Supplementary Materials
PBWR: Parametric Building Wireframe Reconstruction from Aerial LiDAR Point Clouds

Shangfeng Huang[1], Ruisheng Wang[1,*], Bo Guo[2], Hongxin Yang[1]
[1]University of Calgary, [2]Guangdong University of Technology
{shangfeng.huang, ruiswang, hongxin.yang}@ucalgary.ca, guobo.lidar@gmail.com

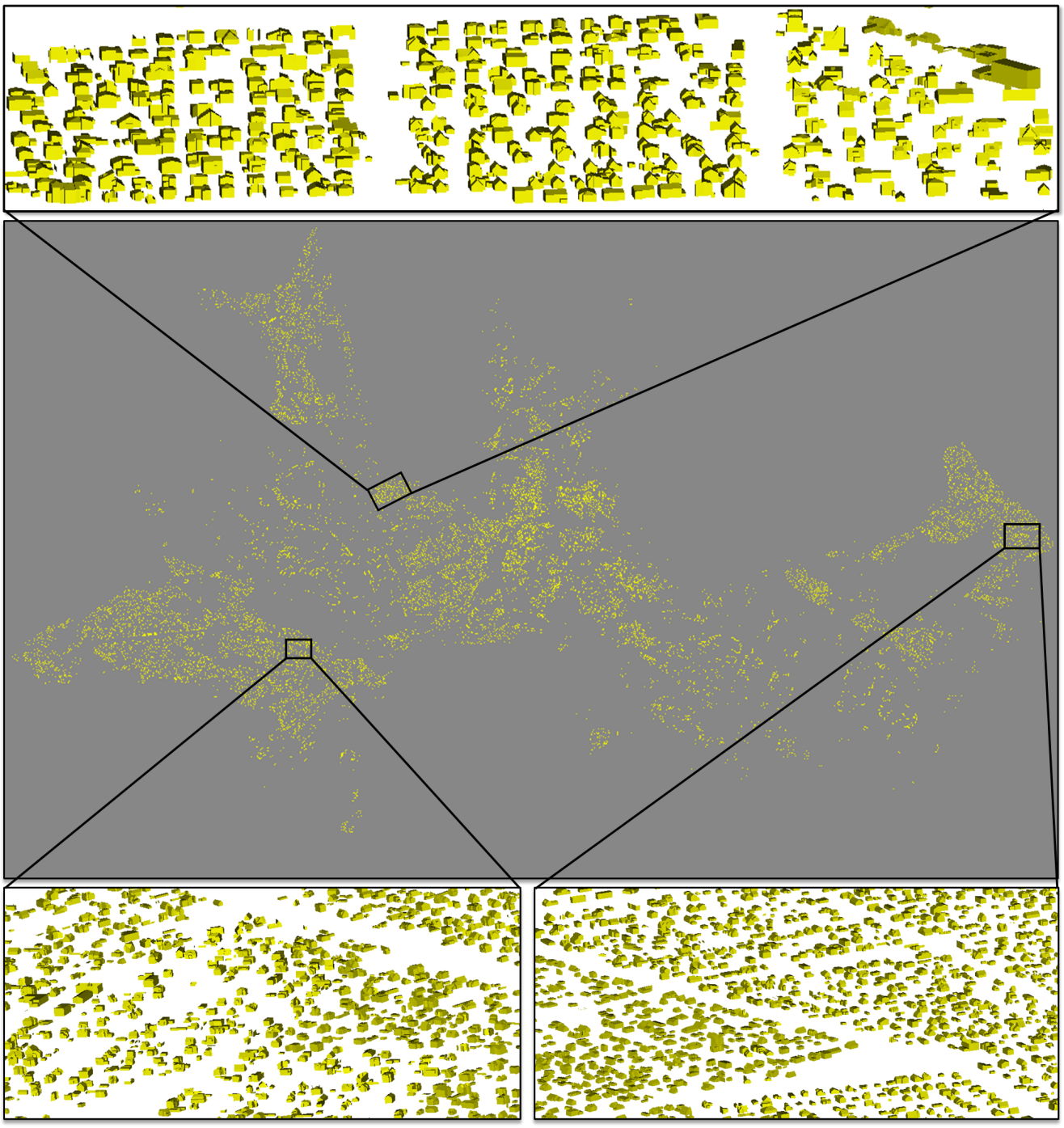


Figure 1. Building reconstruction results of Tallinn city

1. Visualization of building reconstruction in Tallinn City

The building wireframe reconstruction result of Tallinn city comprising 36,084 buildings is shown in Fig. 1. We also provide close-up views of three specific regions to show effectiveness of the reconstruction .

