[Supplementary Material] Conv-MPN: Convolutional Message Passing Neural Network for Structured Outdoor Architecture Reconstruction

The supplementary document provides 1) the full specification of neural architectures; 2) the details of the training data preparation process; and 3) additional results of all testing samples (400 buildings) against the six baselines.

1. Neural architecture specifications

Table 1 shows the full specification of the three neural modules constituting the Conv-MPN architecture. In the comparative evaluations, we used a vanilla graph neural network as a baseline, whose specification is given in Table 2.

2. Training data preparation

Conv-MPN requires building corner detection as a preprocessing, making it impossible to generate a ground-truth planar graph when some corners are missed. Therefore, we generate two sets of training data, one using ground-truth corners and the other using detected corners. We trained Conv-MPN (and its variants) by alternating these two sets of training data until convergence.

From ground-truth corners: We use the ground-truth building corners and the planar-graph as the input and output of Conv-MPN. We added small perturbations (i.e. $\mathcal{N}(0, 2^2)$ to corner (x, y) coordinates for better generalization.

From detected corners: Target corners are determined by assigning each detected corner candidate to the closest ground-truth corner as long as they are within 7 pixels distant from each other. Now, considering every pair of detected corners, we utilize the corner detection and annotation assignment to derive positive and negative targets for edges. A candidate edge is set to be positive if (1) its endpoints map to an edge in the annotations or (2) if there is corner that is colinear and a common neighbour to its endpoints. Worth to mention that we only apply the latter case, if the annotated colinear corner is not assigned to any other detected corner (otherwise we would be allowing duplicated colinear edges). For all other cases not in (1) or (2), the candidate edge is set to be a negative target.

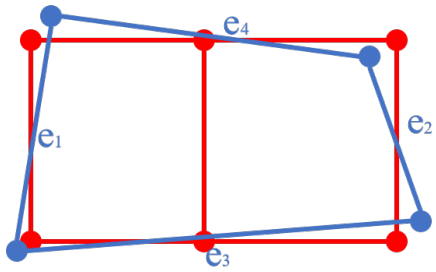

Figure 1. Illustration of a sample training data preparation process. The red shows the ground-truth planar graph. The blue shows the one generated from detected corners. Even with missed building corners, the process is able to generate a reasonable building graph.

3. Additional experimental results

Figures 2- 37 show additional experimental results with the six baseline methods over all testing samples.

References

Table 1. Conv-MPN architecture specification. For the feature initialization module, the three blocks are initialized with the first three blocks of DRN-C-26, which was pretrained on the ImageNet. We max-pool features of all the neighbor nodes and concatenate it with node feature as input of the message passing. $[\cdot]$ represents residual block. $\cdot\} \times i$ denotes repeating the layers i times. t denotes number of convolutional message passing iterations.

Module	Stage	Specification	Output Size
Feature initialization	$conv\_relu\_bn_1$	$4 \times 7 \times 7, 16, stride = 1$	$16 \times 256 \times 256$
	$residual\_block_1$	$\begin{bmatrix} 16 \times 3 \times 3, 16, stride = 1 \\ 16 \times 3 \times 3, 16, stride = 1 \end{bmatrix} \times 1$	$16 \times 256 \times 256$
	$residual\_block_2$	$\begin{bmatrix} 16 \times 3 \times 3, 32, stride = 2 \\ 32 \times 3 \times 3, 32, stride = 1 \end{bmatrix} \times 1$	$32 \times 128 \times 128$
	$residual\_block_3$	$\begin{bmatrix} 32 \times 3 \times 3, 64, stride = 2 \\ 64 \times 3 \times 3, 64, stride = 1 \end{bmatrix} \times 2$	$64 \times 64 \times 64$
	$conv\_relu\_bn_2$	$64 \times 3 \times 3, 32, stride = 1$	$32 \times 64 \times 64$
Convolutional message passing	$conv\_relu\_bn_3$ $conv\_relu\_bn_4$ $conv\_relu\_bn_5$	$\left.\begin{matrix} 64 \times 3 \times 3, 64, stride = 1 \ \} \times 4 \\ 64 \times 3 \times 3, 32, stride = 1 \ \} \times 1 \\ 32 \times 3 \times 3, 32, stride = 1 \ \} \times 2 \end{matrix}\right\} \times t$	$32 \times 64 \times 64$
Building edge verification	$conv\_relu\_bn_6$	$32 \times 3 \times 3, 32, stride = 1$	$128 \times 64 \times 64$
	$conv\_relu\_bn_7$	$32 \times 3 \times 3, 64, stride = 1$	
	$conv\_relu\_bn_8$	$64 \times 3 \times 3, 64, stride = 1$	
	$conv\_relu\_bn_9$	$64 \times 3 \times 3, 128, stride = 1$	
	$conv\_relu\_bn_{10}$	$128 \times 3 \times 3, 128, stride = 1$	
	$max\_pooling$	$32 \times 32, max, stride = 32$	$128 \times 2 \times 2$
	fc	512×2	2

Table 2. The architecture specification of Vanilla GNN, a baseline method in our comparative evaluations. The feature initialization module encodes each building edge into a 512d vector instead of a feature volume. The message passing module max-pools all the neighboring nodes and concatenate the pooled vector to generate an input of the fully connected layer. $[\cdot]$ represents residual block. $\cdot\} \times i$ denotes repeating the layers i times. t denotes number of vanilla message passing iterations.

Module	Stage	Specification	Output Size
Feature initialization	$conv\_relu\_bn_1$	$4 \times 7 \times 7, 16, stride = 1$	$16 \times 256 \times 256$
	$residual\_block_1$	$\begin{bmatrix} 16 \times 3 \times 3, 16, stride = 1 \\ 16 \times 3 \times 3, 16, stride = 1 \end{bmatrix} \times 1$	$16 \times 256 \times 256$
	$residual\_block_2$	$\begin{bmatrix} 16 \times 3 \times 3, 32, stride = 2 \\ 32 \times 3 \times 3, 32, stride = 1 \end{bmatrix} \times 1$	$32 \times 128 \times 128$
	$residual\_block_3$	$\begin{bmatrix} 32 \times 3 \times 3, 64, stride = 2 \\ 64 \times 3 \times 3, 64, stride = 1 \end{bmatrix} \times 2$	$64 \times 64 \times 64$
	$conv\_relu\_bn_2$ $conv\_relu\_bn_3$ $conv\_relu\_bn_4$ $conv\_relu\_bn_5$ $conv\_relu\_bn_6$ $conv\_relu\_bn_7$	$64 \times 3 \times 3, 32, stride = 1$ $32 \times 3 \times 3, 32, stride = 1$ $32 \times 3 \times 3, 64, stride = 1$ $64 \times 3 \times 3, 64, stride = 1$ $64 \times 3 \times 3, 128, stride = 1$ $128 \times 3 \times 3, 128, stride = 1$	$128 \times 64 \times 64$
	$max\_pooling$	$32 \times 32, max, stride = 32$	$128 \times 2 \times 2$
Message passing	$fc\_relu\_bn_8$ $fc\_relu\_bn_9$ $fc\_relu\_bn_{10}$	$\left.\begin{matrix} 1024 \times 1024 \ \} \times 3 \\ 1024 \times 512 \ \} \times 1 \\ 512 \times 512 \ \} \times 2 \end{matrix}\right\} \times t$	512
	fc	512×2	2

