[Supplementary Document] Vectorizing World Buildings: Planar Graph Reconstruction by Primitive Detection and Relationship Inference

Nelson Nauata and Yasutaka Furukawa

Simon Fraser University, Burnaby, Canada
{nnauata,furukawa}@sfu.ca

The supplementary document provides 1) more statistics on our benchmark; 2) implementation details of primitive detection networks; 3) the complete mathematical specification of our Integer Programming (IP) formulation; and 4) additional experimental results.

1 Benchmark statistics

Figure 1 shows histograms of training and testing samples against different numbers of corners, edges, and regions.

2 Implementation details of primitive detectors

Standard neural architectures are used for the primitive detection: Fully Convolutional Network (FCN) for corners [2], Dilated Residual Networks (DRN) [4] for edges, and Mask-RCNN [1] for regions.

Corner detection: In our corner detection pipeline, we borrow an existing architecture [2]. Our Fully Convolutional Network (FCN) divides the image in a $H_b \times W_b$ grid, where each cell is responsible for predicting a confidence score c_{conf} and (x, y) coordinates of a corner residing within a bin. The corner network proposal head is trained using binary cross-entropy loss at each output cell in the grid. Similarly to [2], we utilize a Google's Inception-v2 model [3] for encoding the input image. We train the network with a learning rate of 0.001 (decay ratio $\gamma = 0.1$ every each 5 epochs) using ADAM optimizer for 16 epochs and utilize only corners with $c_{conf} \geq 0.2$, batch size is set to 1. Our output grid size is 120×120 of 256-dimensional features which are regressed to the output of the network.

Edge detection: We utilize the DRN-D-105 architecture [4]. Given an input RGB image I (256×256), we obtain for each building an edge segmentation mask by optimizing a binary cross-entropy loss for each cell in the final feature map at the end of the network. We fine-tune the pre-trained DRN-D-105 architecture with learning rate equal to 0.0001 for 40 epochs and batch size set to 8.

Region detection: We utilize the official code release for Mask-RCNN [1] for performing instance region segmentation. We train the network utilizing regions

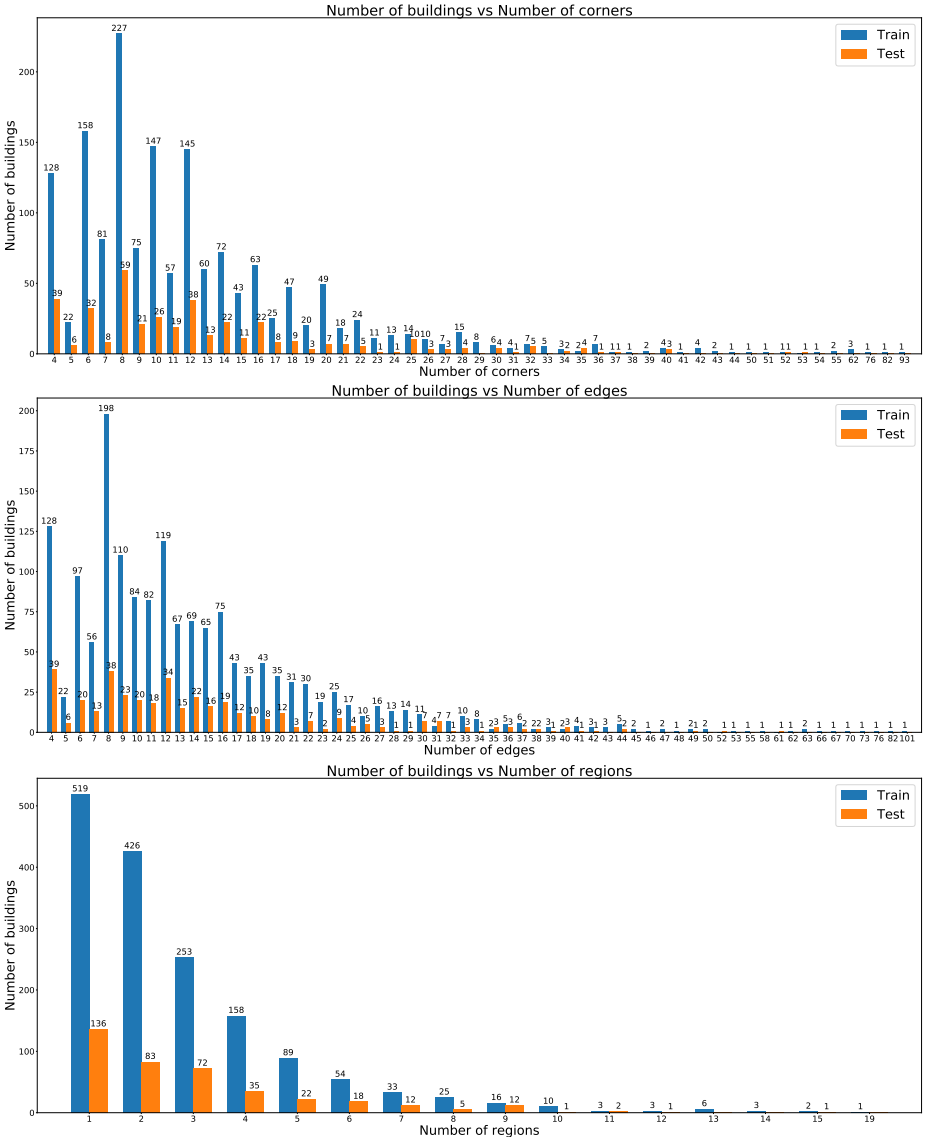

Fig. 1. Building counts against the numbers of corners, edges, and regions.

from the annotations as objects belonging to the same class. Our model was initialized with R-50-FPN architecture and trained with learning rate equal to 0.002 with decay of 0.0001. At test phase for a target building, we extract up to N (=100) regions.

3 Integer Programming (IP) formulation

Objective function: Indicator variables are defined for each primitive: I_{cor} for a corner $c \in \mathcal{C}$; I_{edg} for an edge $e \in \mathcal{E}$; and I_{reg} for a region $r \in \mathcal{R}$. We also have an indicator variable I_{dir} for a corner to an incident edge direction relationship.

$$\begin{aligned} \max_{\{I_{cor}, I_{edg}, I_{reg}, I_{dir}\}} & \sum_{e \in \mathcal{E}} \underbrace{(e_{conf} c'_{conf} c''_{conf} - 0.5^3) I_{edg}(e)}_{\text{corner and edge primitives}} \\ & + 0.1 \sum_{c \in \mathcal{C}} \sum_{\theta \in \mathcal{D}_c} \underbrace{(\theta_{conf} c_{conf} - 0.5^2) I_{dir}(\theta, c)}_{\text{corner-to-edge relationship}} \\ & + \sum_{r \in \mathcal{R}} \underbrace{I_{reg}(r)}_{\text{region primitive}} . \end{aligned} \tag{1}$$

c_{conf} and e_{conf} denotes the confidence scores for the corner and the edge detections, respectively. θ_{conf} denotes the corner-to-edge relationship confidence. Note that region and region-to-region relationship confidences were used for thresholding the detections and will not be in the optimization. With abuse of notation, c' and c'' denotes the end-points of an edge e.

Slack Variables: In the following sections we describe constraints as hard constraints however, we utilize slack variables to soften them. For instance, given a constraint in the form of $I_A I_B = C$, we can split it into two additional constraints $I_A I_B \leq C + S_{up}$ and $I_A I_B \geq C - S_{lo}$ and add $-S_{up}$ and $-S_{lo}$ in the objective function, in order to approximate a lower and upper bound to a constant C. We perform similar procedure for constraints in the form of $I_A I_B \geq C$ and $I_A I_B \leq C$.

Topology constraints: We enforce three topology priors as constraints: (1) degree of each corner should be greater or equal to two (Eq. 2), (2) active edge must have its end-points active (Eq. 3) and (3) two intersecting edges e_k and e_l can not be active at the same time (Eq. 4).

$$\sum_{e \in \mathcal{E}_c} I_{edg}(e) \geq 2 I_{cor}(c), \tag{2}$$

$$I_{cor}(c') + I_{cor}(c'') = I_{edg}(e), \tag{3}$$

$$I_{edg}(e_k) I_{edg}(e_l) = 0, \tag{4}$$

where $\mathcal{E}_c$ represents the set of all candidate edges incident to c.

Region primitive constraints: Region primitives are added as constraints by (1) enforcing indicator variables of intersecting edges and regions to not be active at the same time (Eq. 5) and (2) enforcing the activation of edge indicator variables surrounding a region (Eq. 6). For the latter, we trace a boundary of the predicted region and cast rays γ (i.e. line segments with length and width equal to 100 and 2 pixels, respectively) in the outward direction (i.e. normal to

the traced boundary) for every 2 pixels. We collect edges that intersect a γ and enforce that at least one edge should be active.

$$\sum_{e \in \mathcal{E}^r} I_{edg}(e) I_{reg}(r) = 0, \tag{5}$$

$$\sum_{e \in \mathcal{E}^\gamma} I_{edg}(e) \geq I_{reg}(r), \tag{6}$$

where $\mathcal{E}^r$ and $\mathcal{E}^\gamma$ are the set of edges that intersect a region r and a ray γ, respectively.

Region-to-region relationship constraints: Considering a pair of regions sharing a common boundary predicted as a segmentation mask. We fit a line segment to the boundary segment, consider an orthogonal line segment β (16 pixels in length) at the center. We collect all the edge primitives that intersect with the last line segment. One of them must be the boundary edge.

$$\sum_{e \in \mathcal{E}^\beta} I_{edg}(e) = 1, \tag{7}$$

$$\tag{8}$$

where $\mathcal{E}^\beta$ represents the set of all candidate edges intersecting β.

Corner-to-edge relationship constraints: We design constraints to enforce incident edge indicator variables to be active consistently with its corresponding corner and directional bin. In addition, if a corner-to-edge confidence is below 0.2 for a corner and an incident direction, we do not allow any edges in that direction bin to be on. In order to achieve this the following two constraints are sufficient.

$$\sum_{e \in \mathcal{E}^\theta} I_{edg}(e) = I_{dir}(\theta, c), \tag{9}$$

$$\sum_{e \in \mathcal{E}'} I_{edg}(e) = 0, \tag{10}$$

$$\tag{11}$$

where $\mathcal{E}^\theta$ is a set of collected edges in a direction θ within 5 degrees in angular distance and $\mathcal{E}'$ is a set of edges incident in all directions with confidence lower than 0.2.

4 Additional experimental results

Figure 2 presents intermediate results for detected primitives and relationships from our method. Figures 3-35 present additional experimental results against the five competing methods over all the testing samples.

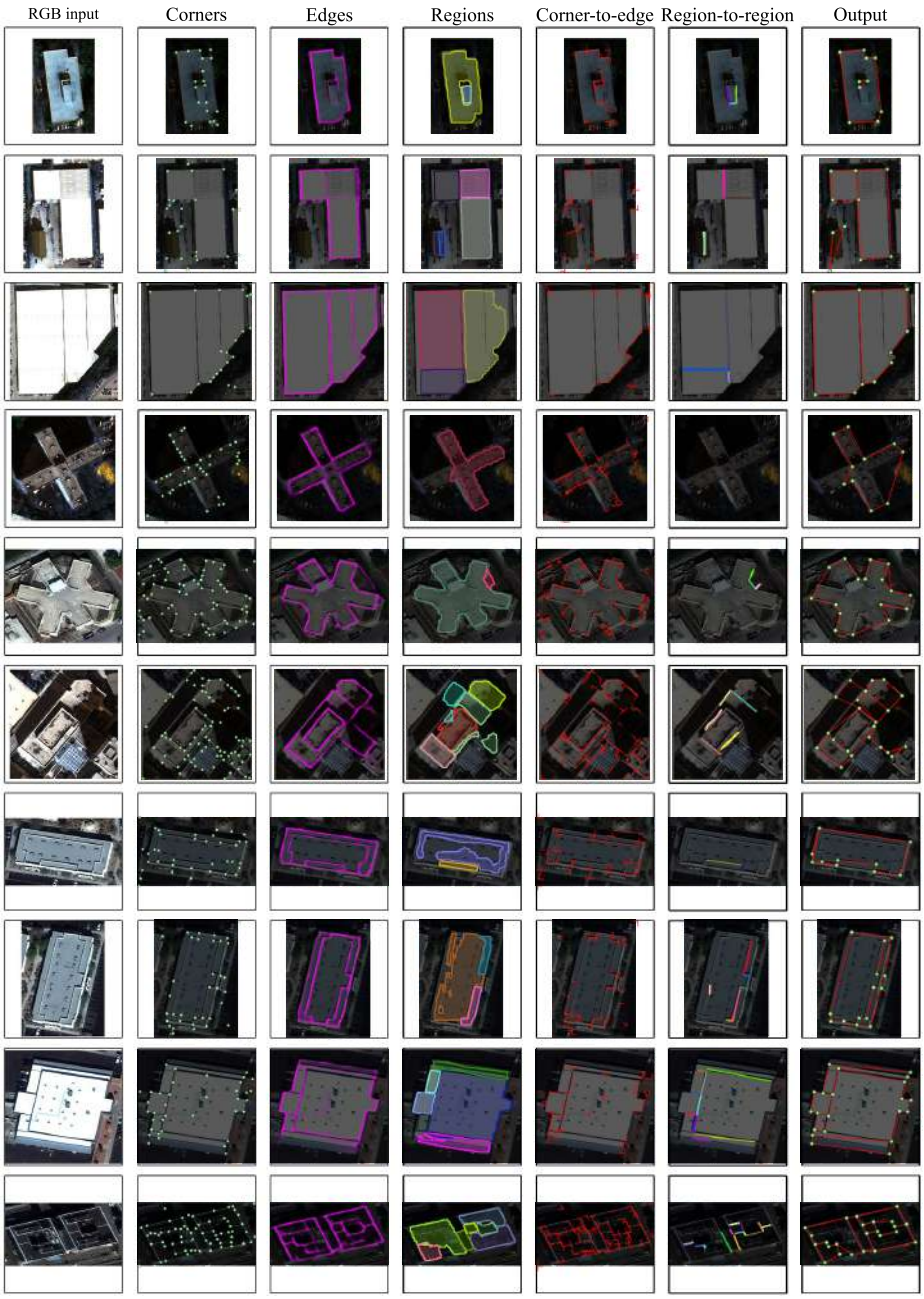

Fig. 2. Intermediate results displaying detected primitives and relationships in our pipeline.