2. Edge Non-Maximum Suppression (E-NMS)

Algorithm 1 shows the pseudo-code for the proposed E-NMS algorithm, incorporating the edge similarity function in bipartite edge matching and edge precision evaluation.

3. Quantitative Comparison Between PBWR and Traditional Methods

Tab. 1 shows quantitative results between traditional methods and PBWR. The corresponding visual results can be found in Fig. 4 in our paper. Specifically, Tab. 1 demonstrates that our PBWR outperforms compared traditional methods. The PBWR can represent building detail with fewer faces, and represent architectural intricacies with less storage space.

Algorithm 1: Edge Non-Maximum Suppression (E-NMS) Algorithm

Input: List of edges E, corresponding confidences C, Edge similarity threshold τ
Output: List of filtered edges $\hat{E}$

Sort the edges E based on confidences C in descending order;
Define Hausdorff Distance function $H_d(e_i, e_j)$;
Define Direction Similarity function $Dir_{sim}(e_i, e_j)$;
Define Length Similarity function $Len_{sim}(e_i, e_j)$;
Edge Similarity function $Edge_{sim}(e_i, e_j) = 2 * H_d(e_i, e_j) + 0.5 * Dir_{sim}(e_i, e_j) + 0.5 * Len_{sim}(e_i, e_j)$;
Initialize an empty list $\hat{E}$ to store selected edges;
while *E is not empty* **do**
    Select the edge e with highest confidence from E;
    Add e to $\hat{E}$;
    Remove e from E;
    for *each edge e' in E* **do**
        if $Edge_{sim}(e, e') < \tau$ **then**
            Remove e' from E;
        end
    end
end

4. Method Details

4.1. Input Embedding Module

In the Input Embedding module, we use a robust aggregation method [4] that learns local structural features. The MLP (Multi-Layer Perceptron) is employed to aggregate local features and output embedding features in 256 channels.

4.2. Transfromer Encoder and Decoder

The standard Transformer with self-attention mechanism [5] is employed. Specifically, the Transformer Encoder consists of six feature encoder layers, each comprising a self-attention and a feed-forward layer. The embedding feature F_{embed}, which already incorporates positional information from the XYZ coordinates, is input into the Transformer Encoder to extract individual point features $F_{en} \in \mathbb{R}^{N \times C_{en}}$. The Transformer Decoder includes six feature decoder layers, with each layers containing a self-attention, multihead-attention and feed-forward module.

4.3. Edge Similarity and Loss Function

Based on experimental results, the balance coefficient α, β, γ of edge similarity are set to 2, 0.5, 0.5, respectively. The balance coefficient λ_{mid}, λ_{comp}, λ_{con}, λ_{quad} and λ_{sim} are set to 5, 1, 1, 2 and 2, respectively.

5. Experimental Details

5.1. Building3D Dataset Processing

The original coordinates of point clouds in the dataset are normalized to a range between -1 m and 1 m before training. Moreover, the data undergoes random augmentation employing strategies delineated in the paper. The ACO (Average Corner Offset) metric is calculated from the results after reverting the normalization process. Specifically, for the Entry-level dataset, the ACO derived from normalized data stands at 0.03 m. However, after performing the reverse normalization process, the calculated ACO rises to 0.22 m. We believe that the reverse normalization process magnifies the ACO error, which can be mitigated or eliminated in future work.

5.2. Implementation Details

During the experiments, the threshold of edge similarity τ was set at 0.15. The threshold of confidence score during training was set to 0.5, but it was set to 0.7 during evaluation to determine whether an edge is a positive edge or not. The model was trained with a batch size of 20, using a single NVIDIA A6000 graphics card. The training time for the entry-level dataset is approximately one day, but it extends to six days for the Tallinn city dataset.