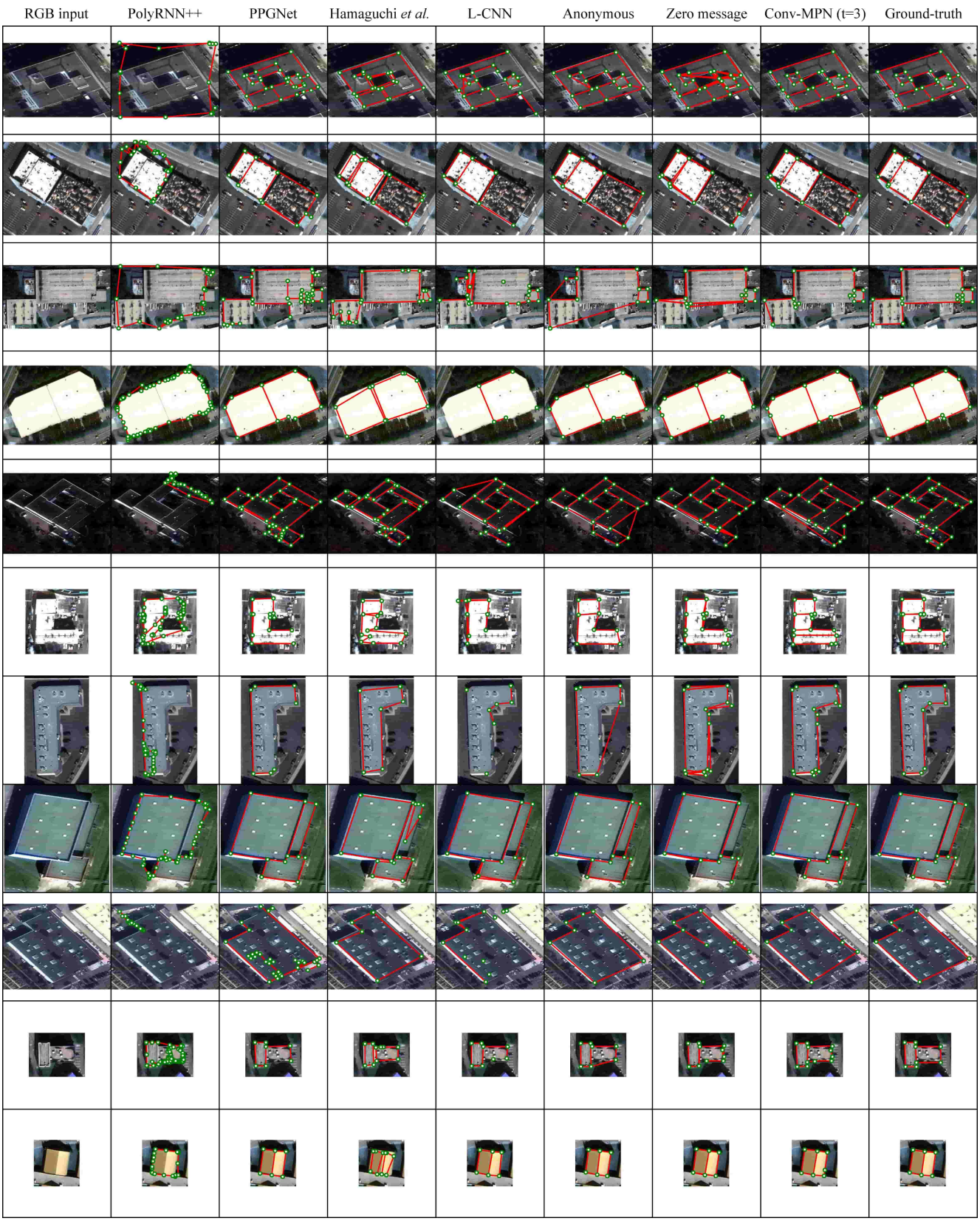

Figure 2. Additional qualitative results.

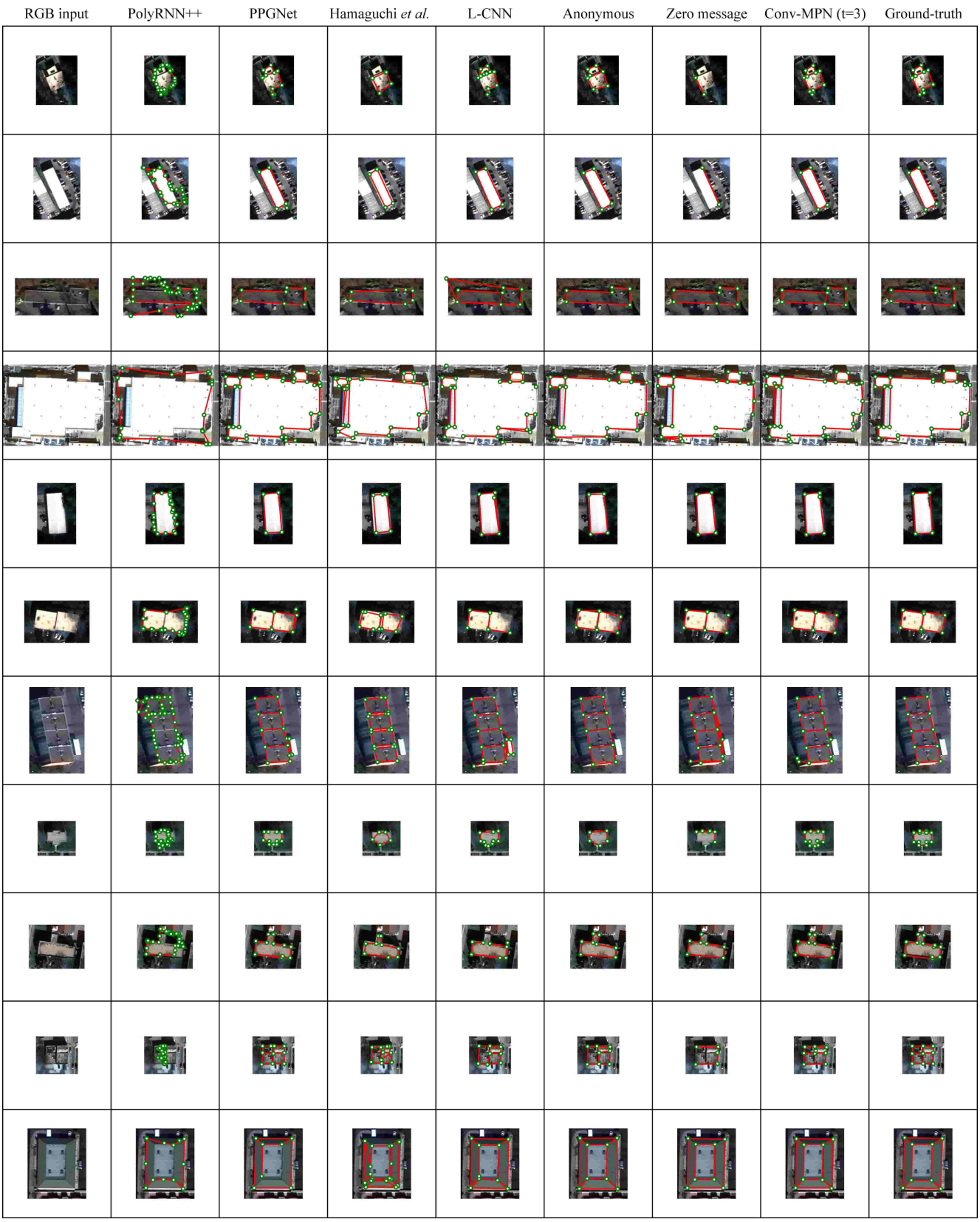

Figure 3. Additional qualitative results.