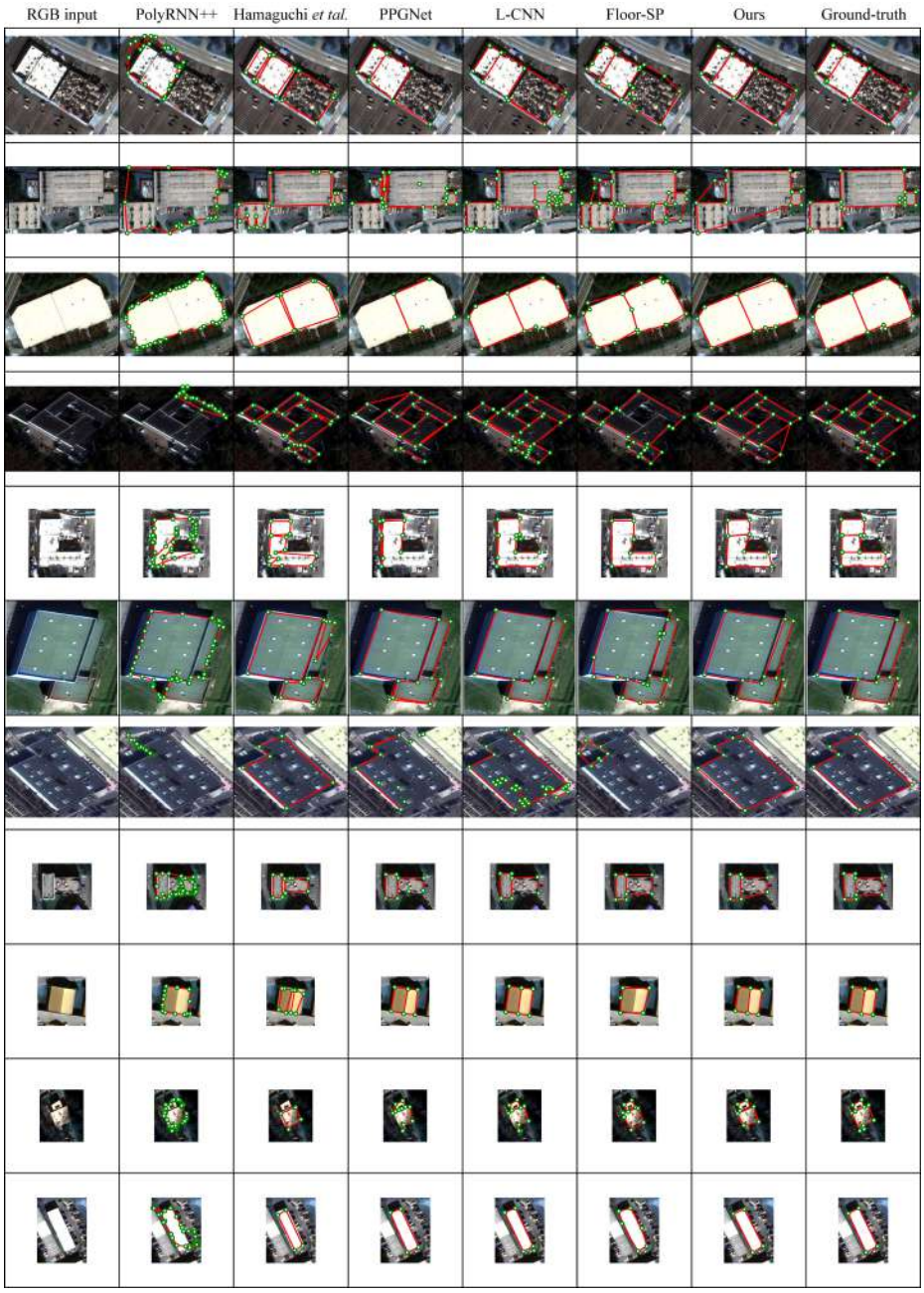

Fig. 3. Additional qualitative results.

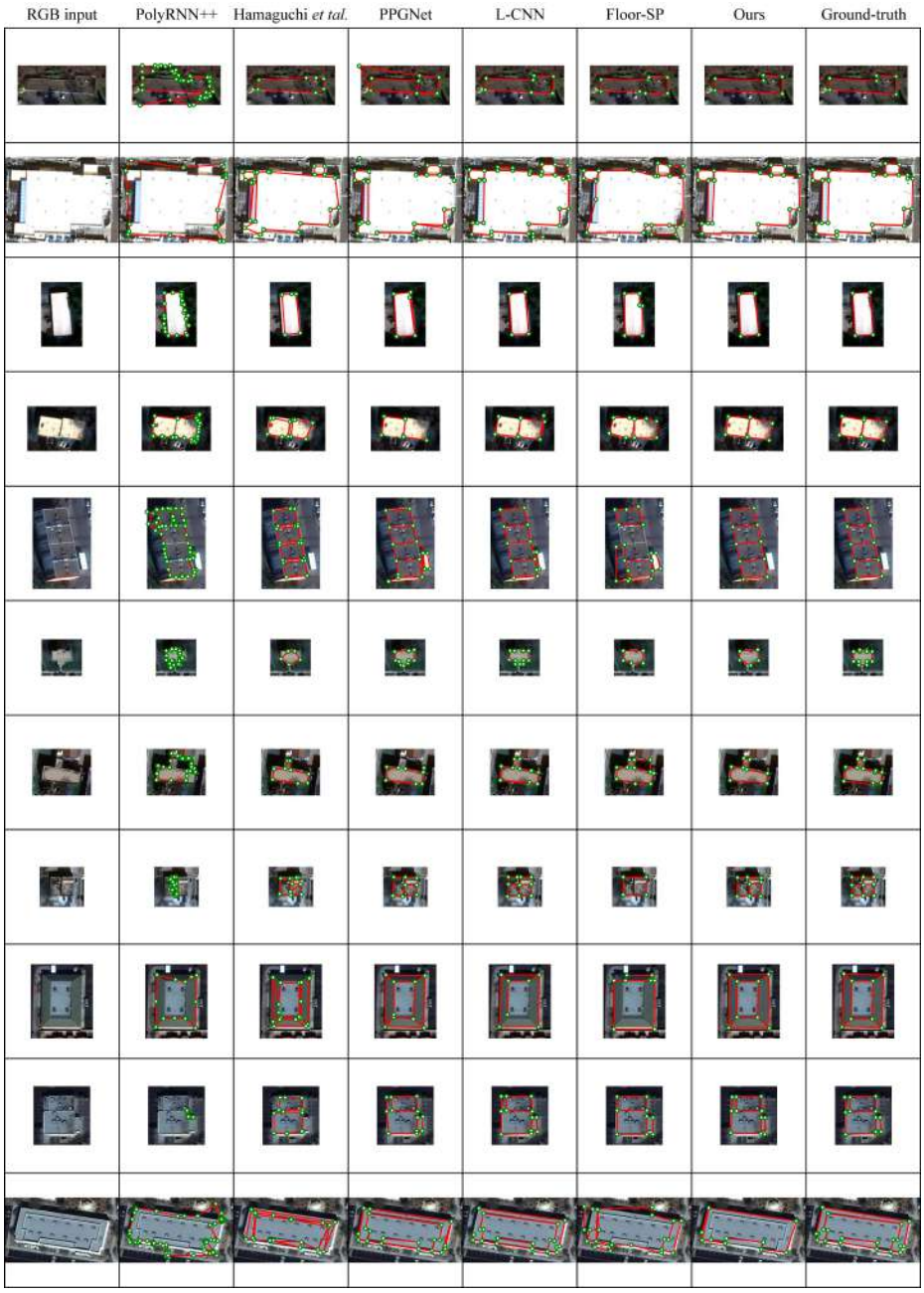

Fig. 4. Additional qualitative results.

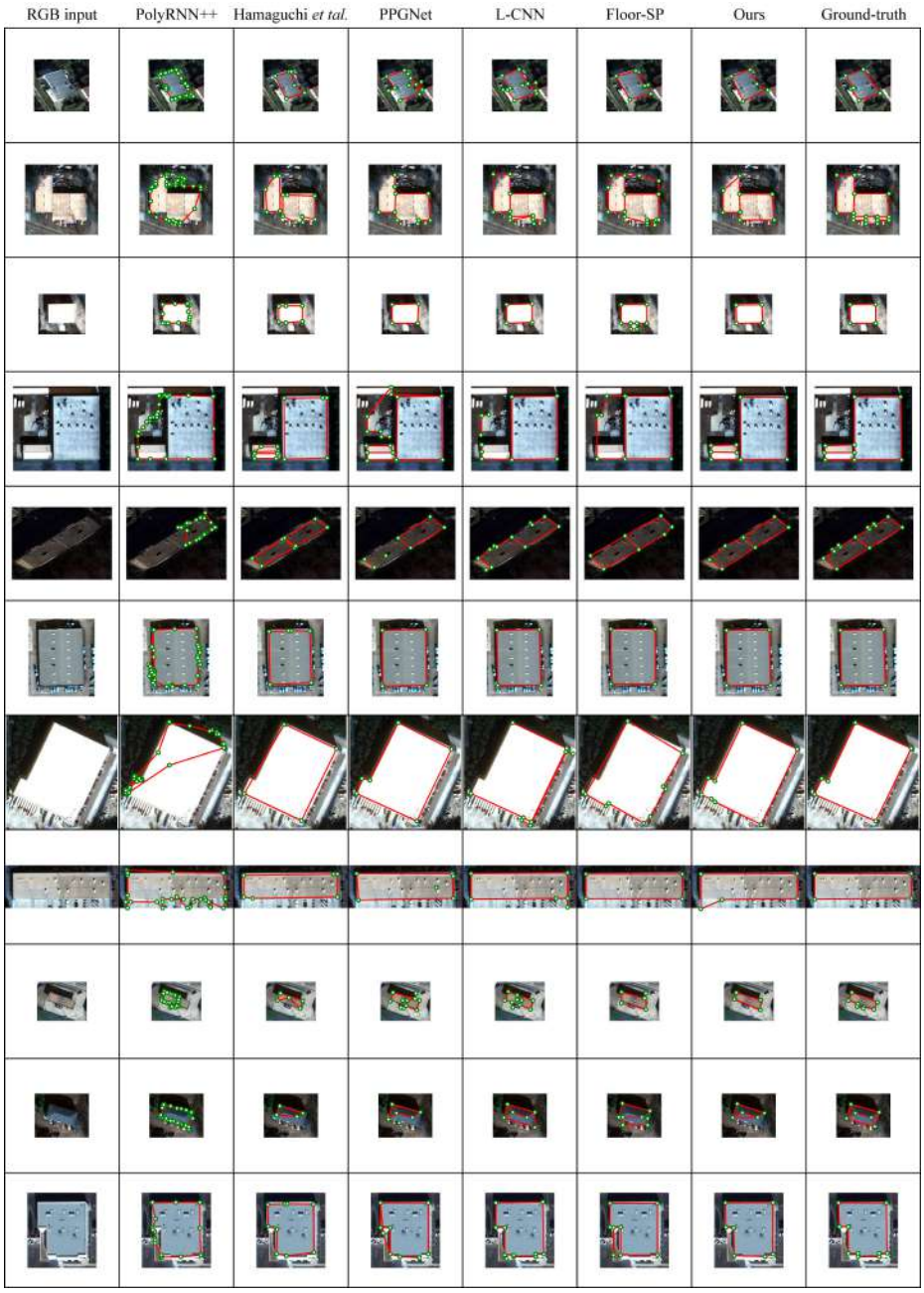

Fig. 5. Additional qualitative results.

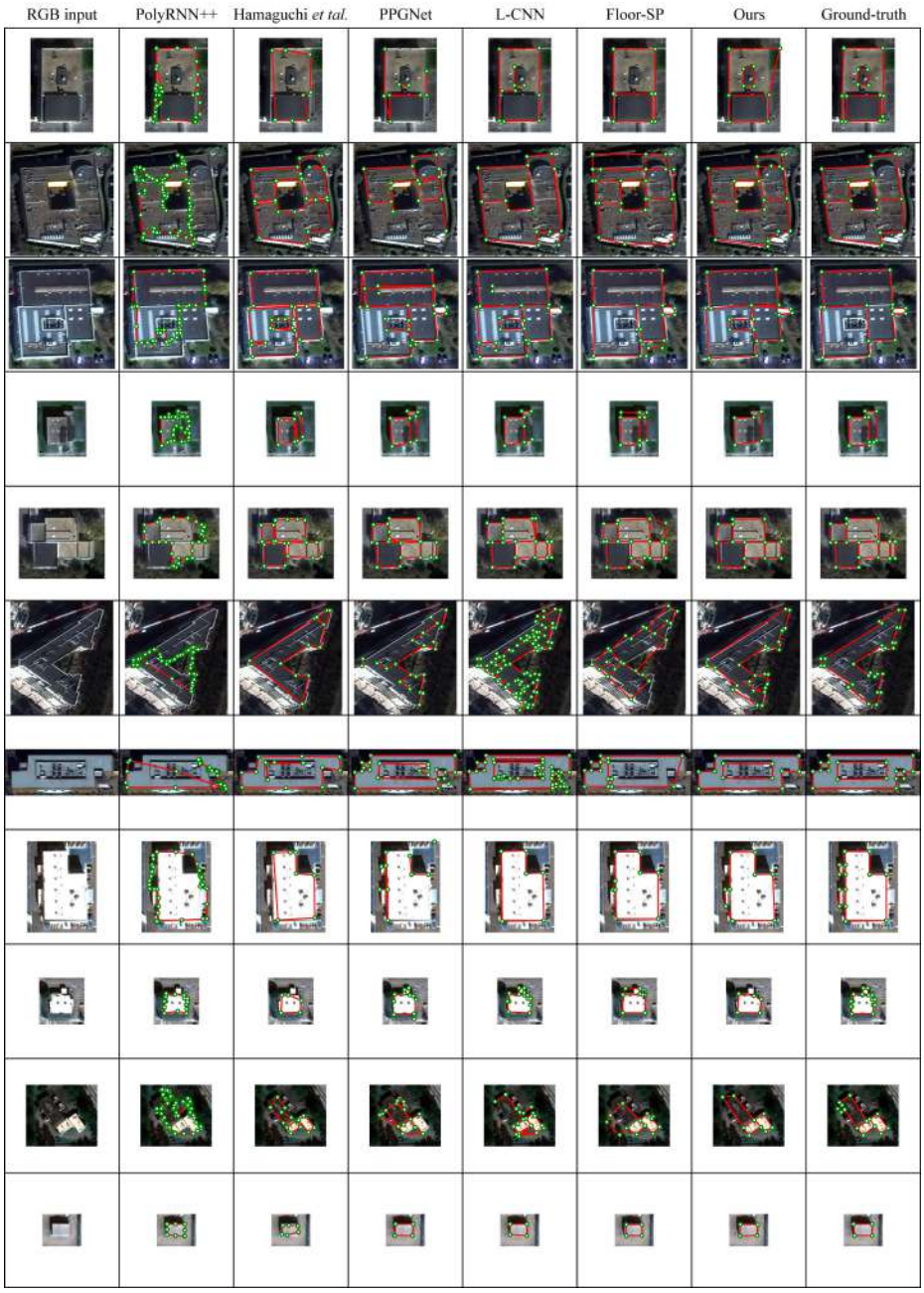

Fig. 6. Additional qualitative results.