	Building									
RMSE (m) ↓	2.5D Dual [3]	**0.051**	0.086	0.093	0.072	0.113	0.111	0.144	0.111	0.140
	Topology Aware [1]	0.164	0.777	0.718	0.235	0.568	0.483	1.177	5.248	2.019
	City3D [2]	2.362	0.008	0.092	**0.027**	**0.050**	**0.041**	0.529	0.627	**0.061**
	PBWR	0.054	**0.002**	**0.084**	0.042	0.059	0.053	**0.074**	**0.054**	0.086
3D IOU ↑	2.5D Dual [3]	0.786	0.636	0.738	0.4772	0.524	0.421	0.313	0.192	0.283
	Topology Aware [1]	0.441	0.460	0.420	0.435	0.471	0.505	0.658	0.268	0.204
	City3D [2]	0.754	0.942	0.952	0.940	**0.966**	0.888	0.721	0.973	0.905
	PBWR	**0.969**	**0.961**	**0.960**	**0.949**	0.962	**0.982**	**0.966**	**0.977**	**0.965**
Number of Faces ↓	2.5D Dual [3]	136	384	239	68	124	152	628	468	163
	Topology Aware [1]	5483	2056	985	562	132	2204	1484	1334	2869
	City3D [2]	58	216	289	72	336	80	161	290	180
	PBWR	**30**	**24**	**19**	**9**	**24**	**9**	**9**	**17**	**20**

Table 1. **Quantitative comparison.** The quantitative results between traditional methods and PBWR in terms of RMSE (Root Mean Square Error), 3D IOU (Intersection over Union), and number of faces.

6. Model Generalization

The Building3D dataset provides additional aerial LiDAR point clouds without corresponding wireframe models in specific Estonian cities. We leverage this data to evaluate PBWR's generalization capability. Specifically, we use aerial point clouds from four small Estonian cities—**Hiiumaa**, **Keila**, **Loksa**, and **Sillamae**—to assess PBWR's performance. Among these, Hiiumaa comprises 2,863 buildings, Keila encompasses 2,909 buildings, Loksa includes 1,010 buildings, and Sillamae consists of 2,011 buildings. Fig. 2 shows visualization results of Hiiumaa data. Fig. 3 and Fig. 4 depict visualization results of Keila data. Fig. 5 presents visualization results of Loksa data, while Fig. 6 shows visualization results of Sillamae data. Based on the visualized results, our model exhibits generalization across diverse cities.

7. More Visualization Results

Fig. 8 and Fig. 7 present additional visualized results of Tallinn City data, offering a more intuitive observation of the model's performance. In addition, we provide comparative visualization between prediction and ground truth in the fourth and final columns.

References

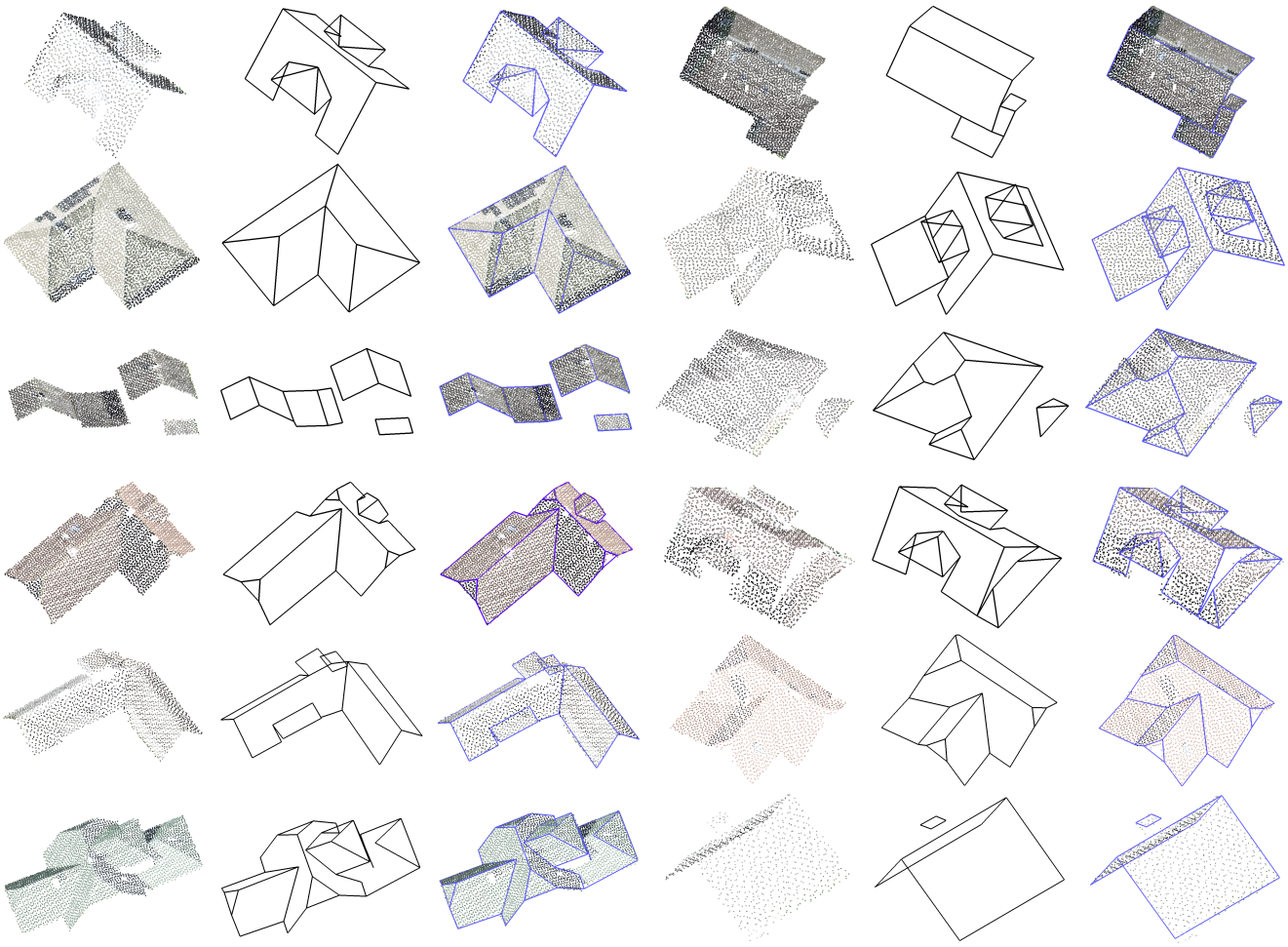
[1] Dong Chen, Ruisheng Wang, and Jiju Peethambaran. Topologically aware building rooftop reconstruction from airborne laser scanning point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 55(12):7032–7052, 2017. 3