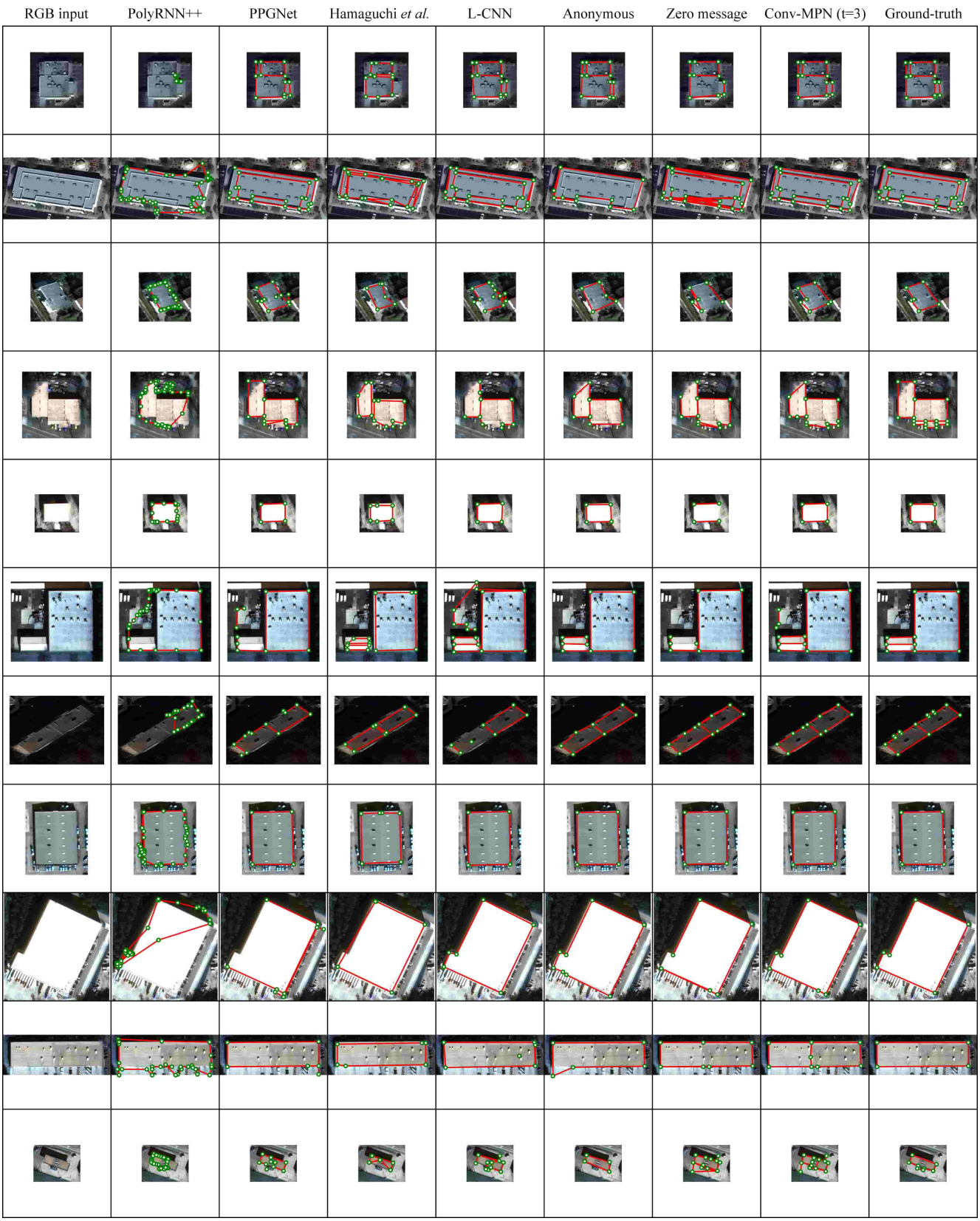

Figure 4. Additional qualitative results.

