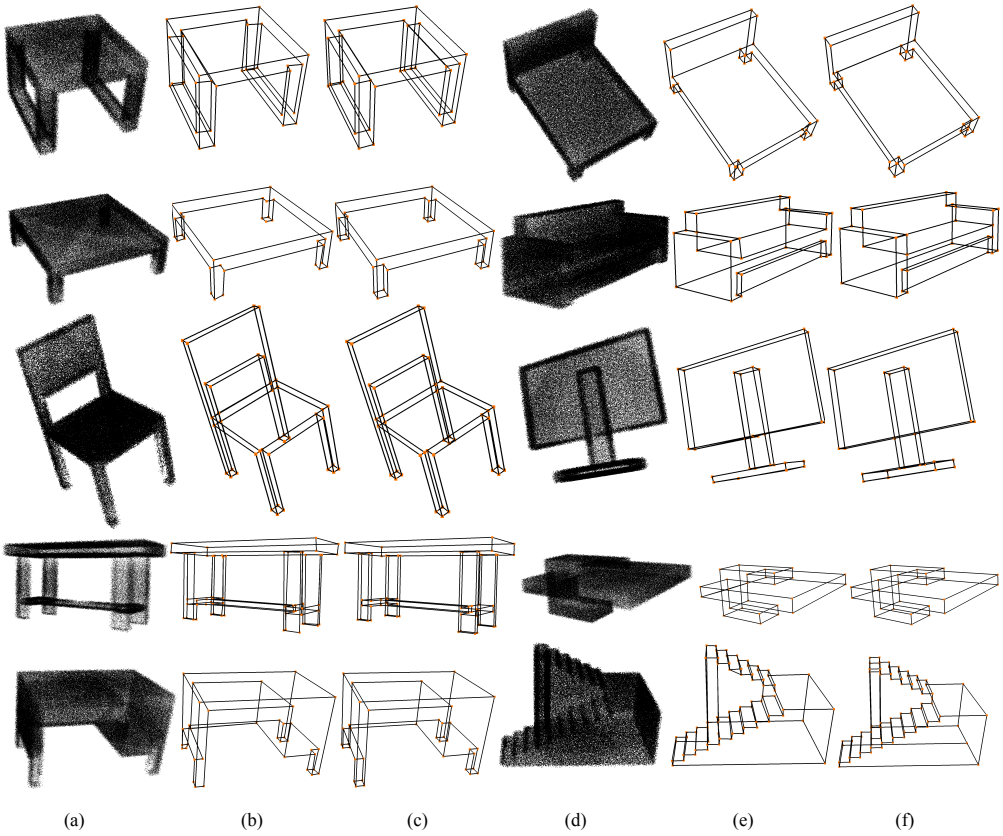

Figure 10: Wireframe reconstruction results on furniture dataset. (a)(d) input raw point clouds; (b)(e) predicted wireframes; (c)(f) ground truth

REFERENCES

Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019a.

Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *IEEE International Conference on Computer Vision (ICCV)*, 2019b.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

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

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 (MICCAI)*, 2015.