[2] Jin Huang, Jantien Stoter, Ravi Peters, and Liangliang Nan. City3d: Large-scale building reconstruction from airborne lidar point clouds. *Remote Sensing*, 14(9):2254, 2022. 3

[3] Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. Dual contouring of hermite data. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 339–346, 2002. 3

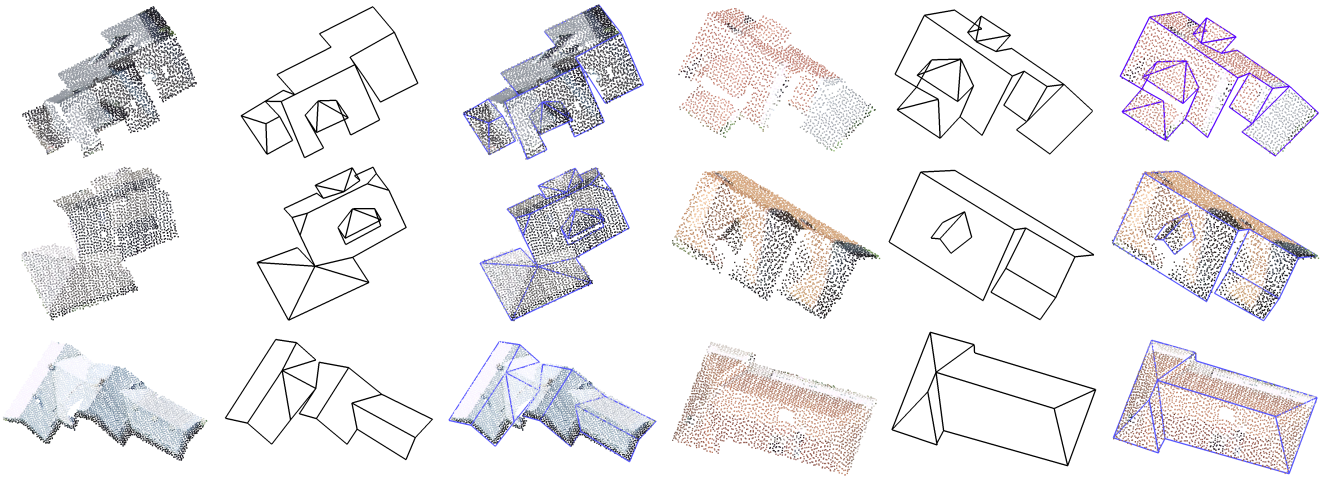
[4] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2