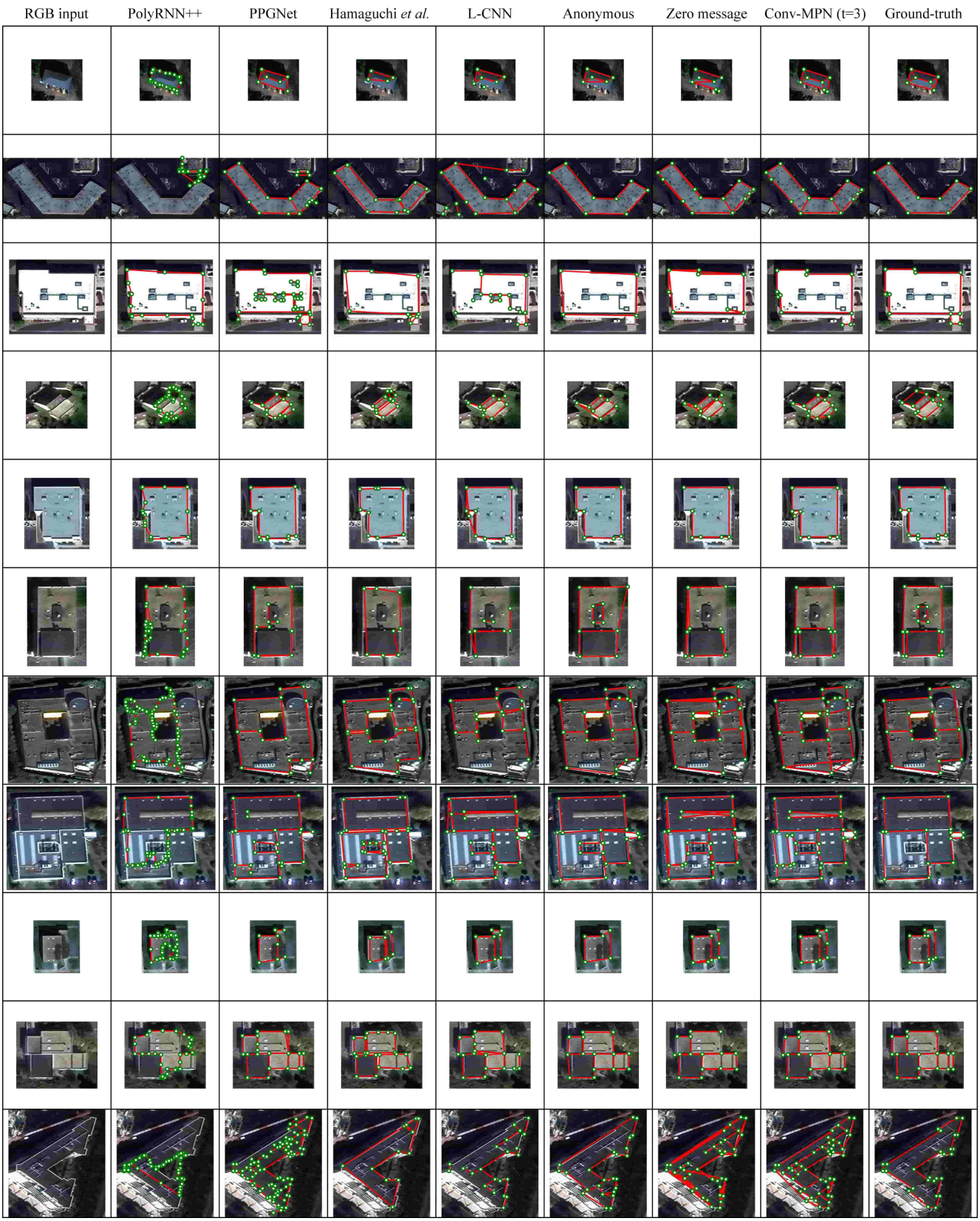

Figure 5. Additional qualitative results.

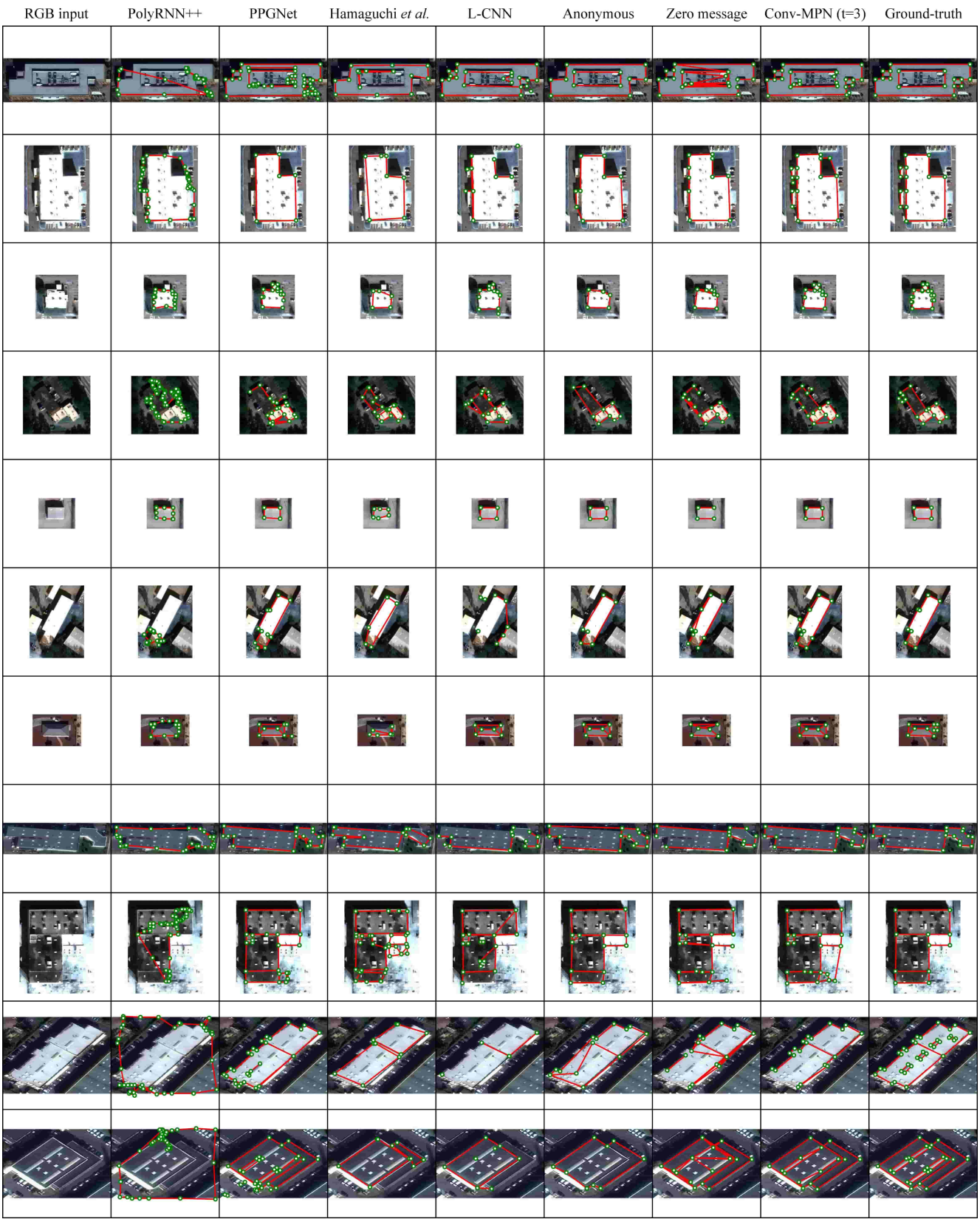

Figure 6. Additional qualitative results.

Figure 7. Additional qualitative results.