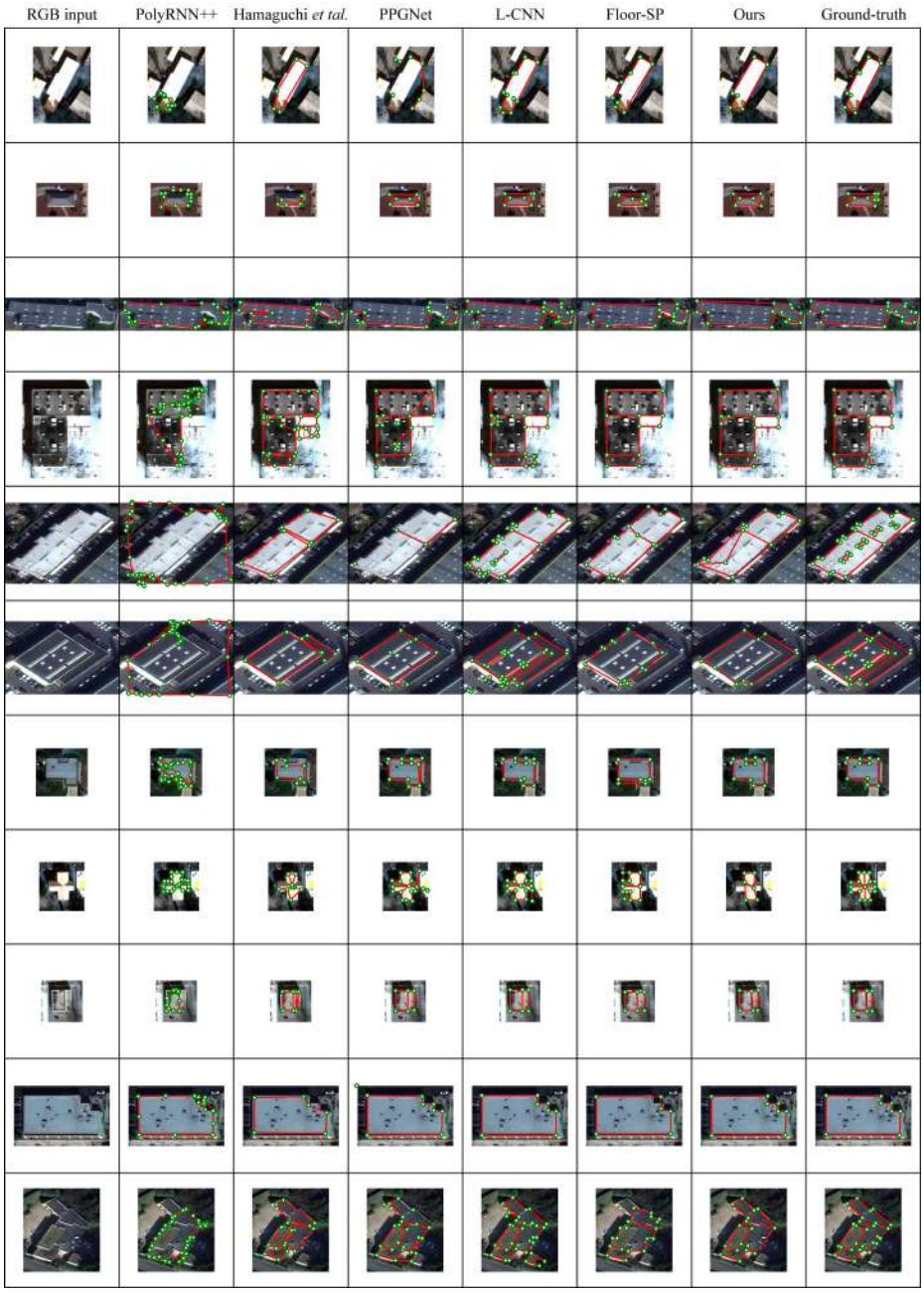

Fig. 7. Additional qualitative results.

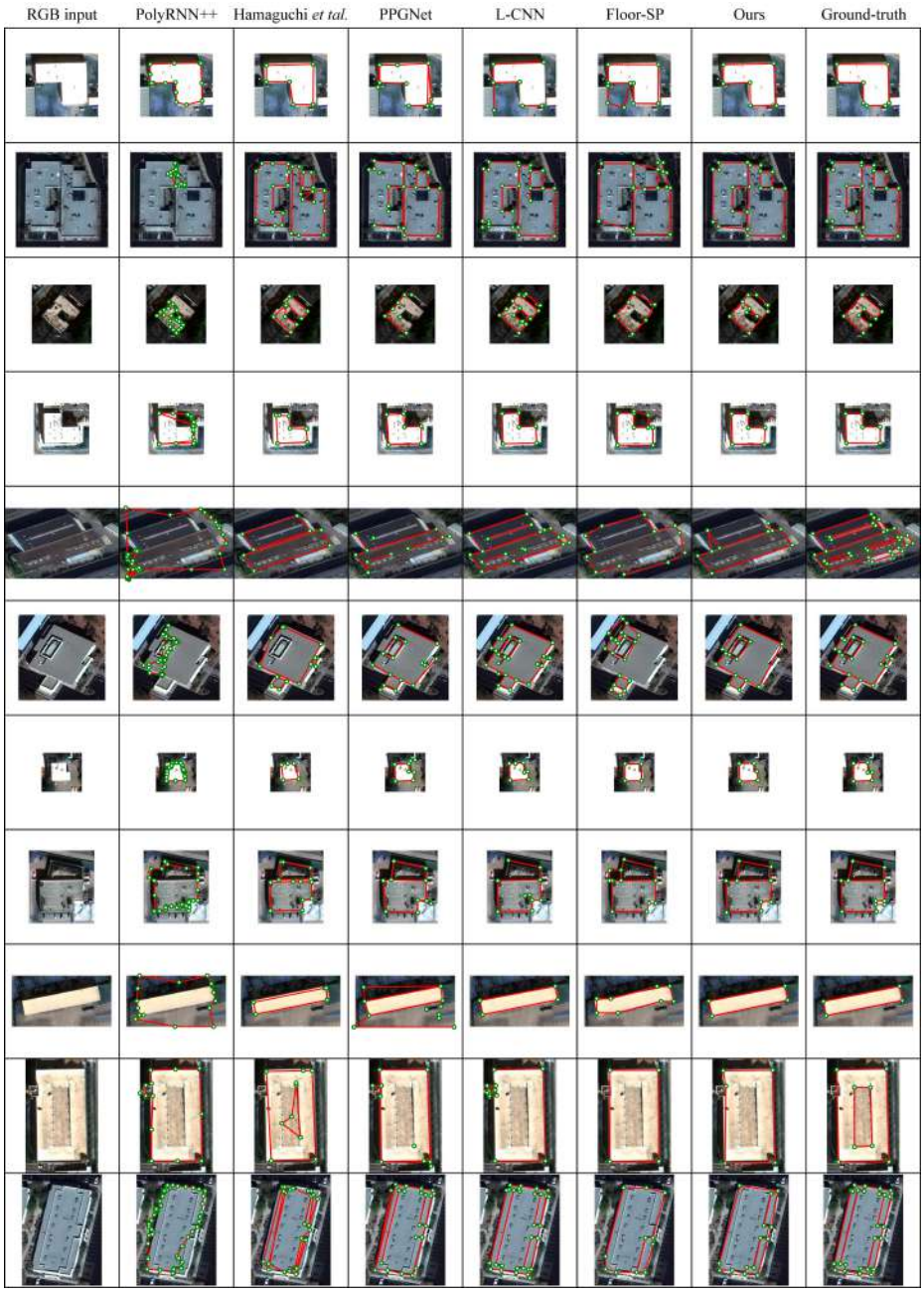

Fig. 8. Additional qualitative results.

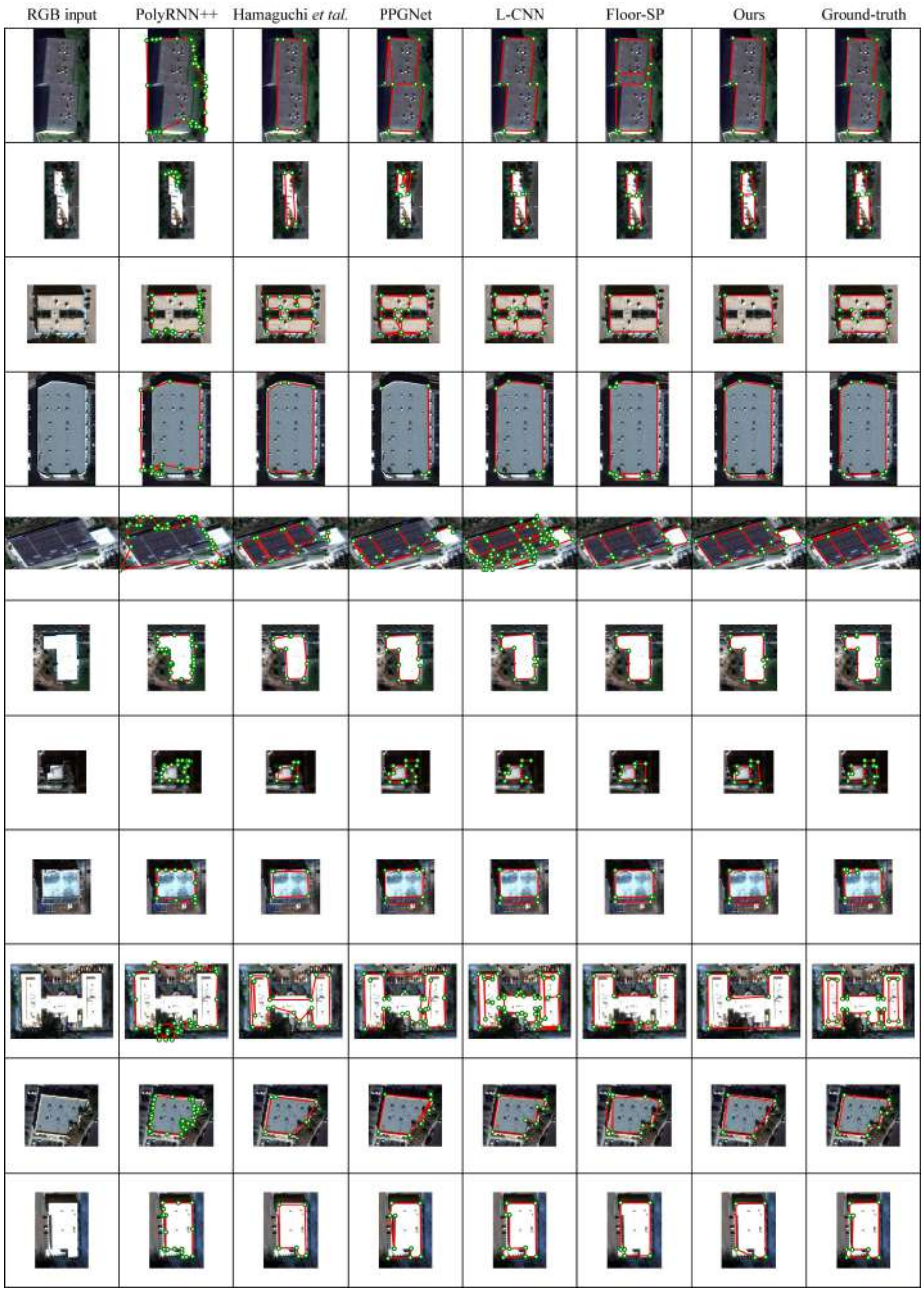

Fig. 9. Additional qualitative results.

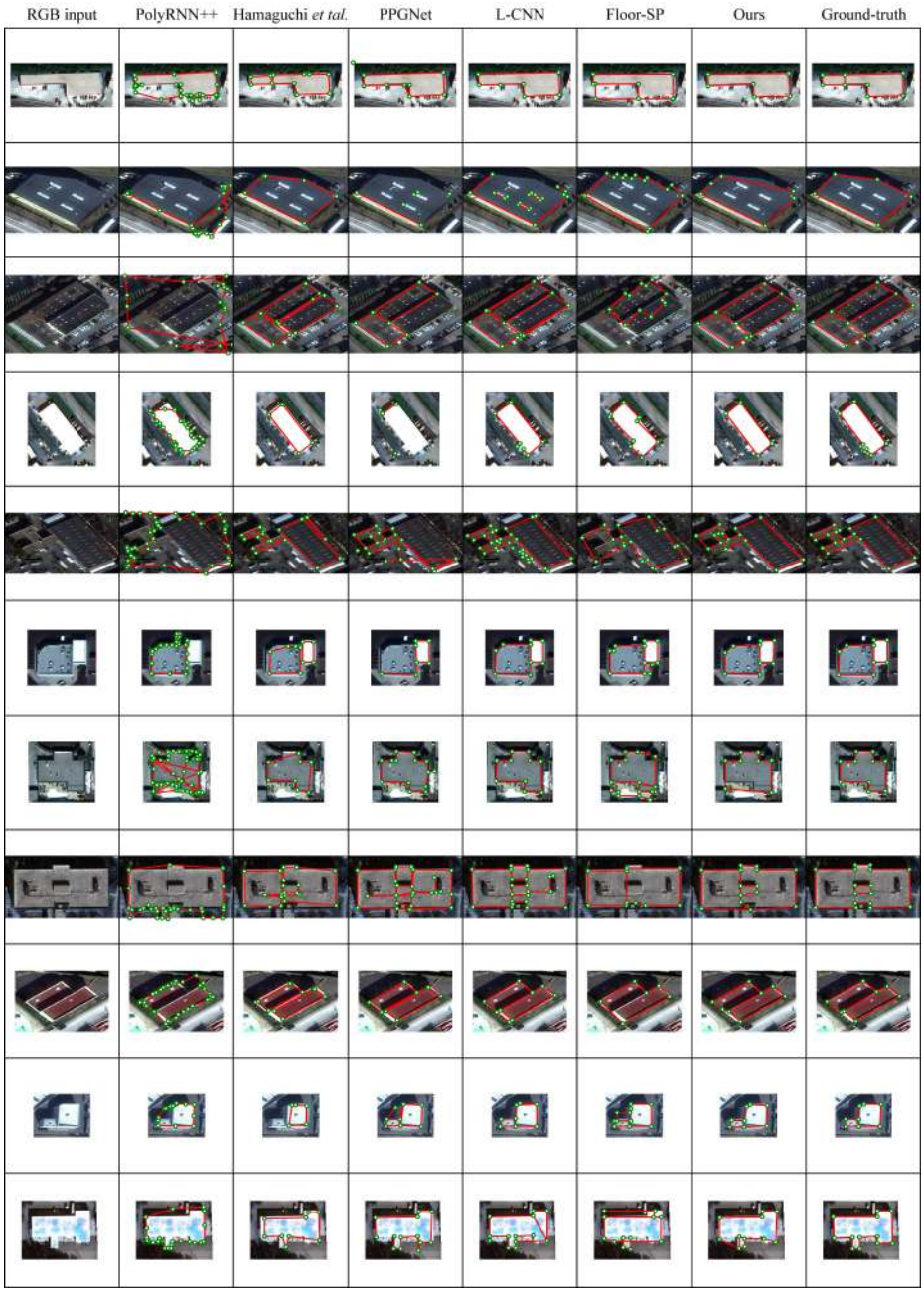

Fig. 10. Additional qualitative results.

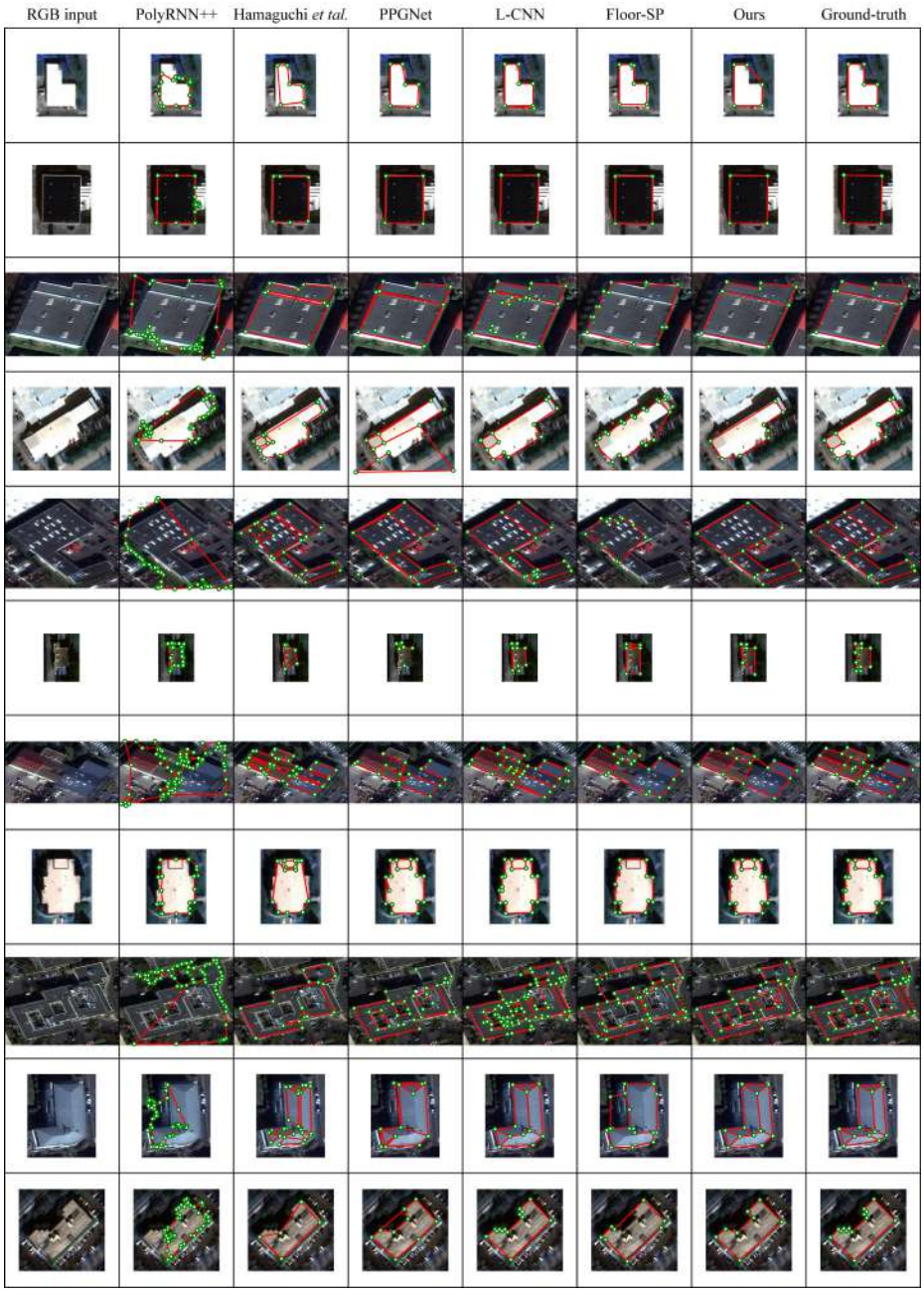

Fig. 11. Additional qualitative results.