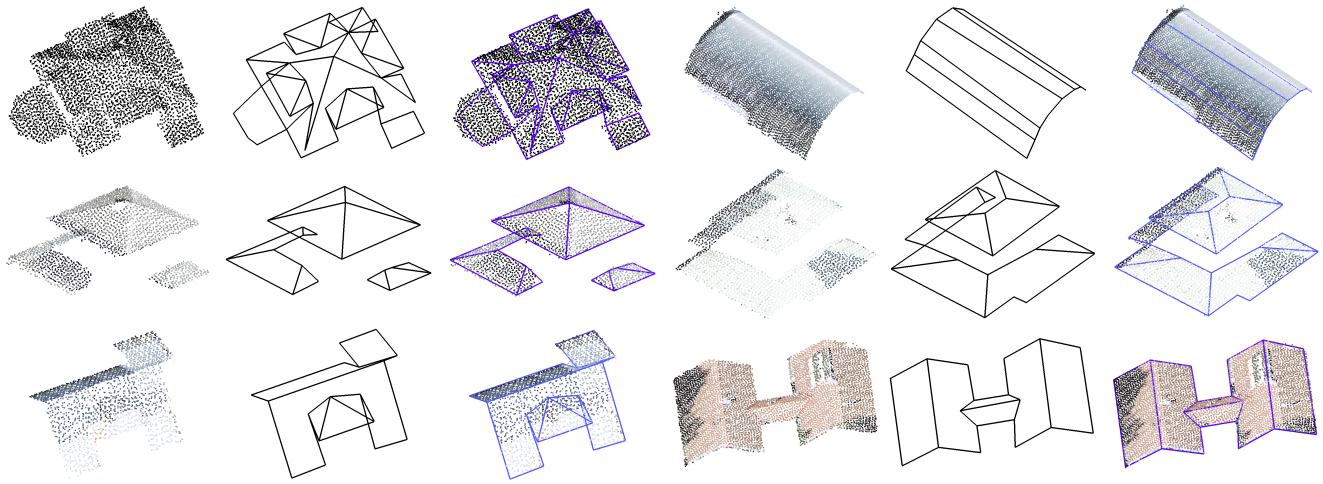
(a) Point Clouds (b) PBWR (c) PBWR (d) Point Clouds (e) PBWR (f) PBWR

Figure 2. **Model generalization**.Visualization results of Hiiumaa city data.



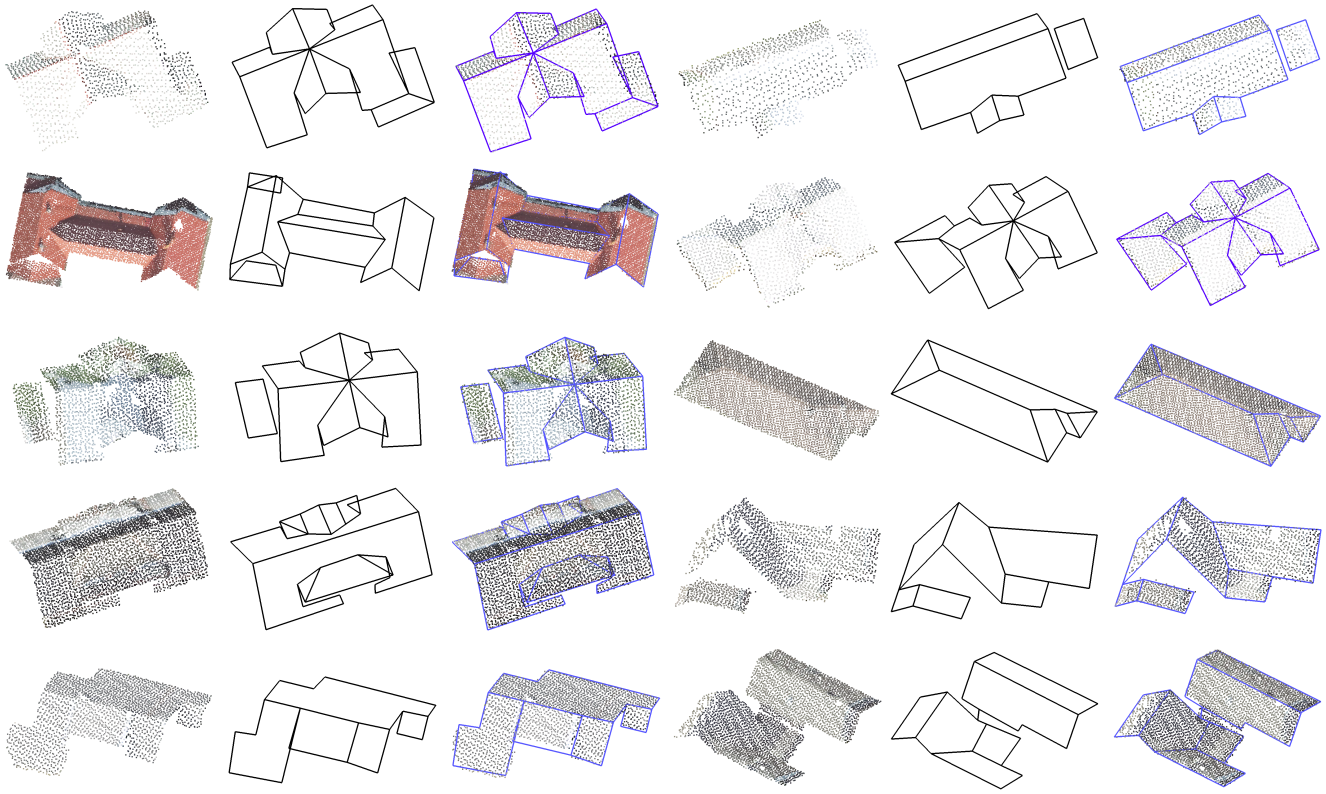
(a) Point Clouds (b) PBWR (c) PBWR (d) Point Clouds (e) PBWR (f) PBWR