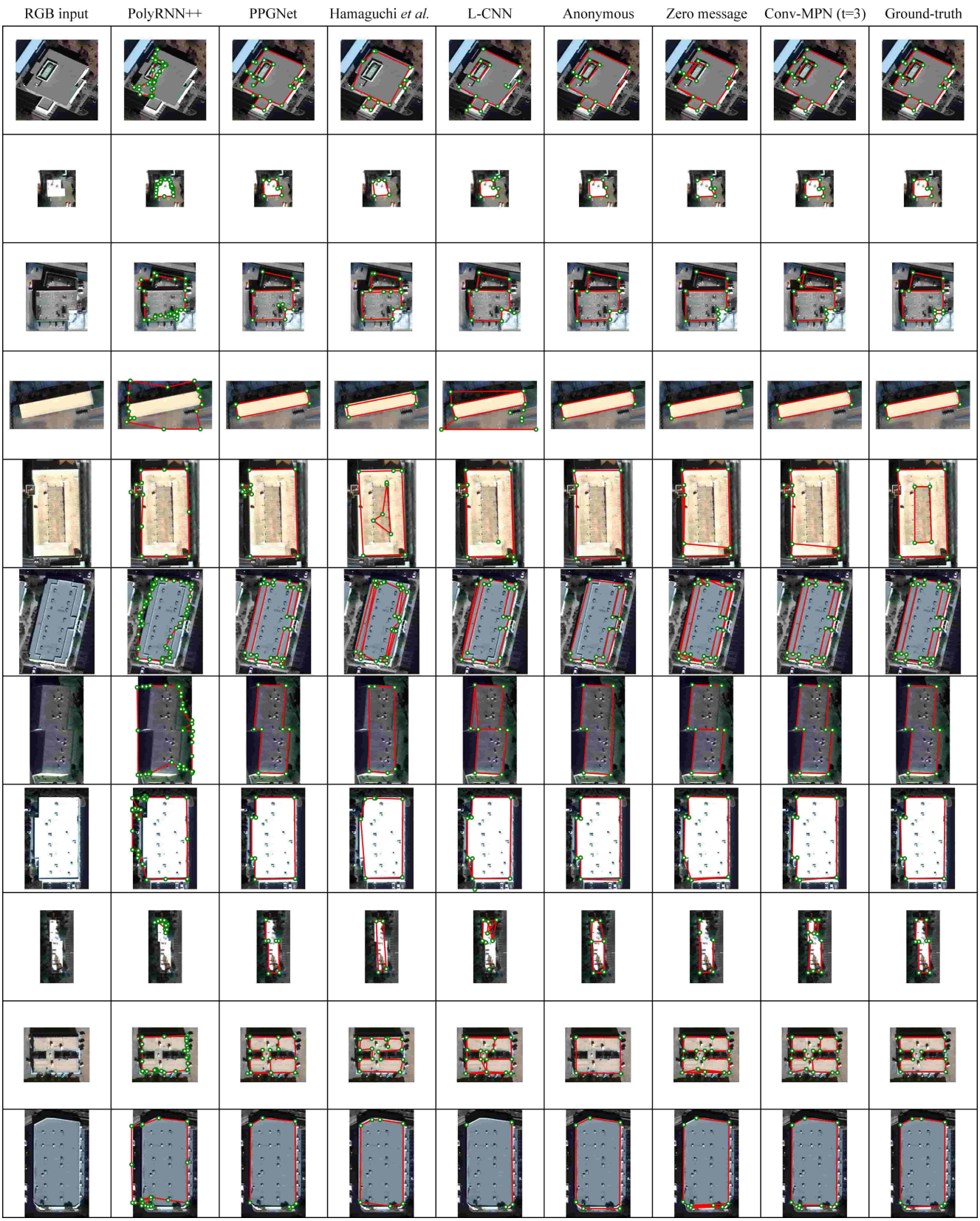

Figure 8. Additional qualitative results.

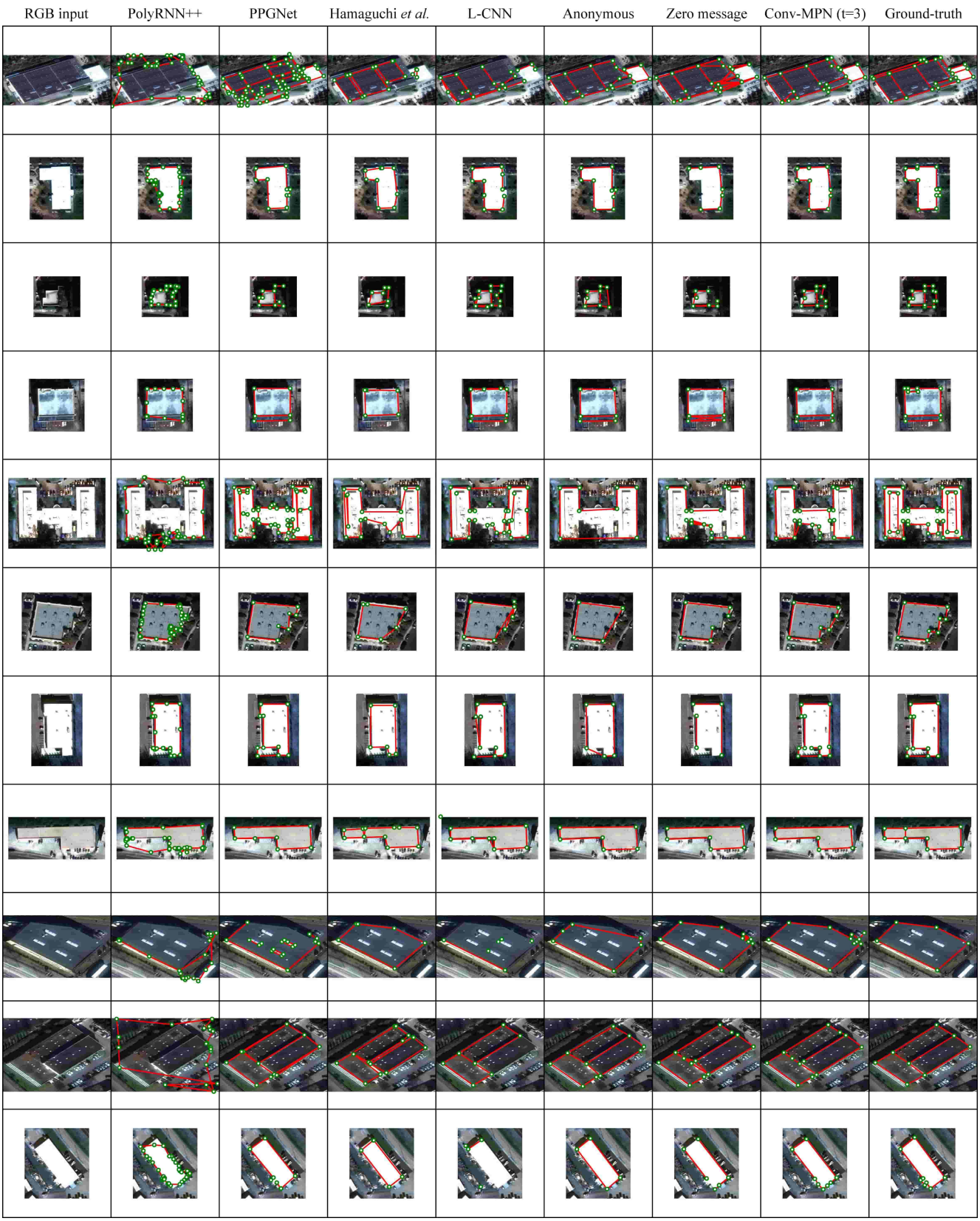

Figure 9. Additional qualitative results.