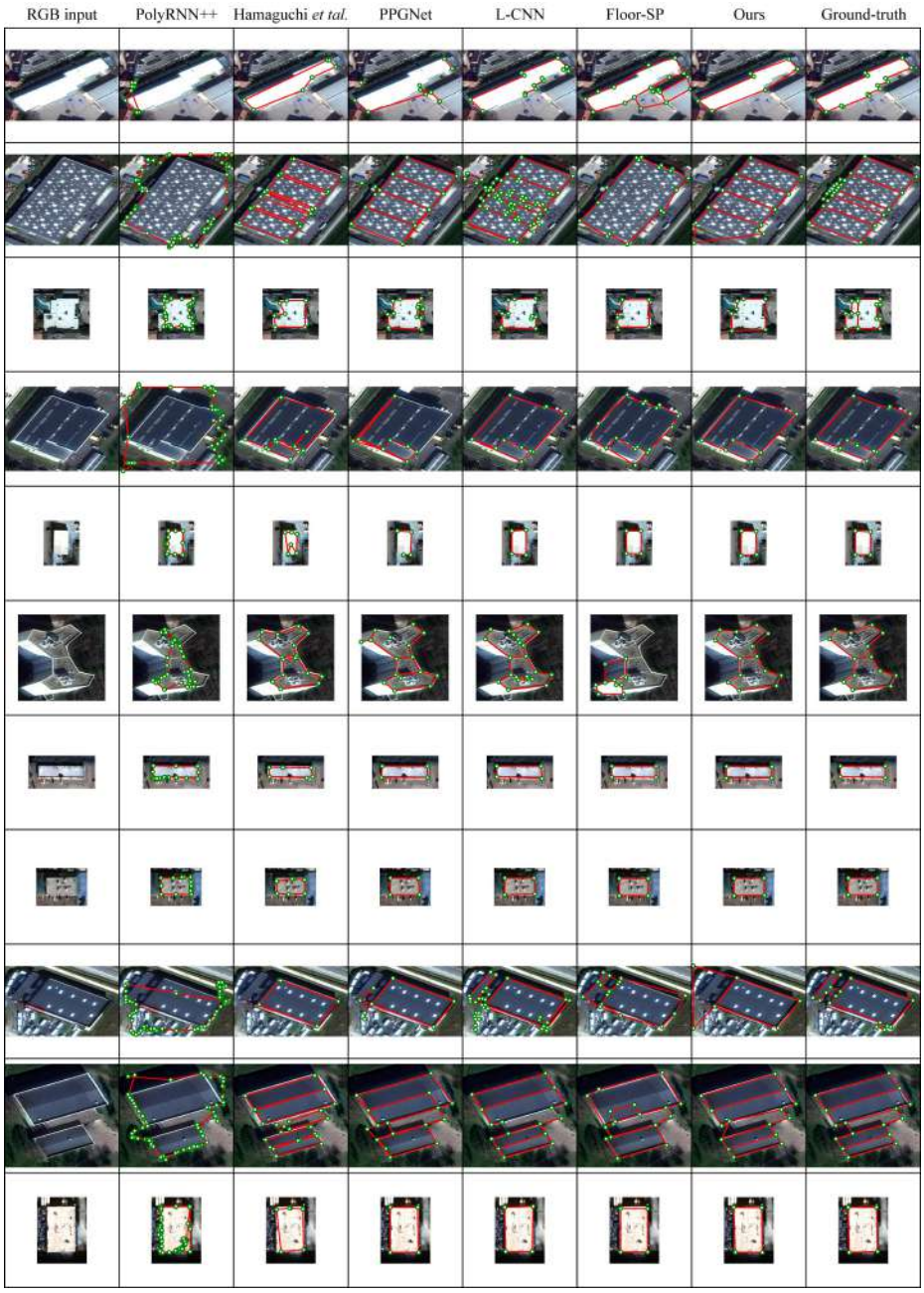

Fig. 12. Additional qualitative results.

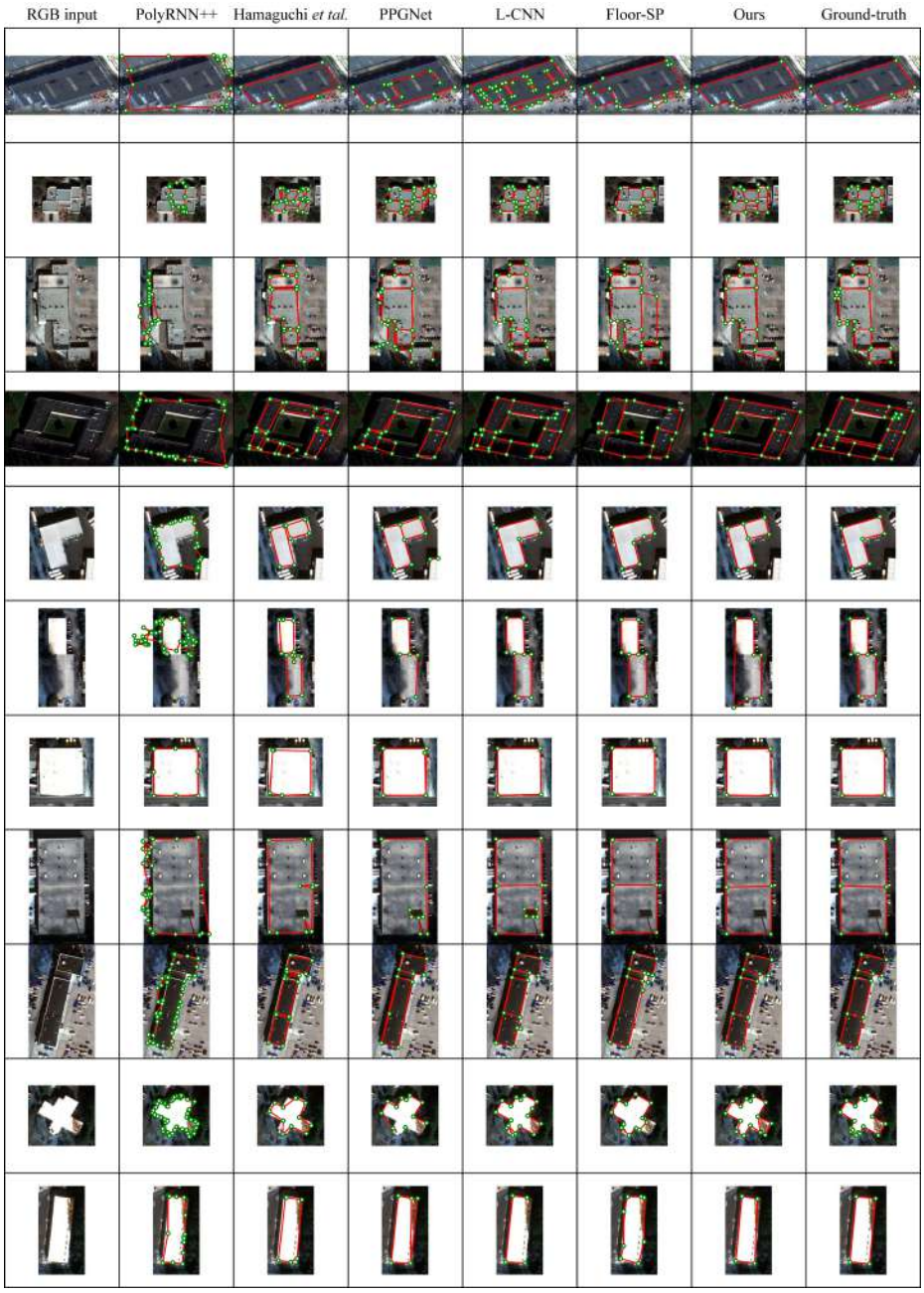

Fig. 13. Additional qualitative results.

Fig. 14. Additional qualitative results.

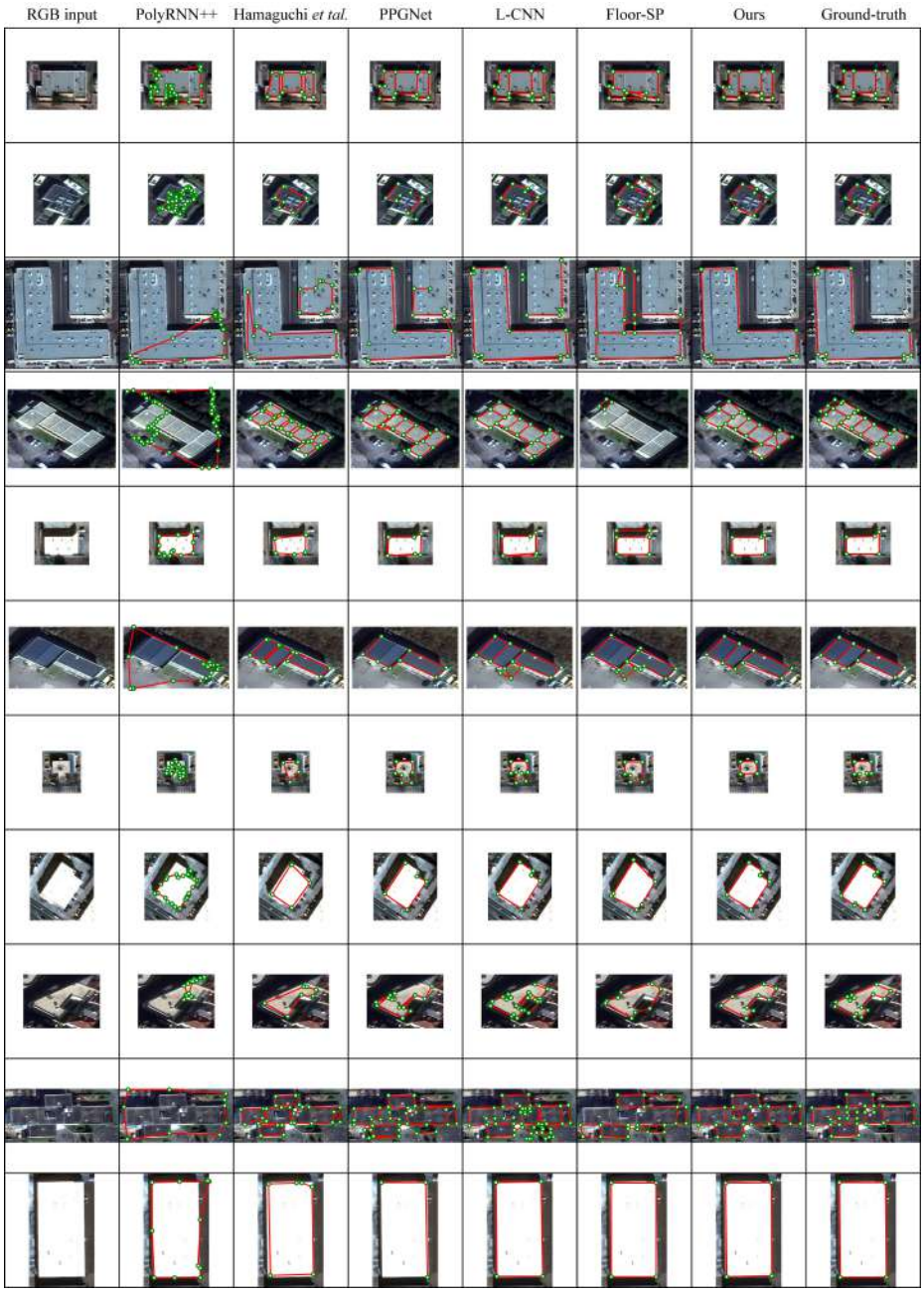

Fig. 15. Additional qualitative results.

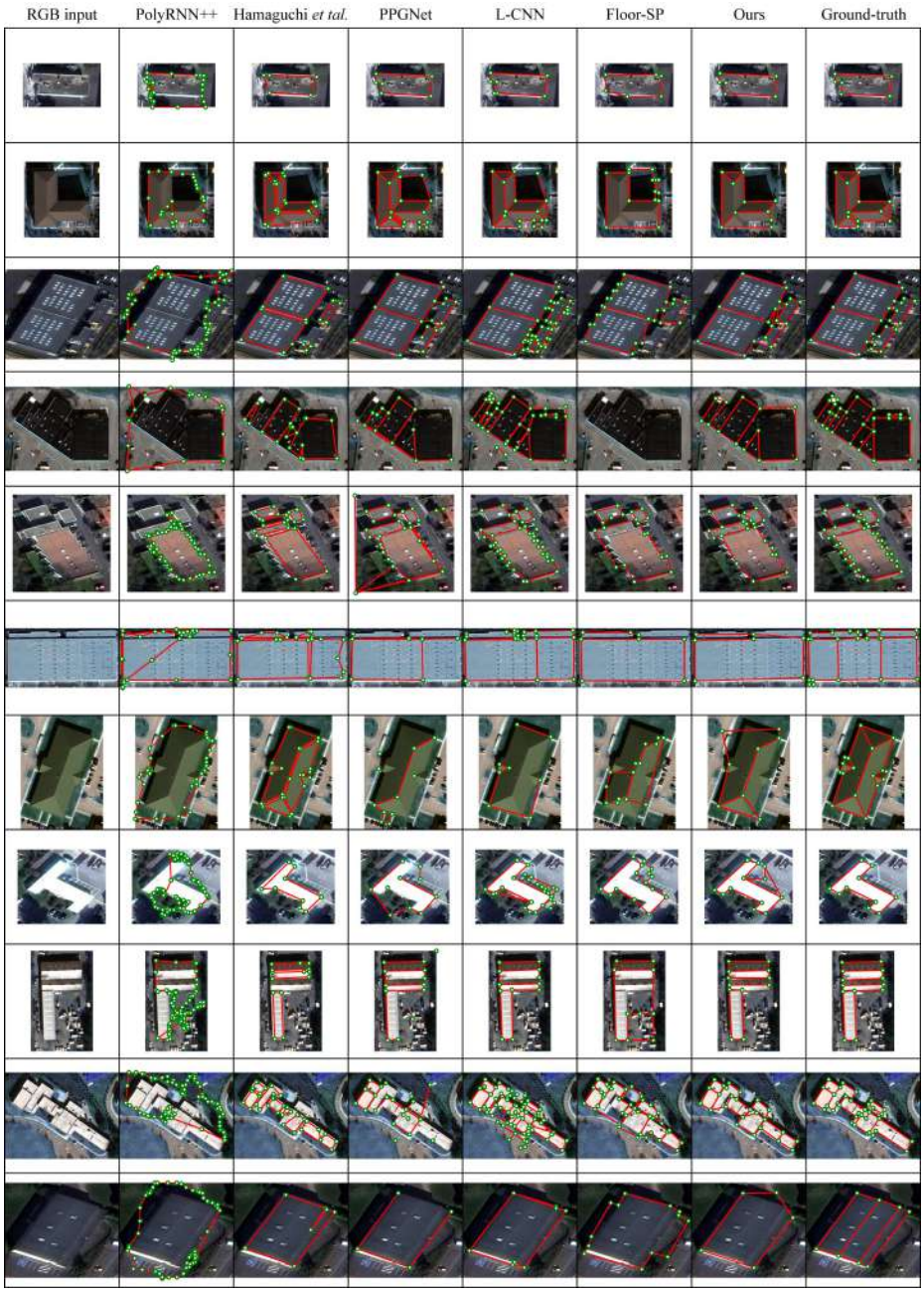

Fig. 16. Additional qualitative results.