Figure 3. **Model generalization**.Visualization results for Keila city data.



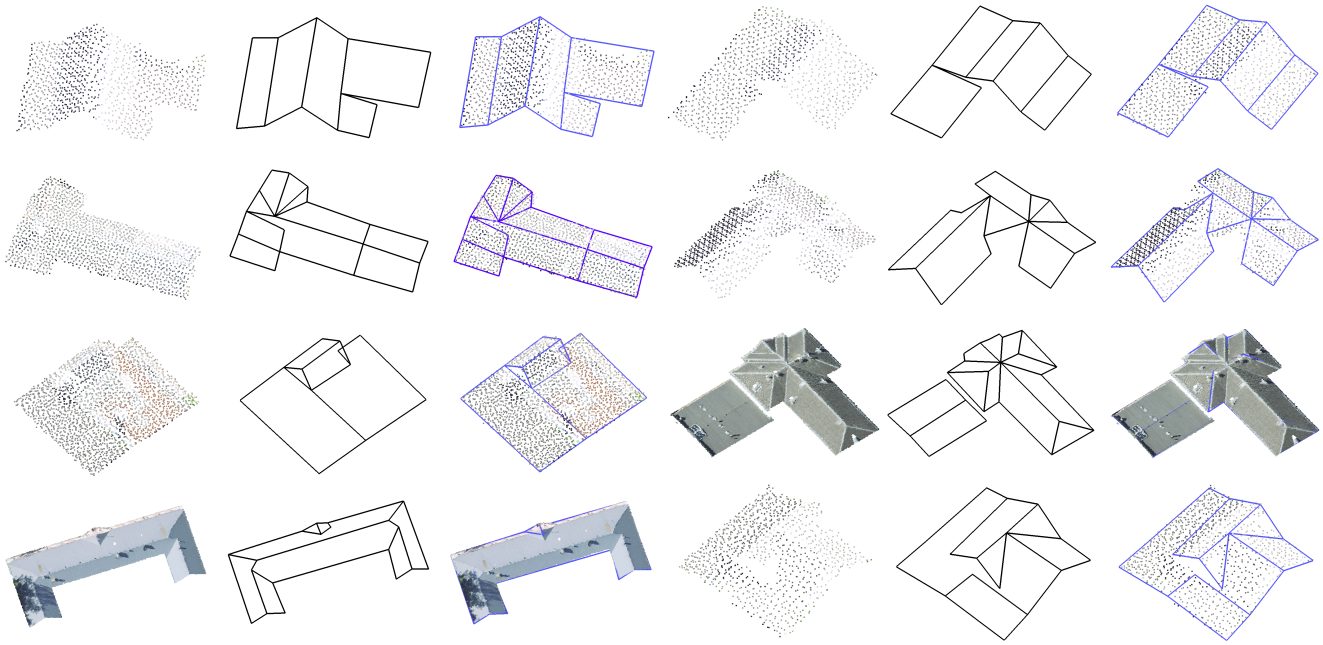
(a) Point Clouds (b) PBWR (c) PBWR (d) Point Clouds (e) PBWR (f) PBWR

Figure 4. **Model generalization**.Visualization results of Keila city data.