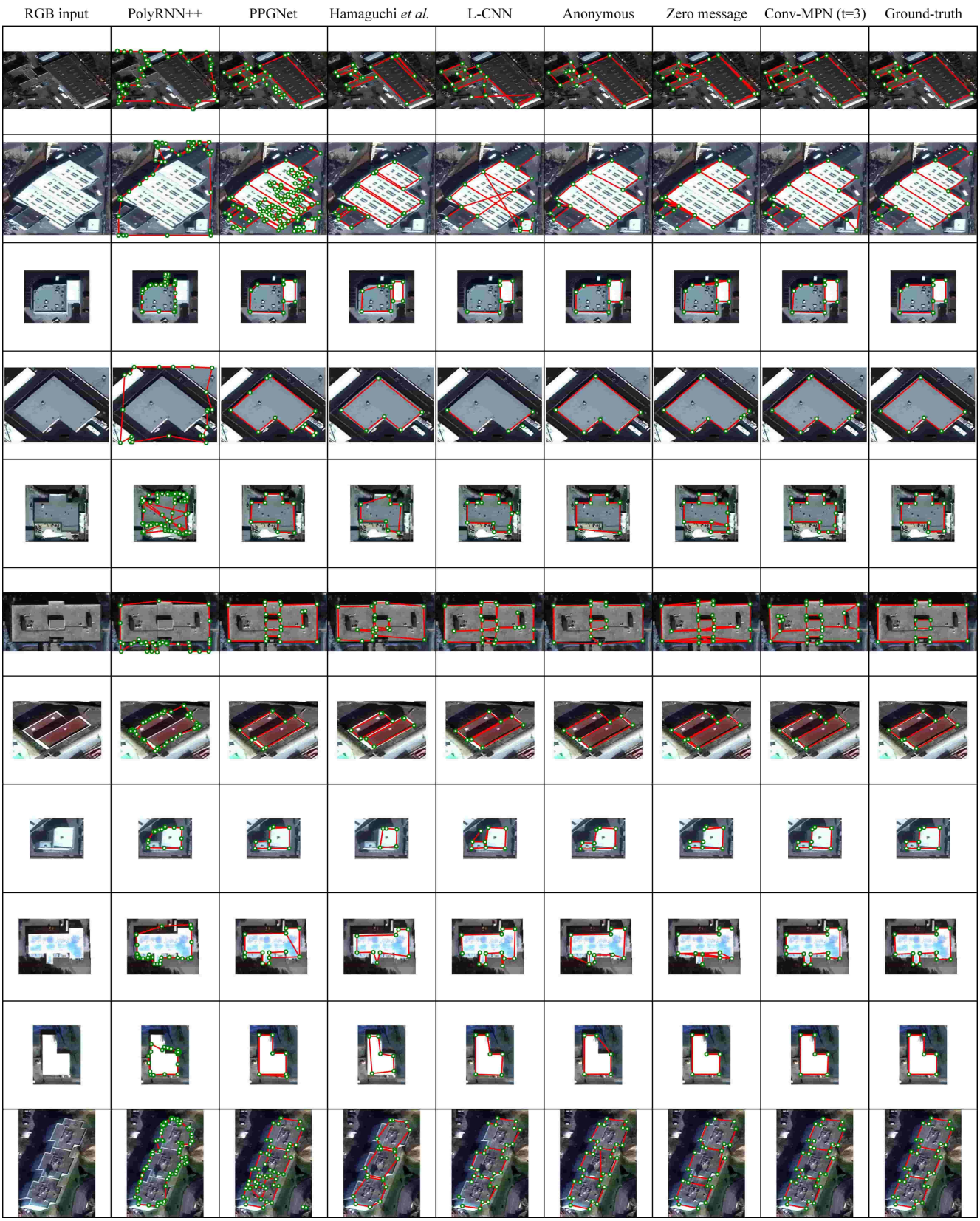

Figure 10. Additional qualitative results.

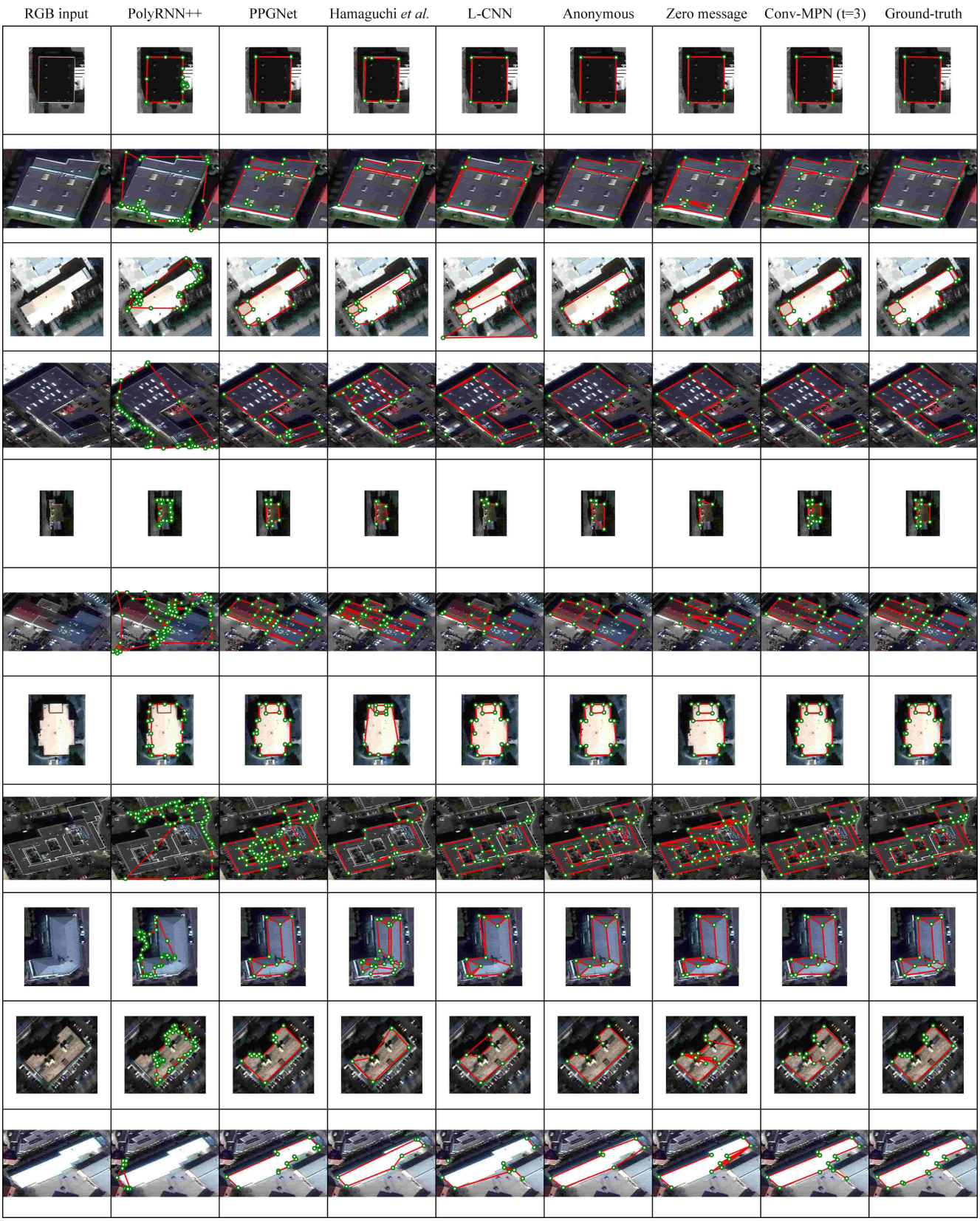

Figure 11. Additional qualitative results.