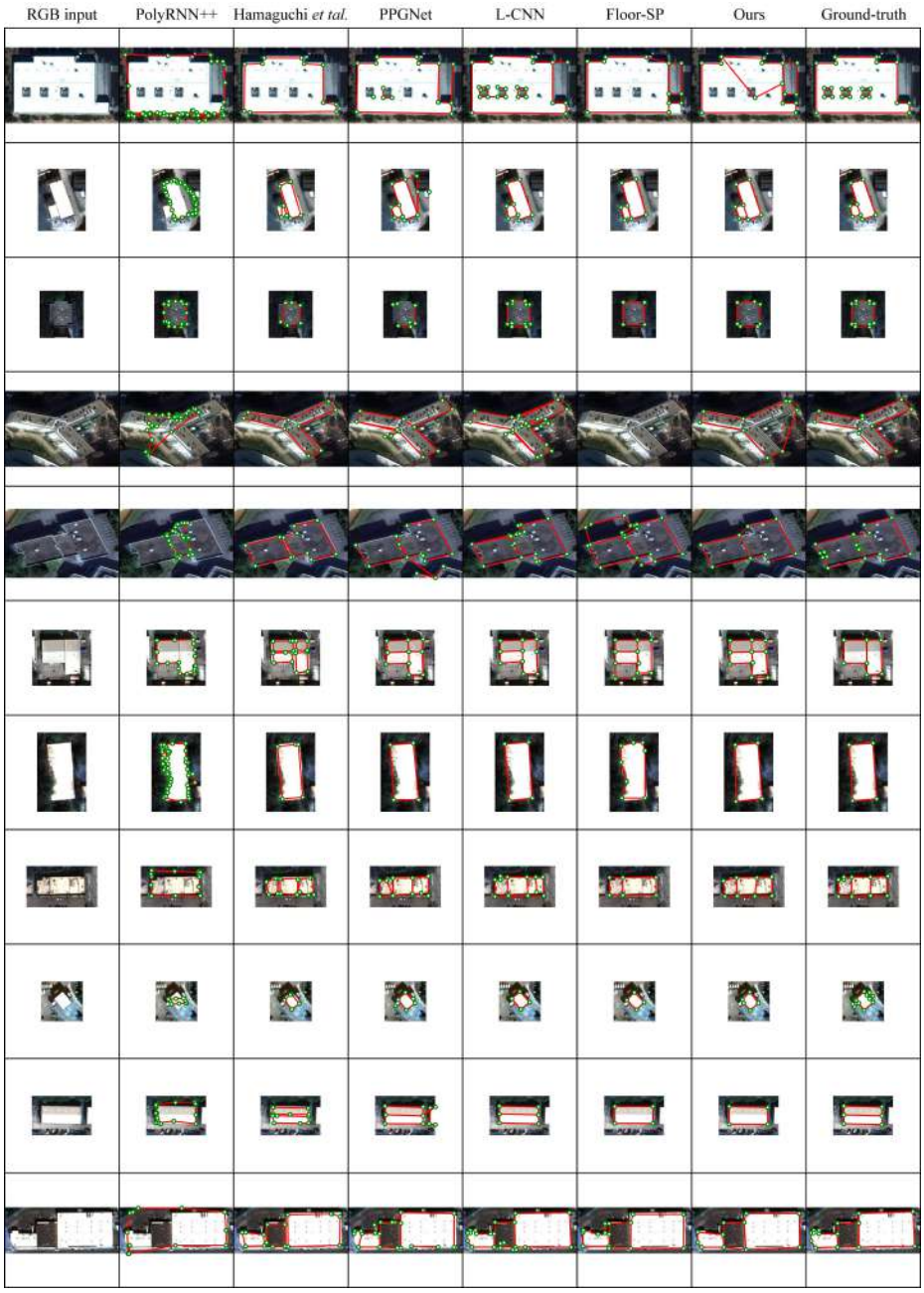

Fig. 17. Additional qualitative results.

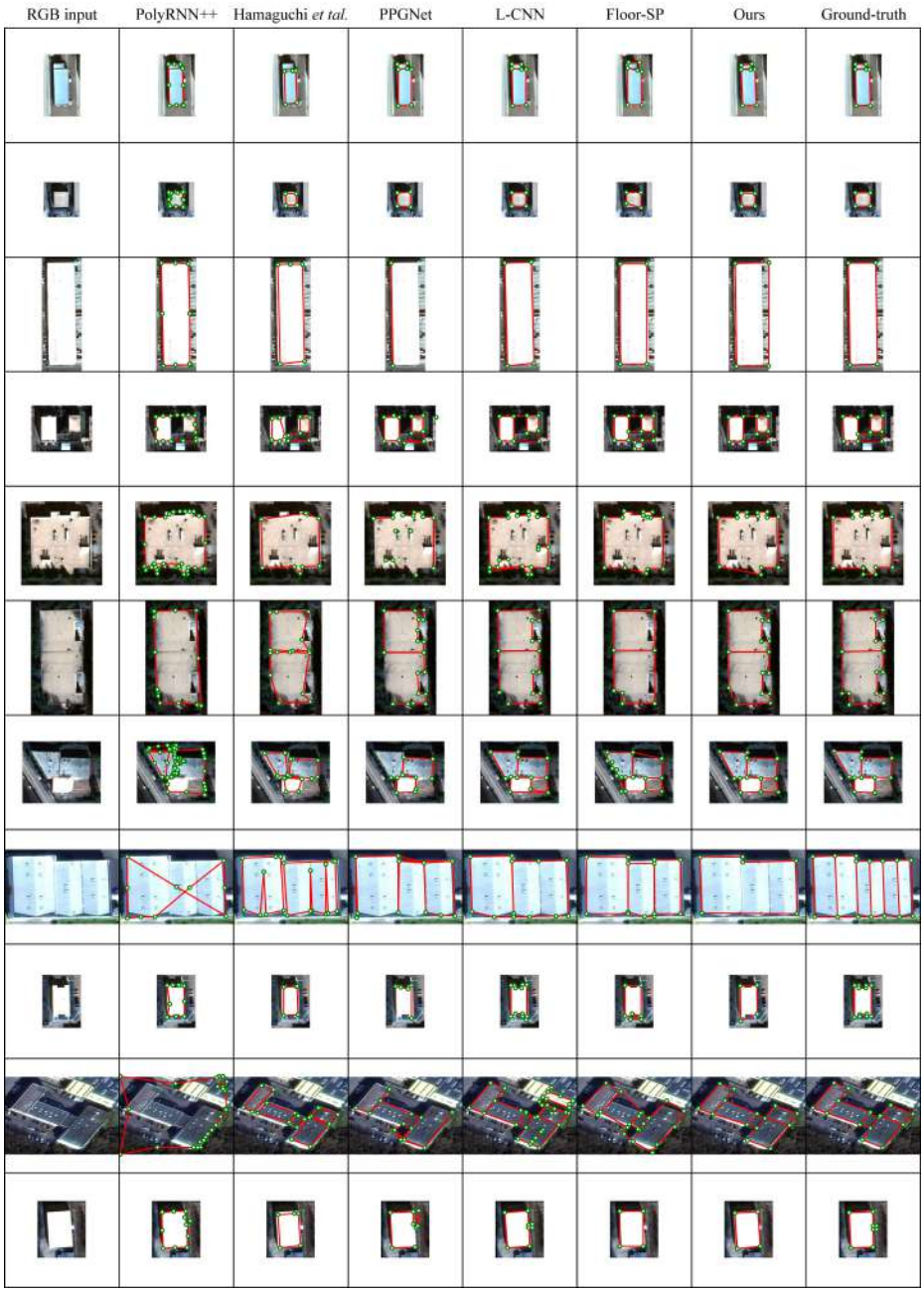

Fig. 18. Additional qualitative results.

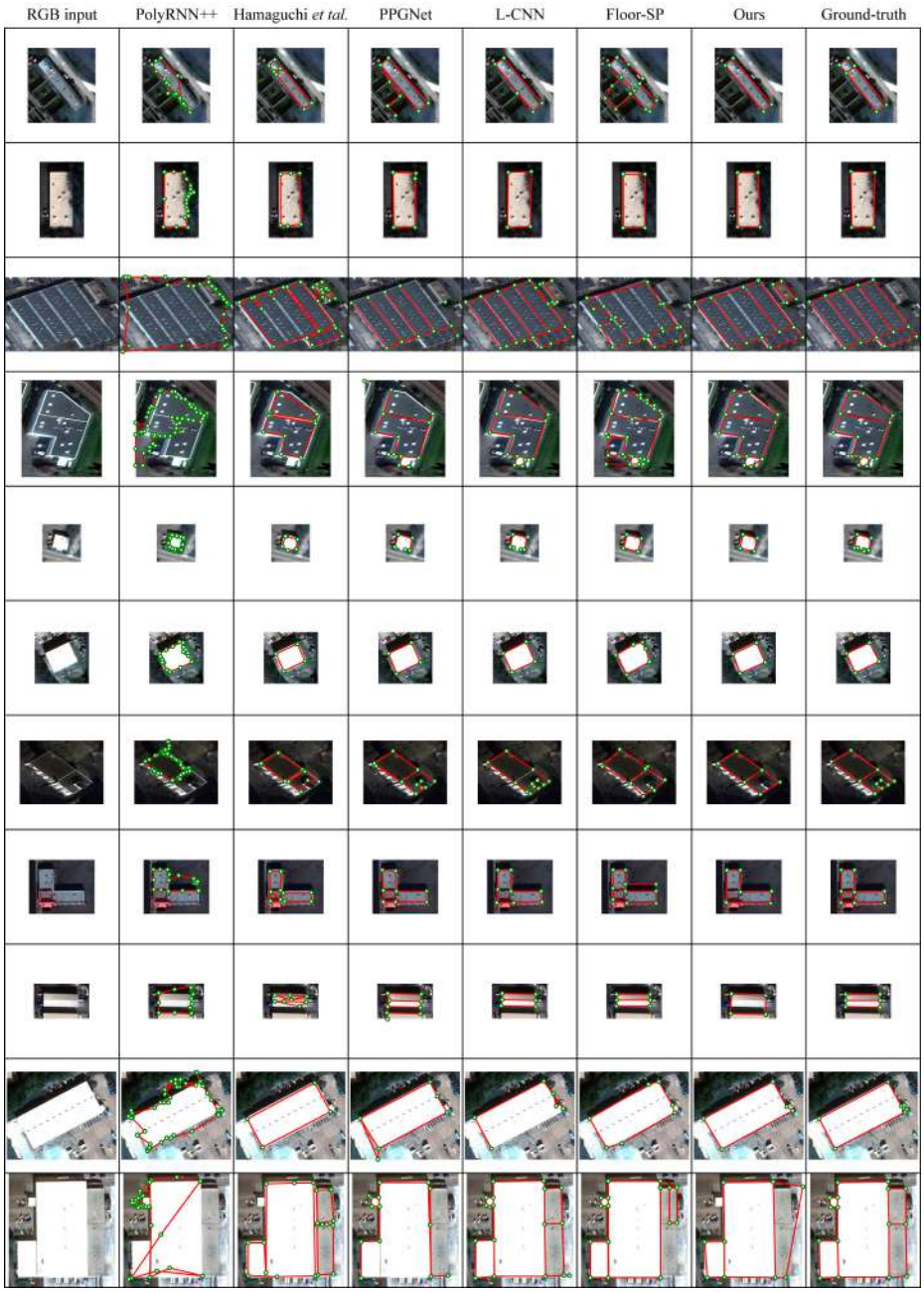

Fig. 19. Additional qualitative results.

Fig. 20. Additional qualitative results.

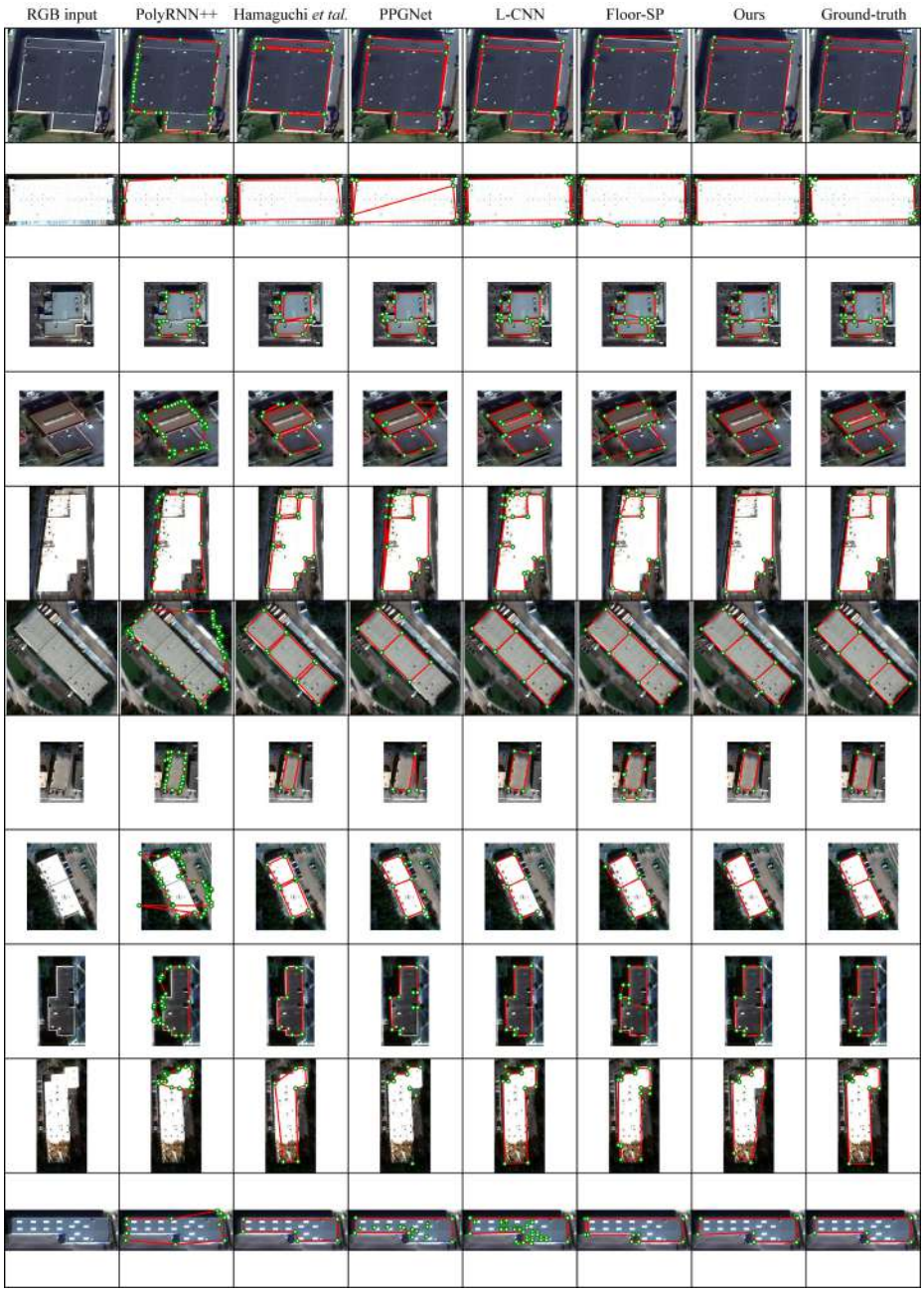

Fig. 21. Additional qualitative results.