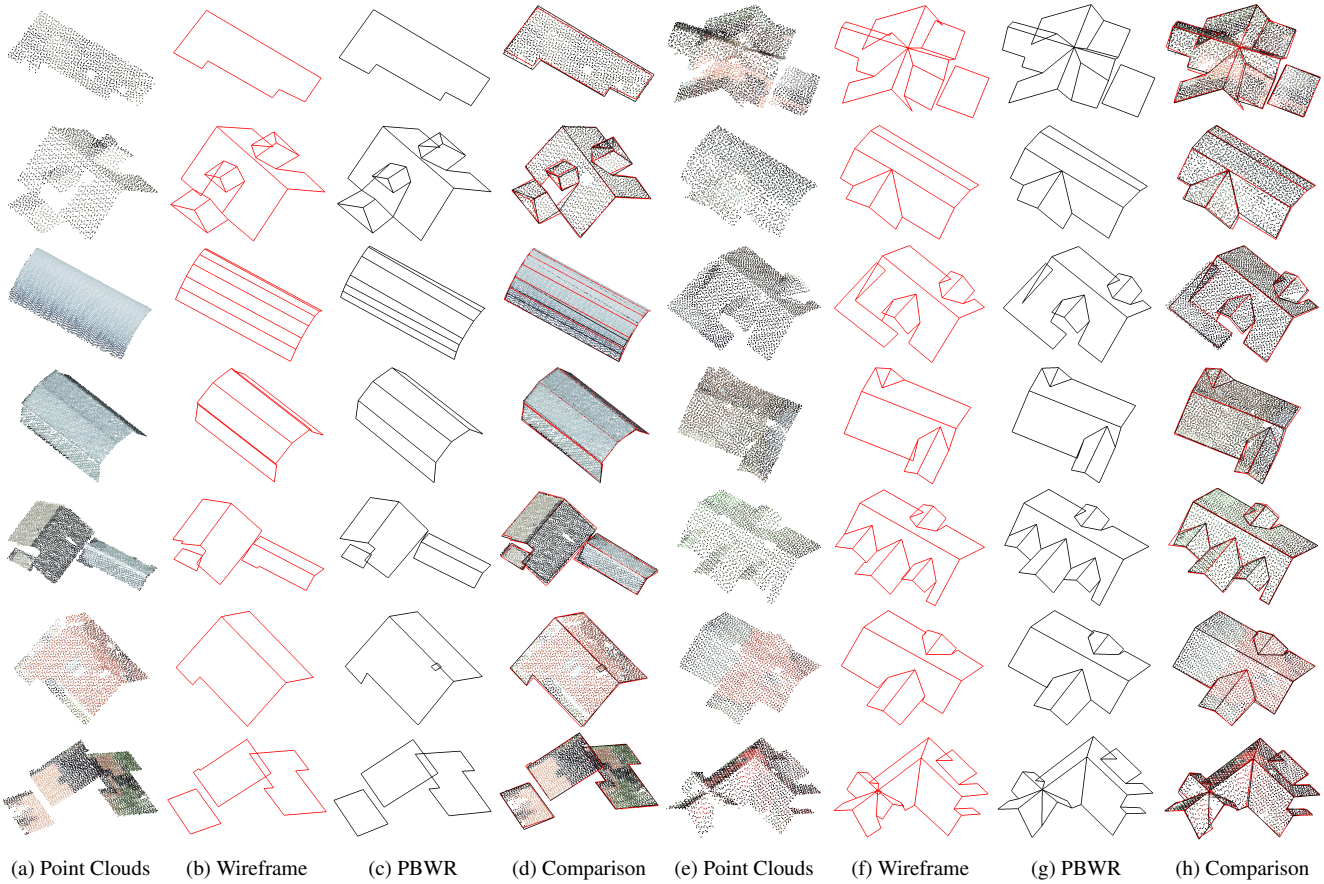
(a) Point Clouds (b) PBWR (c) PBWR (d) Point Clouds (e) PBWR (f) PBWR

Figure 5. **Model generalization**.Visualization results of Loksa city data.