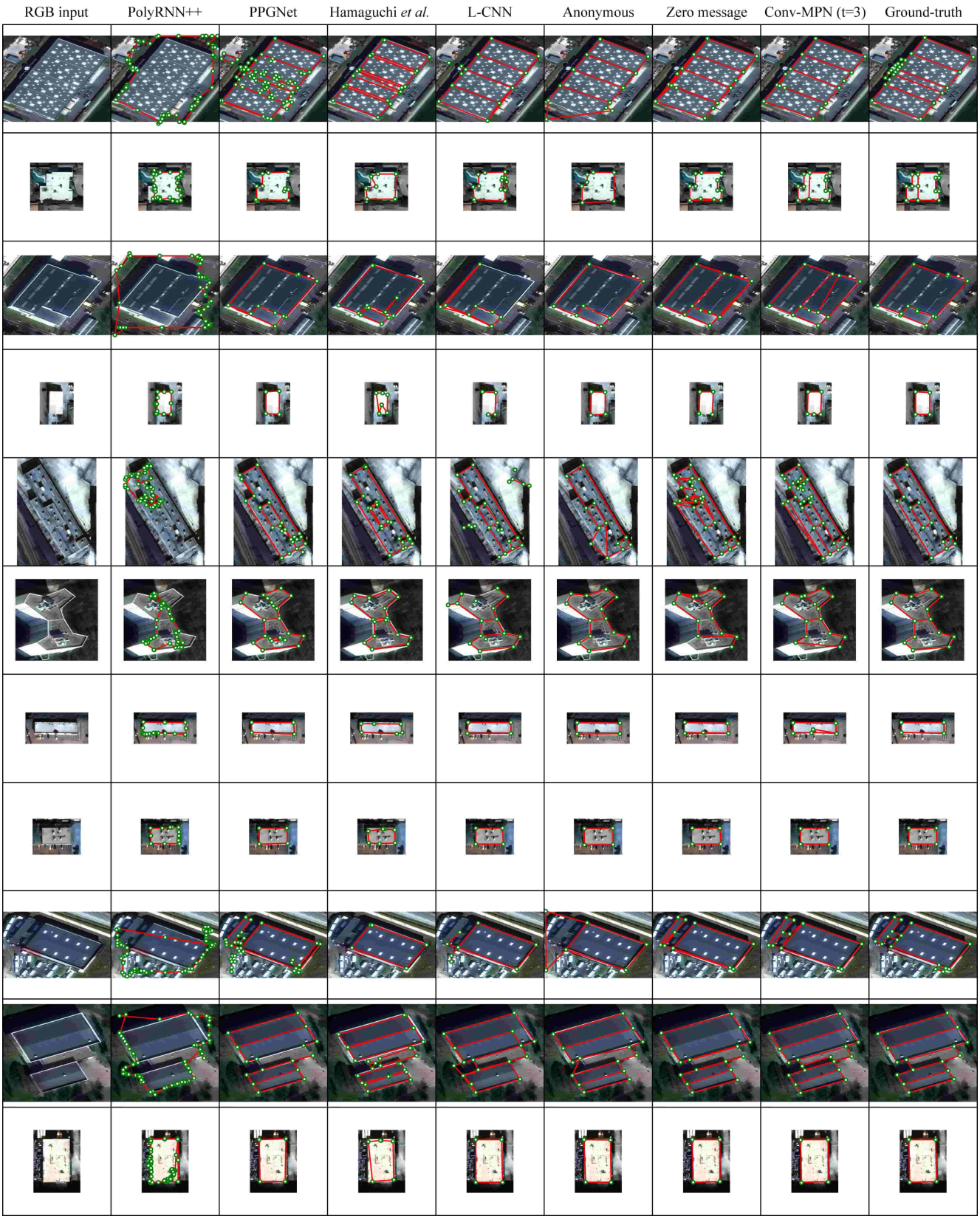

Figure 12. Additional qualitative results.

Figure 13. Additional qualitative results.

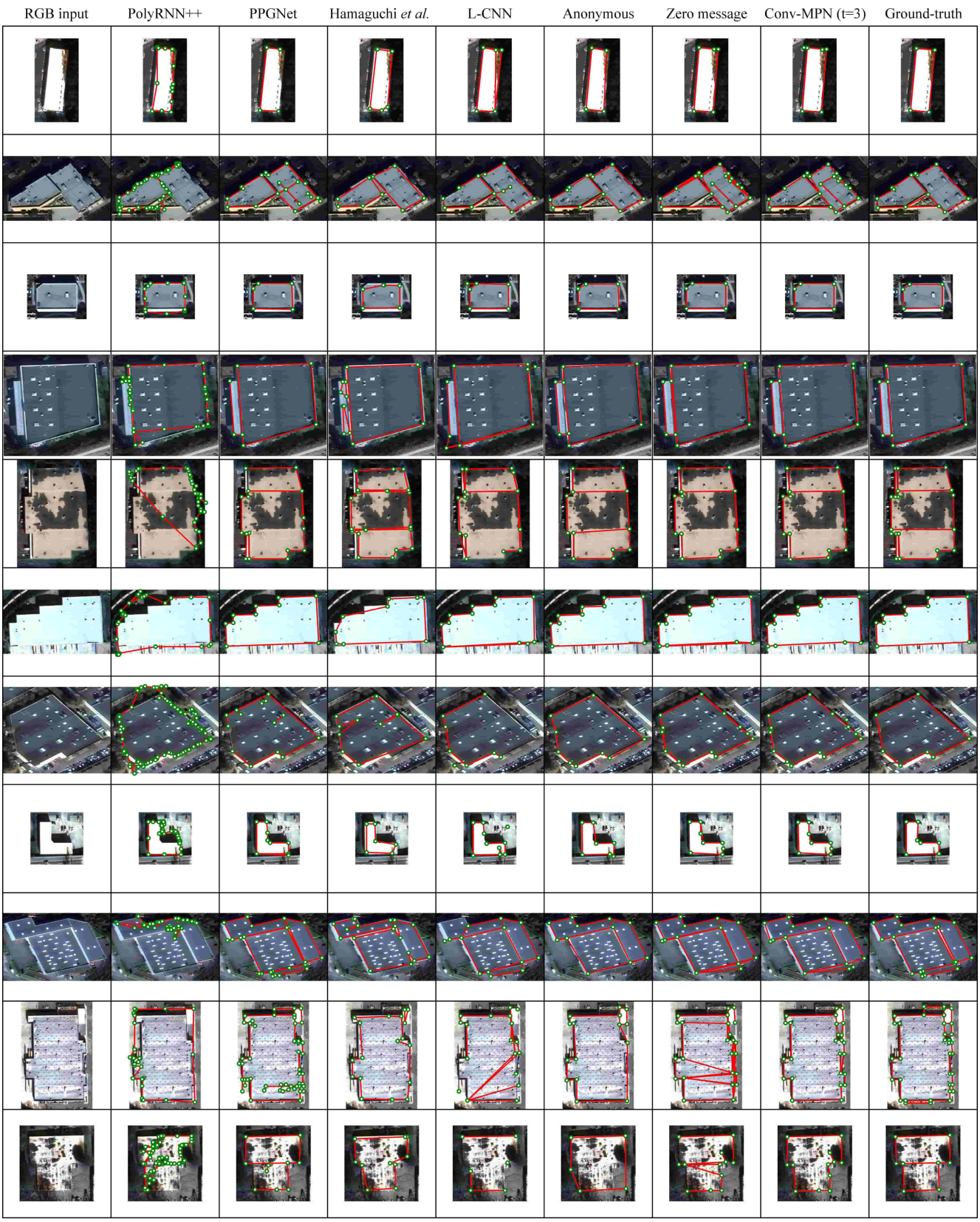

Figure 14. Additional qualitative results.