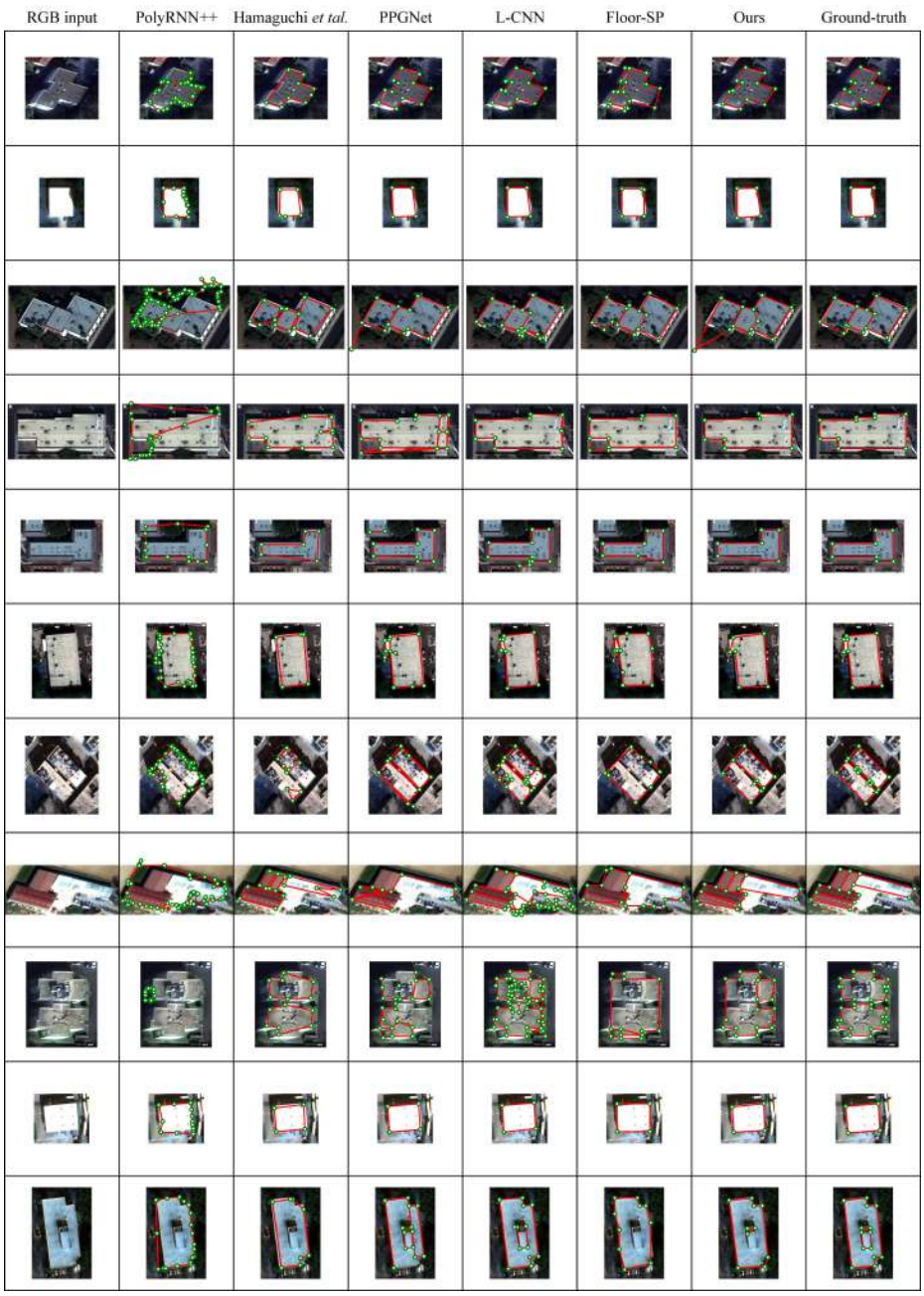

Fig. 22. Additional qualitative results.

Fig. 23. Additional qualitative results.

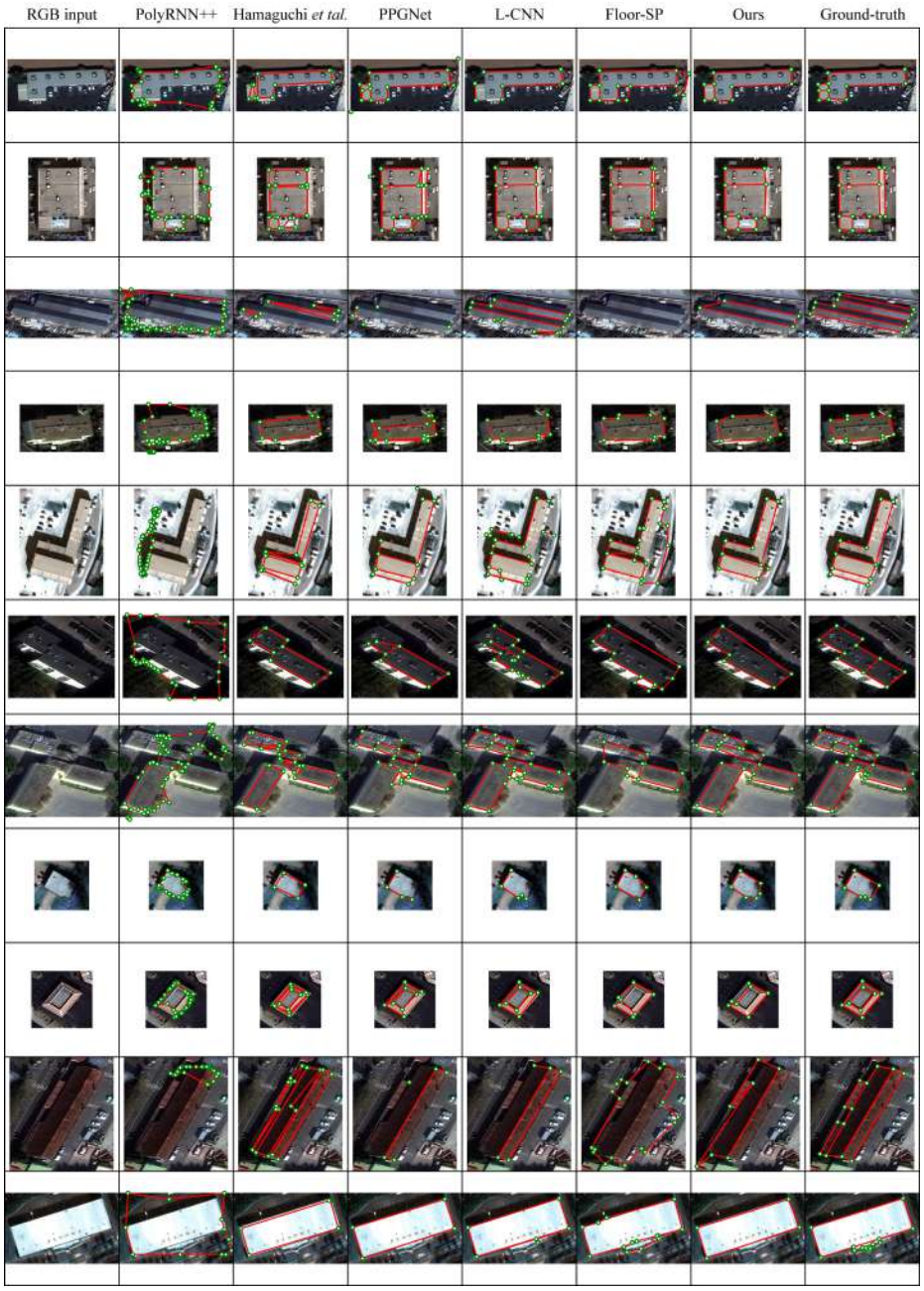

Fig. 24. Additional qualitative results.

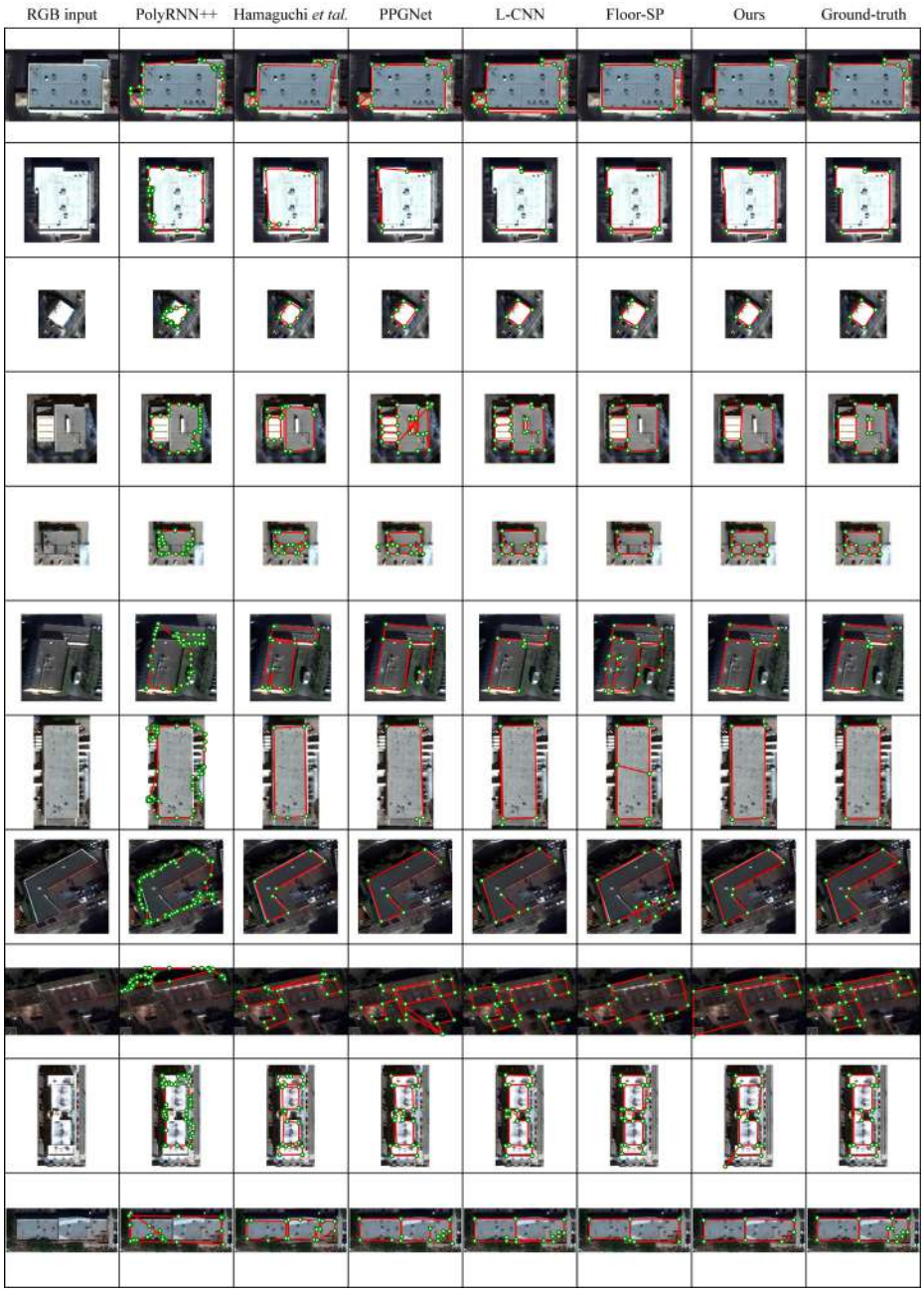

Fig. 25. Additional qualitative results.

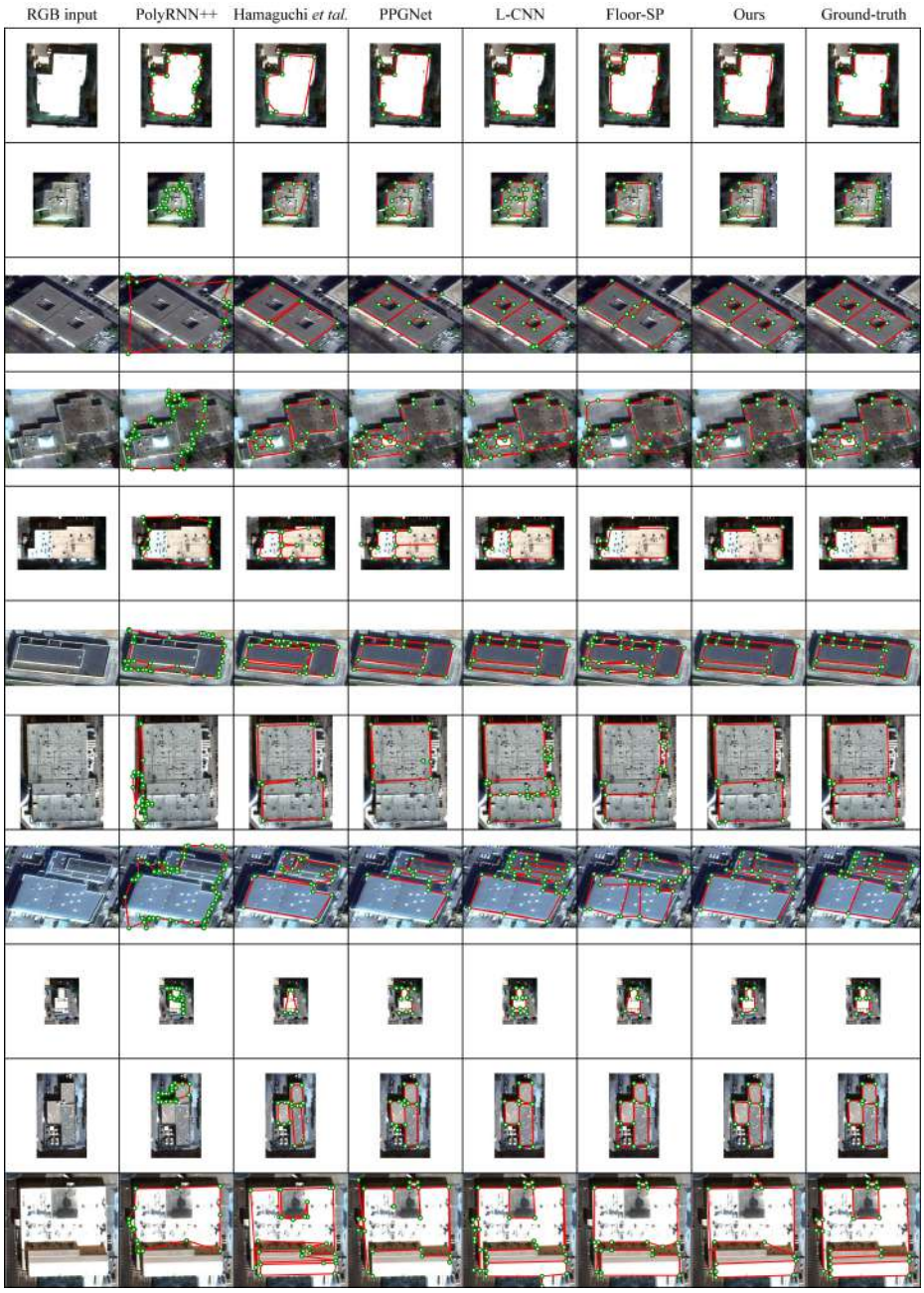

Fig. 26. Additional qualitative results.