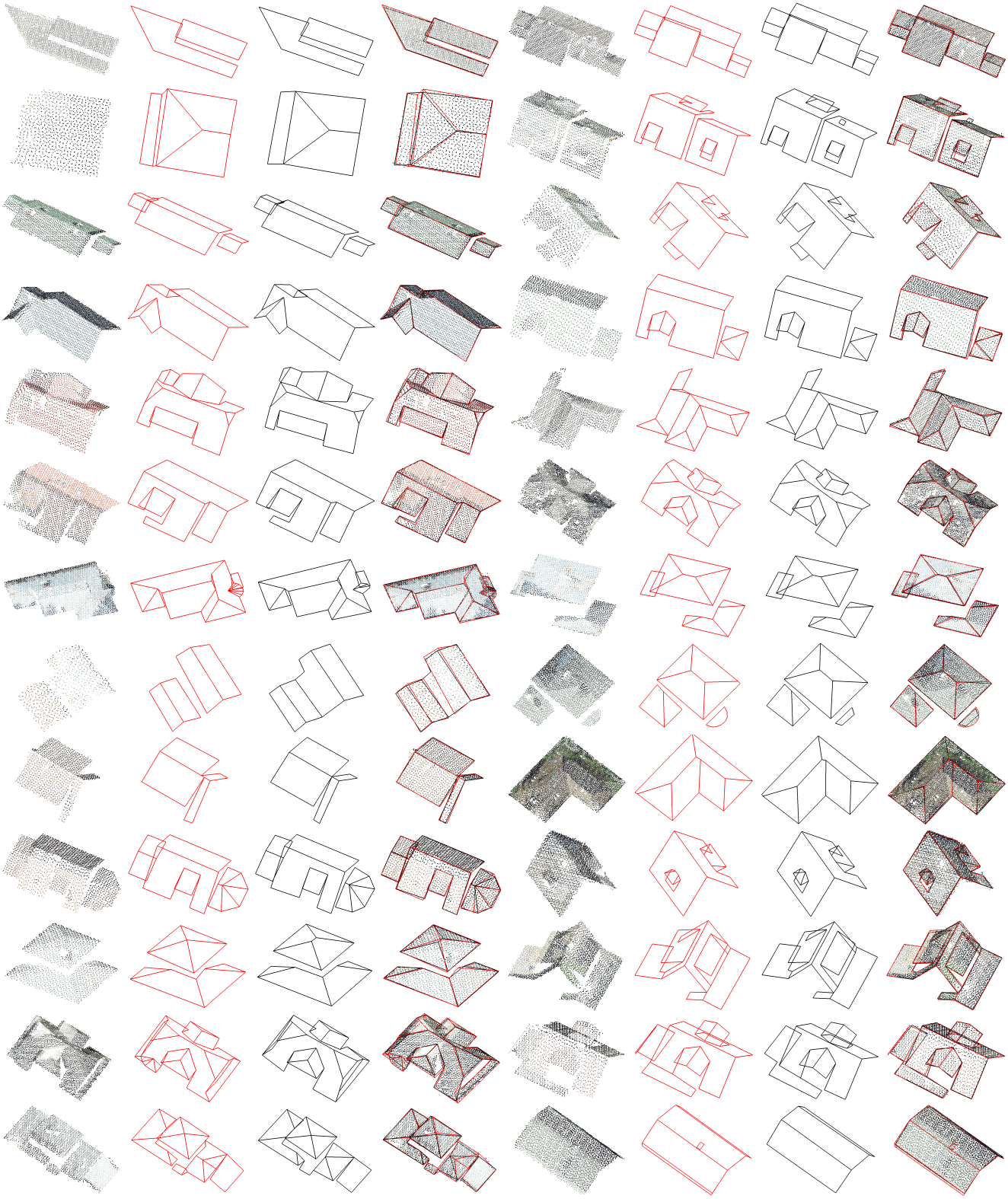
(a) Point Clouds (b) PBWR (c) PBWR (d) Point Clouds (e) PBWR (f) PBWR

Figure 6. **Model generalization**.Visualization results of Sillamae city data.



(a) Point Clouds (b) Wireframe (c) PBWR (d) Comparison (e) Point Clouds (f) Wireframe (g) PBWR (h) Comparison

Figure 7. PBWR-Tallinn Visualization Results in Tallinn City



(a) Point Clouds (b) Wireframe (c) PBWR (d) Comparison (e) Point Clouds (f) Wireframe (g) PBWR (h) Comparison

Figure 8. PBWR-Tallinn Visualization Results in Tallinn City