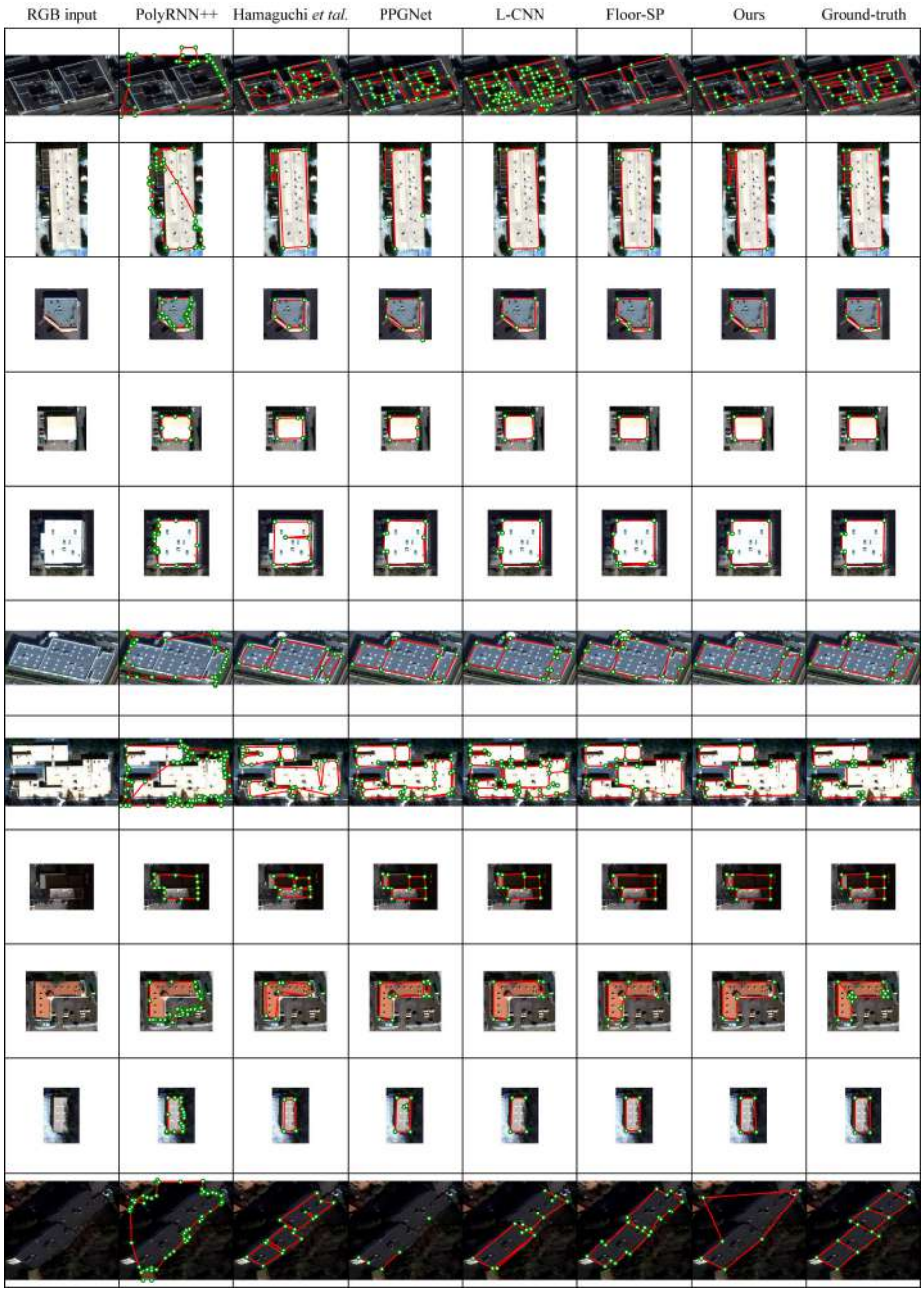

Fig. 27. Additional qualitative results.

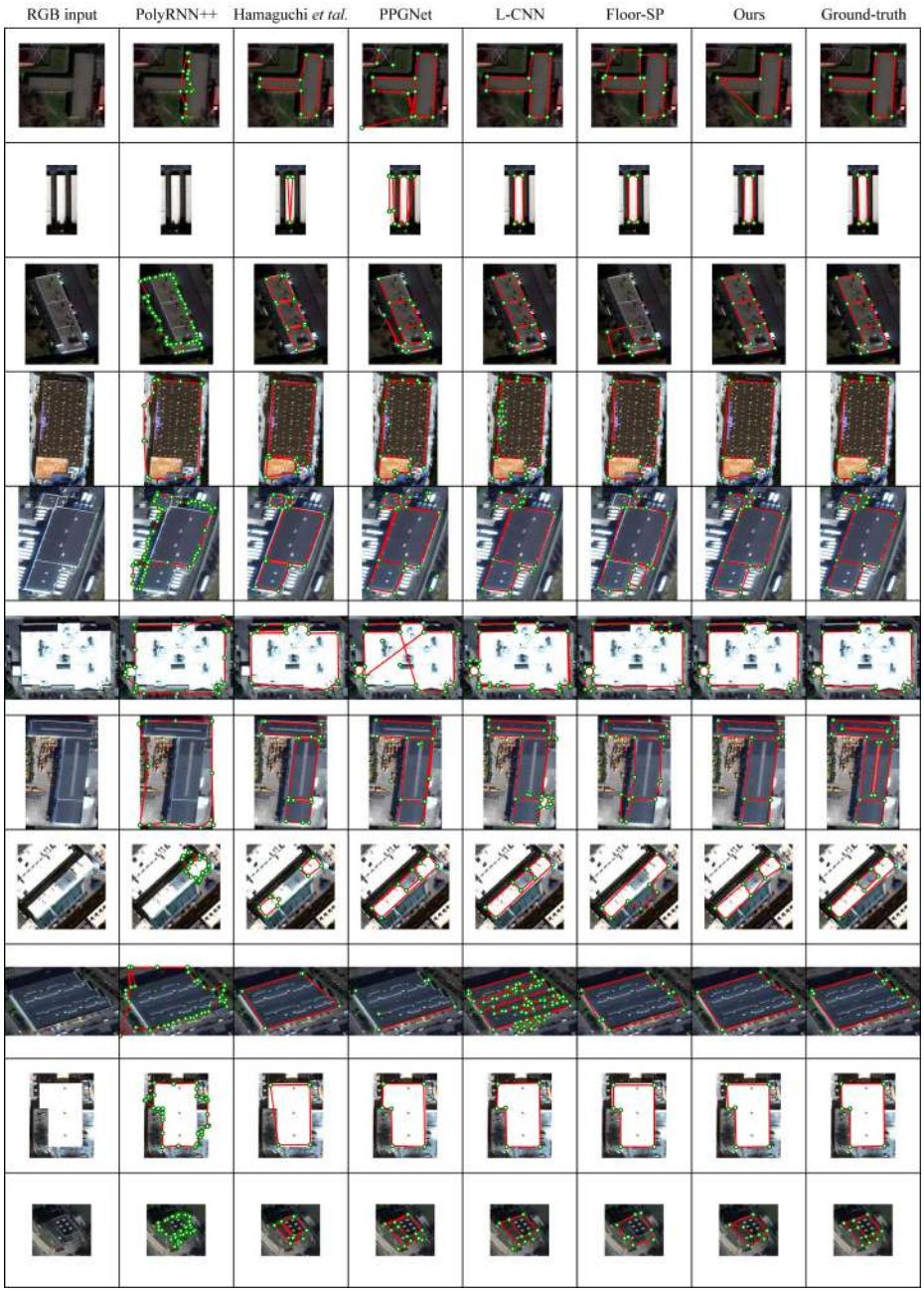

Fig. 28. Additional qualitative results.

Fig. 29. Additional qualitative results.

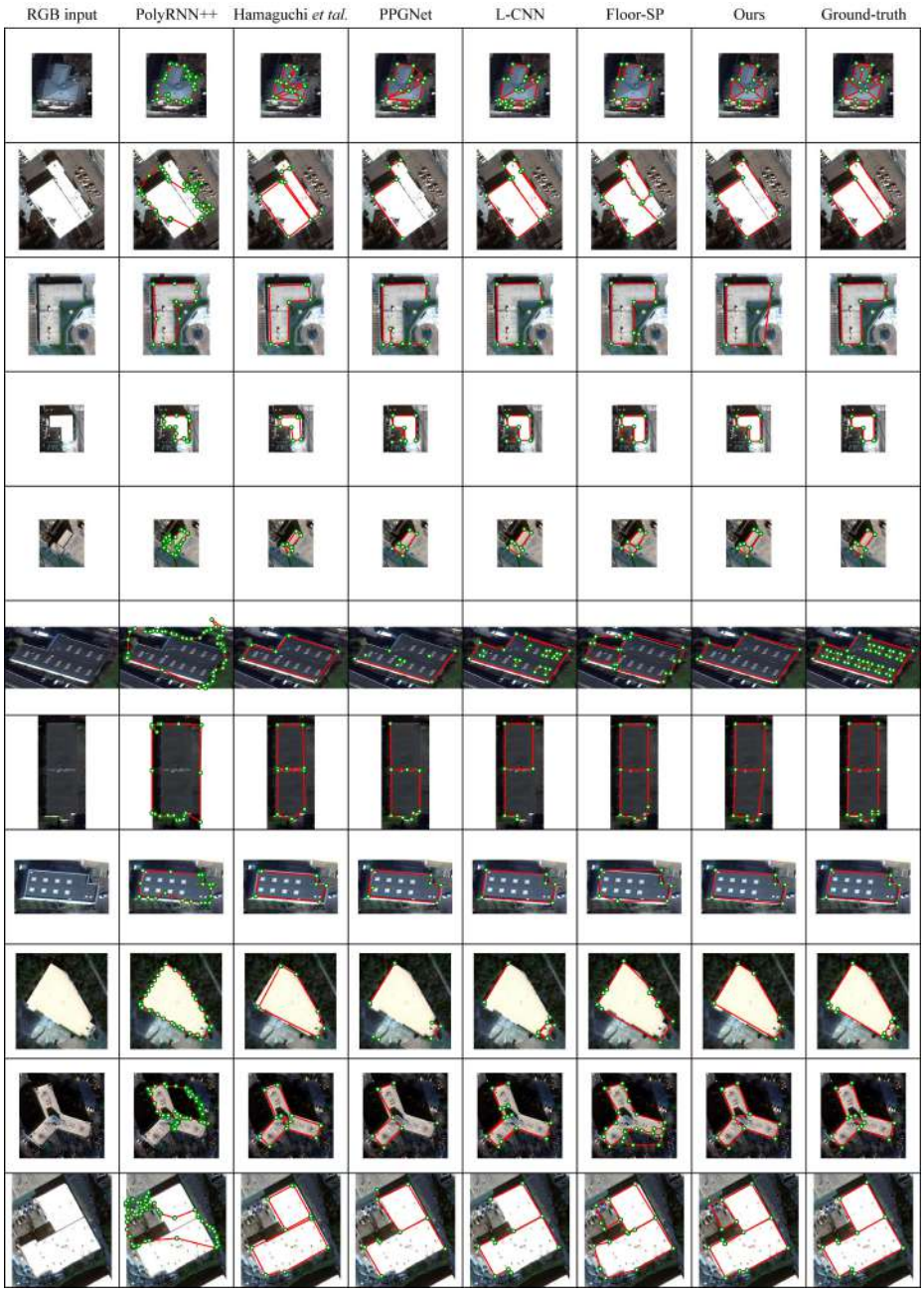

Fig. 30. Additional qualitative results.

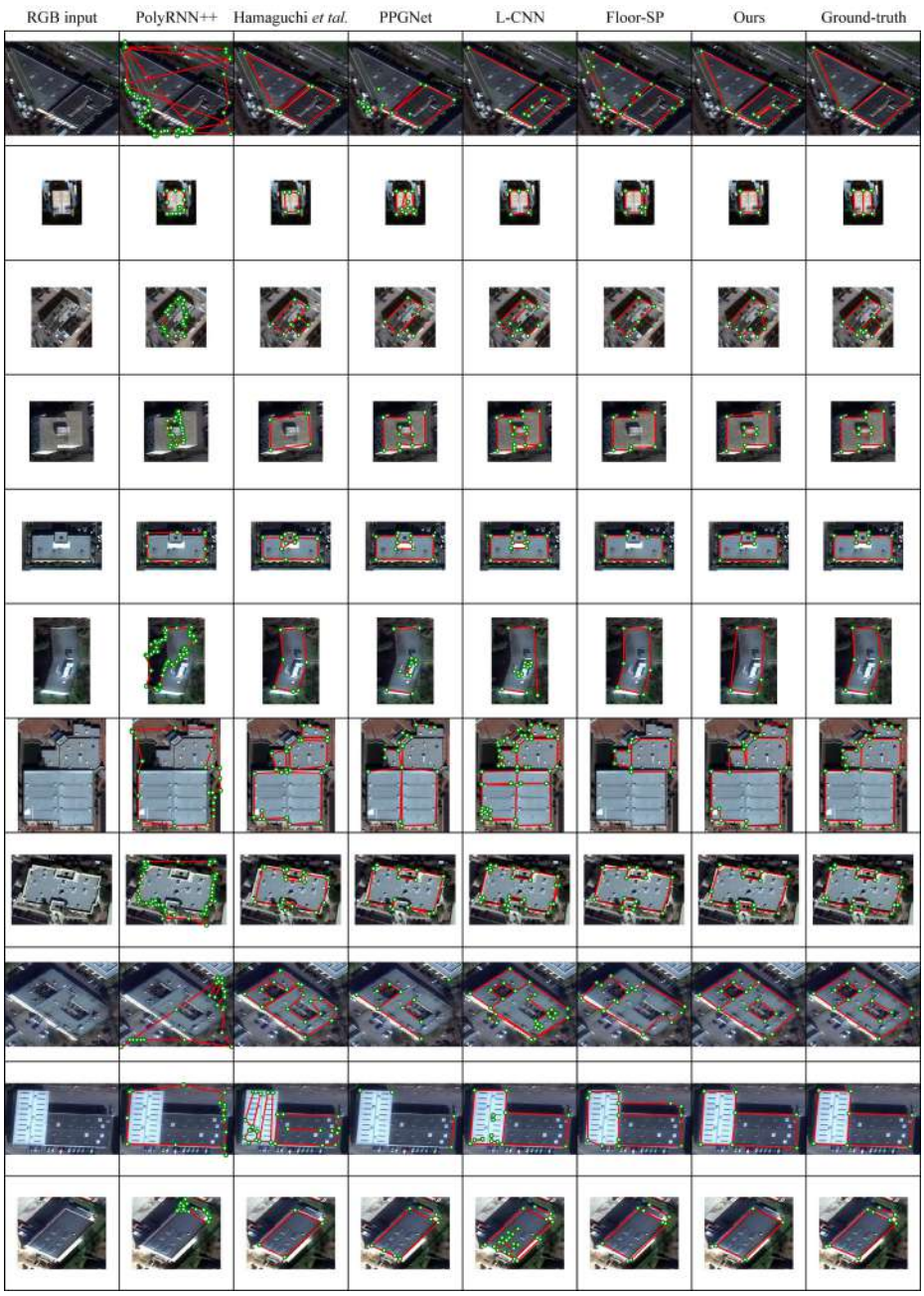

Fig. 31. Additional qualitative results.

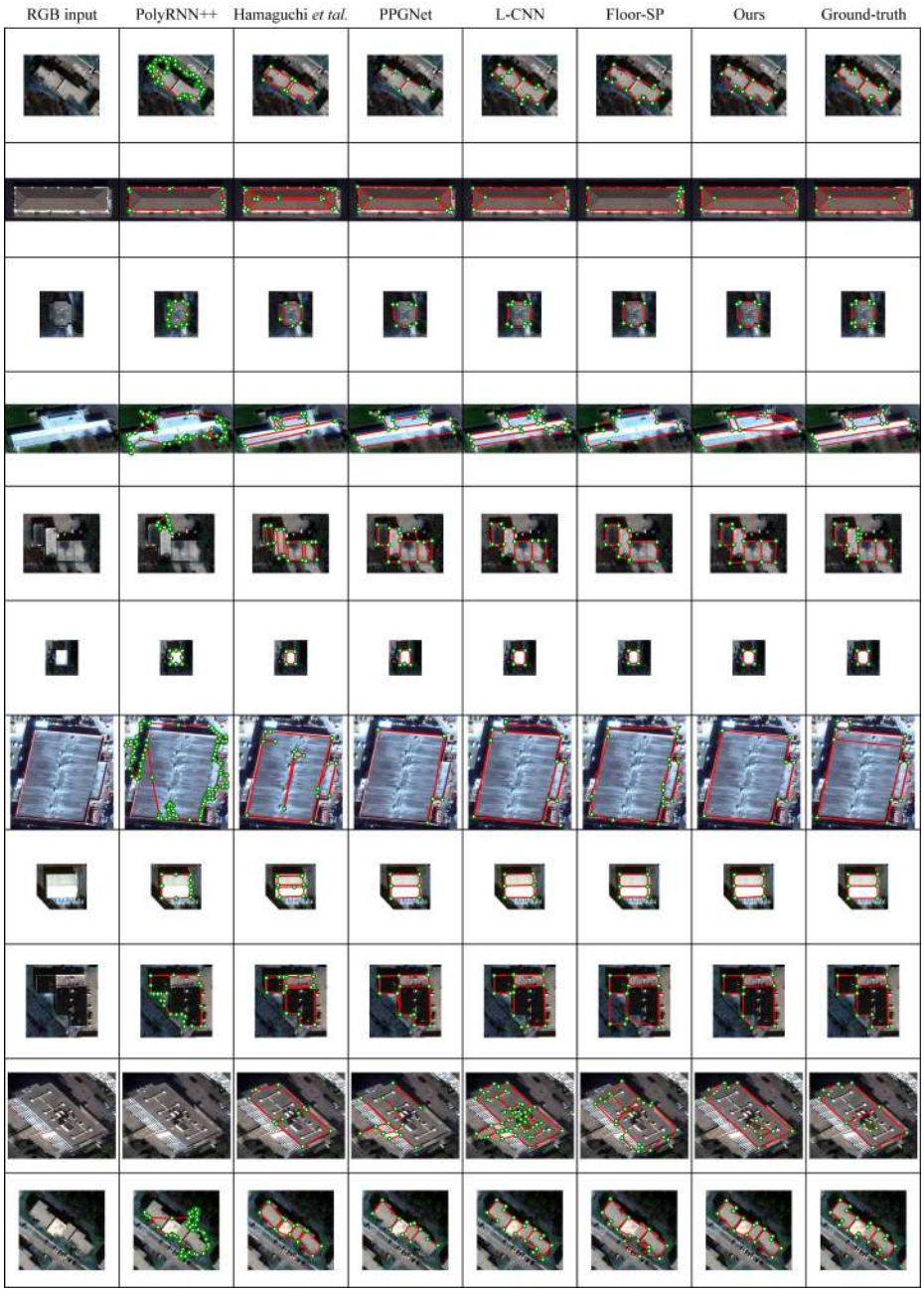

Fig. 32. Additional qualitative results.

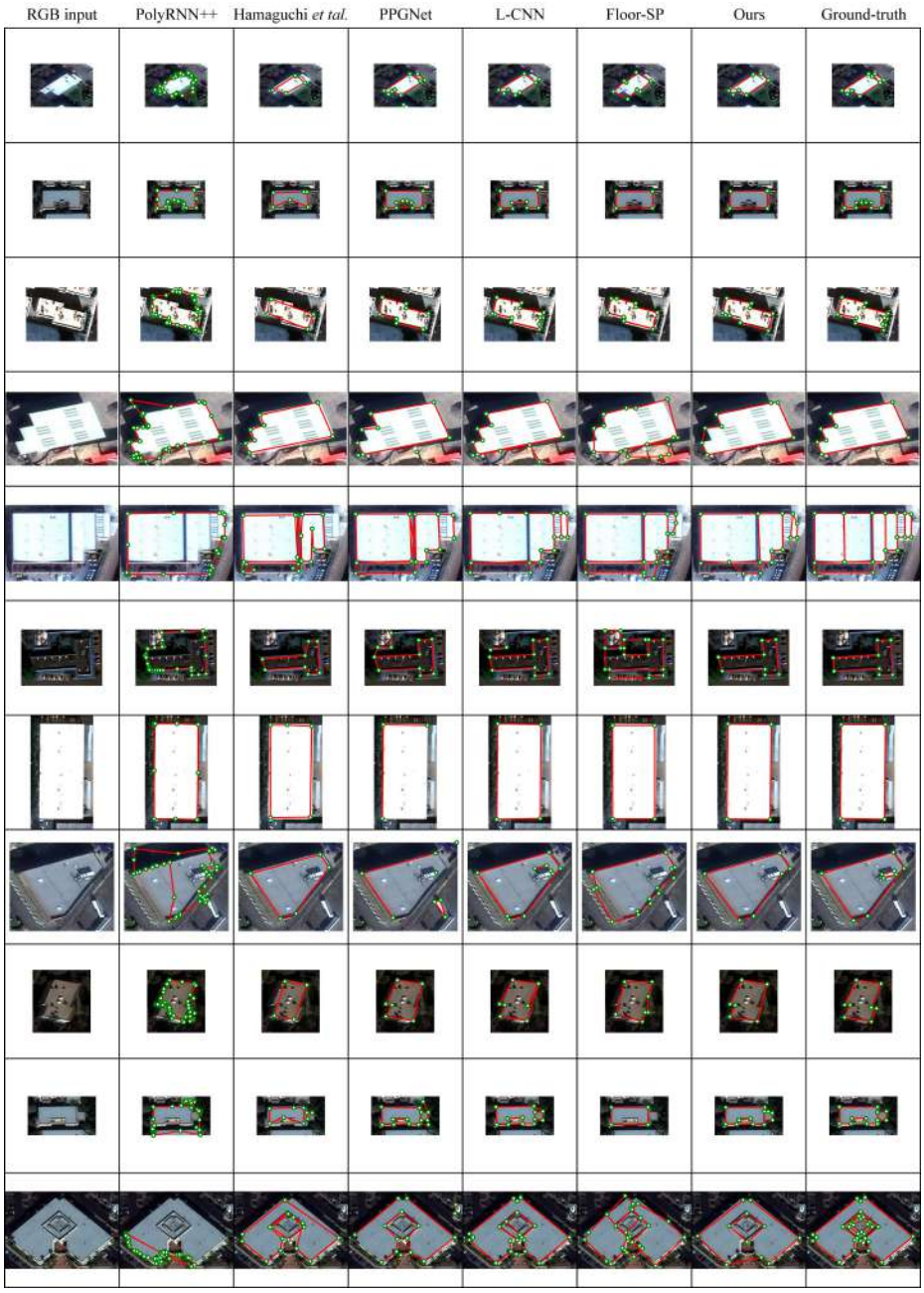

Fig. 33. Additional qualitative results.

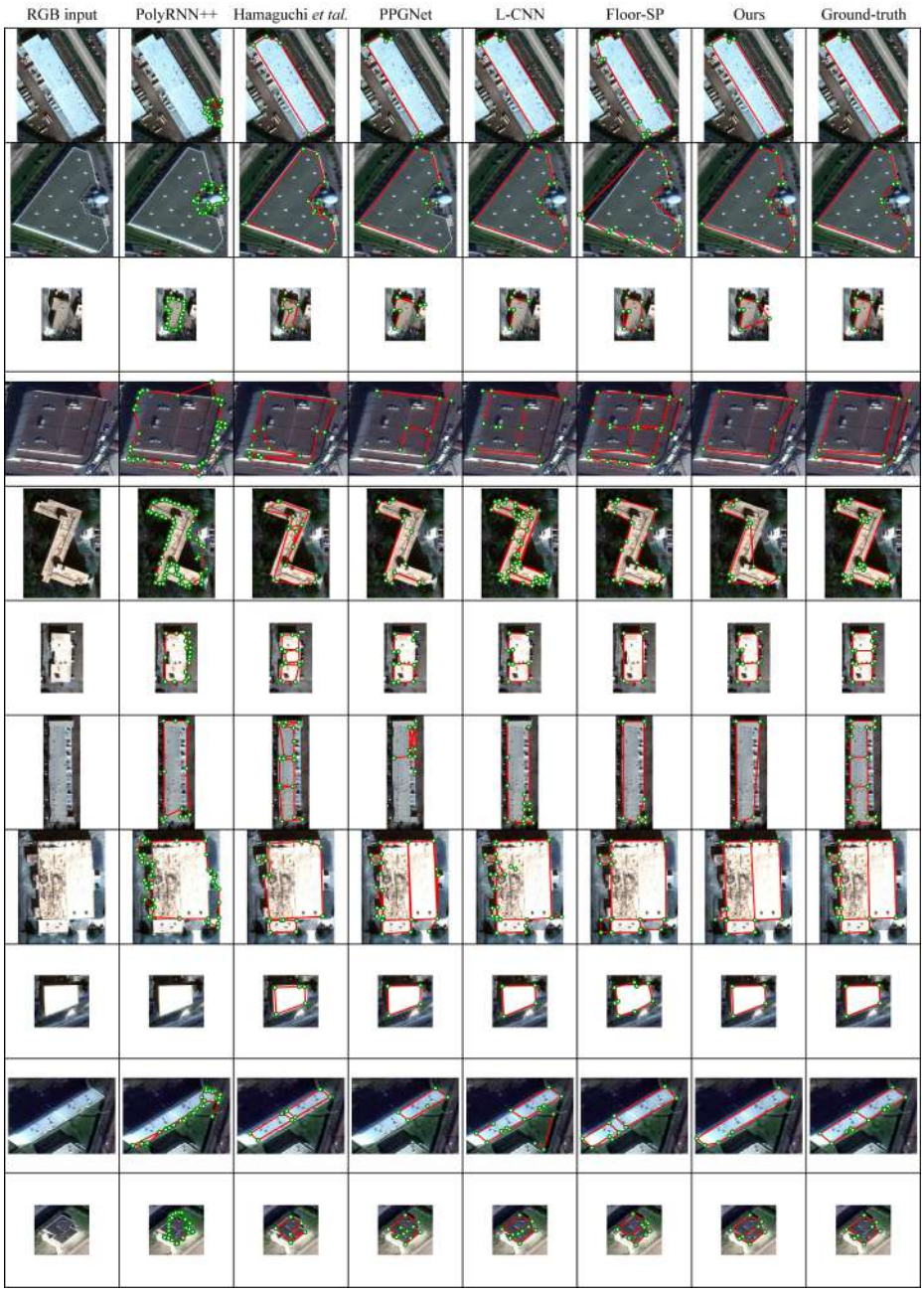

Fig. 34. Additional qualitative results.

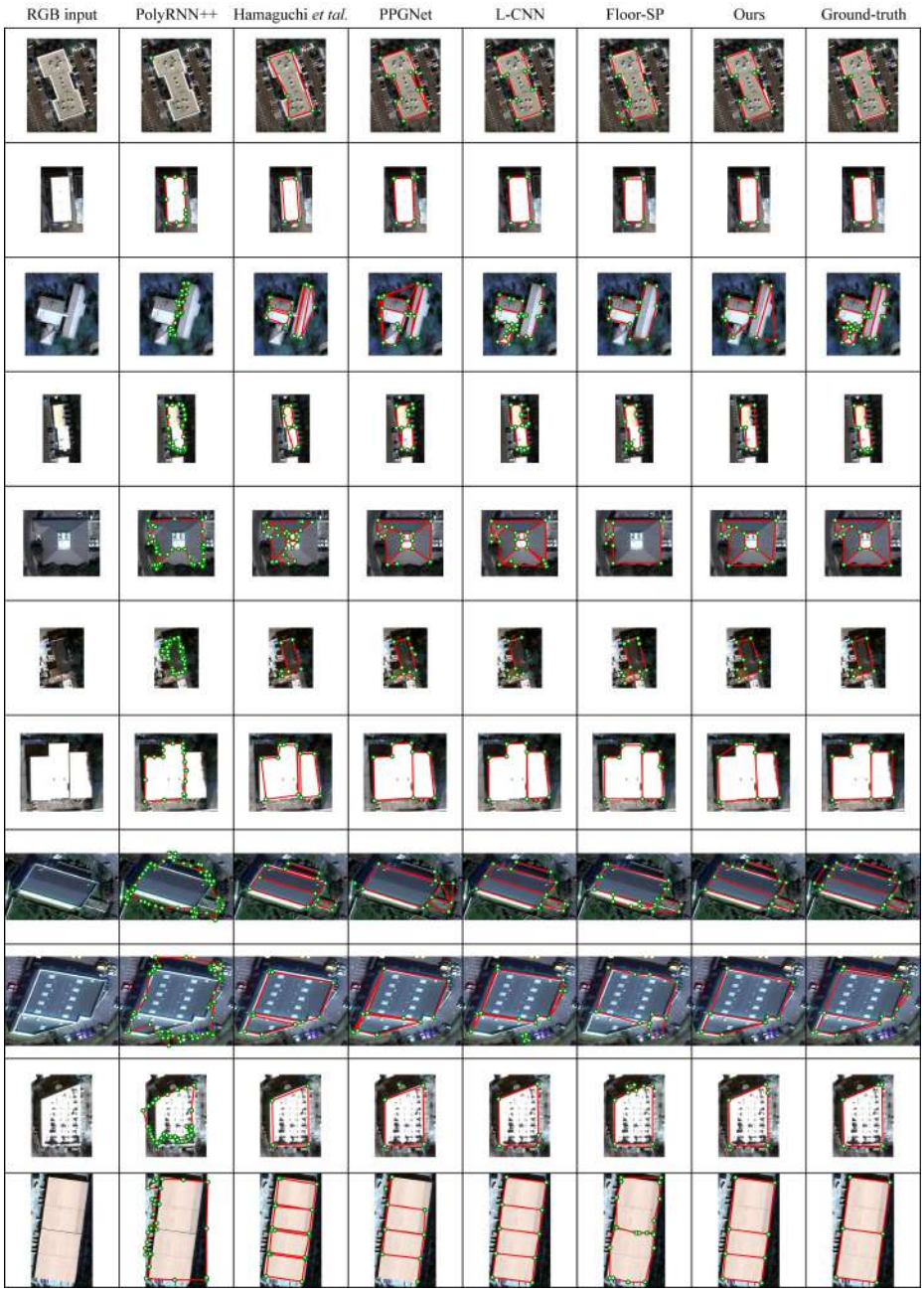

Fig. 35. Additional qualitative results.

References

1. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Computer Vision (ICCV), 2017 IEEE International Conference on. pp. 2980–2988. IEEE (2017)
2. Huang, K., Wang, Y., Zhou, Z., Ding, T., Gao, S., Ma, Y.: Learning to parse wireframes in images of man-made environments. In: CVPR (June 2018)
3. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
4. Yu, F., Koltun, V., Funkhouser, T.A.: Dilated residual networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 636–644 (2017)