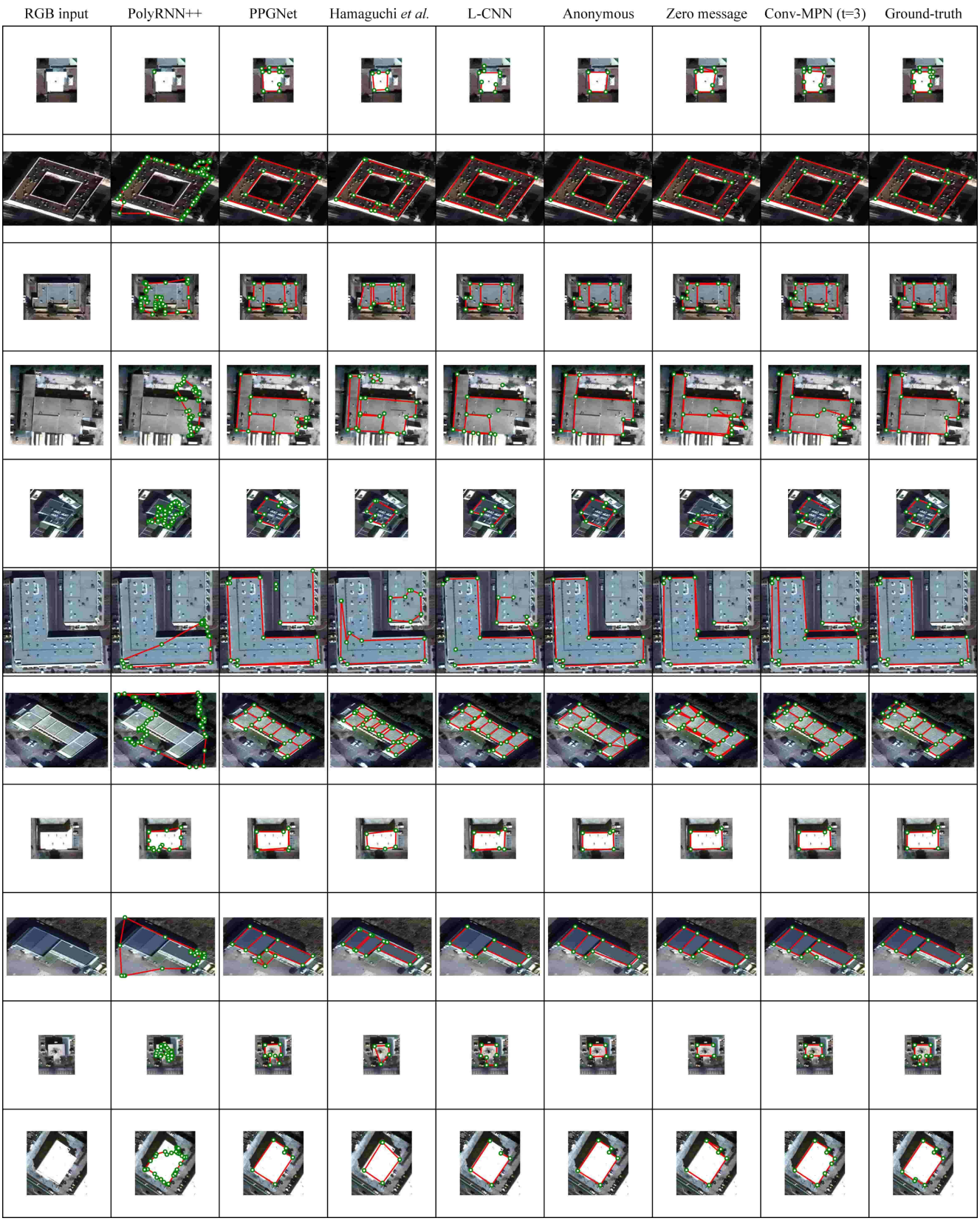

Figure 15. Additional qualitative results.

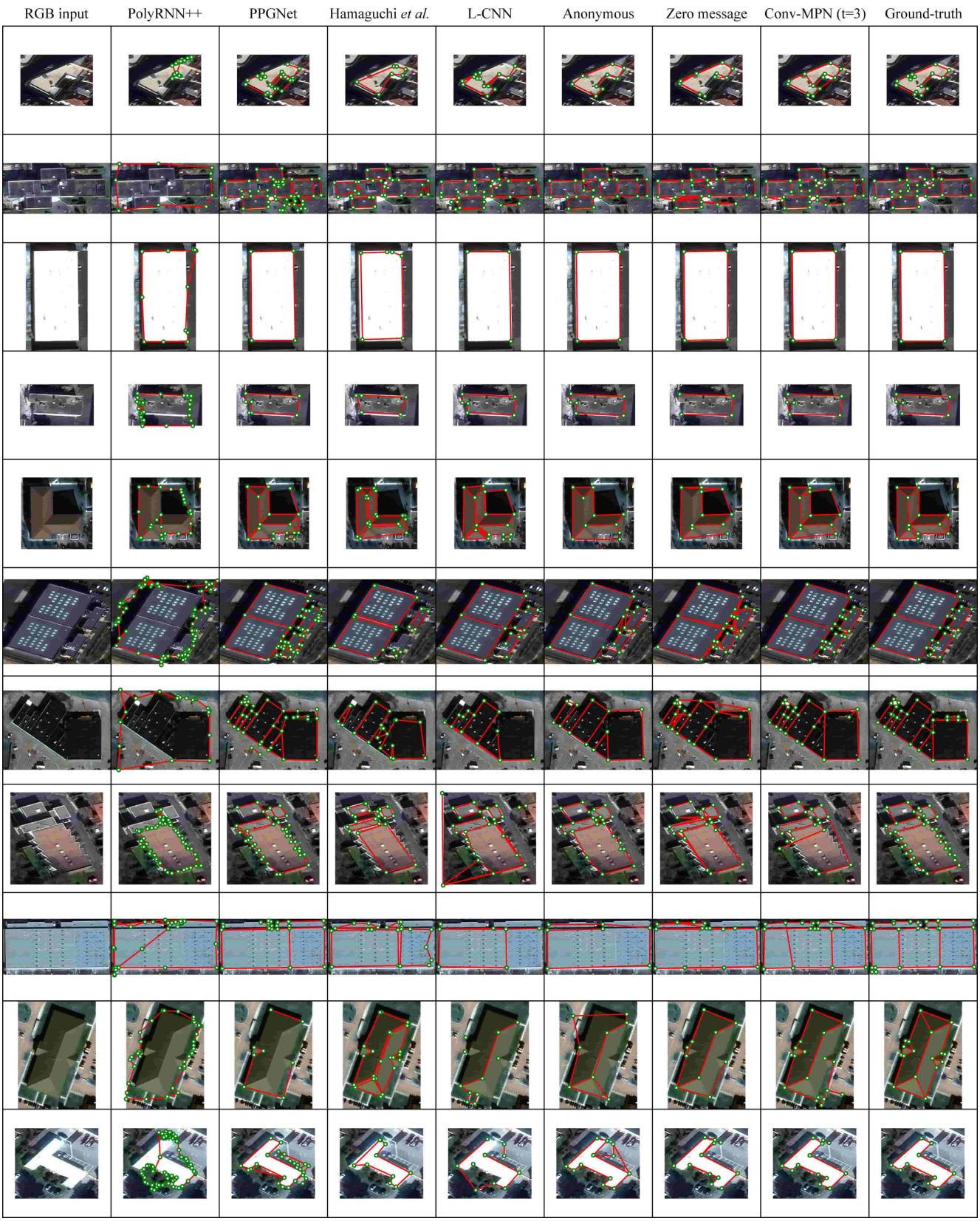

Figure 16. Additional qualitative results.

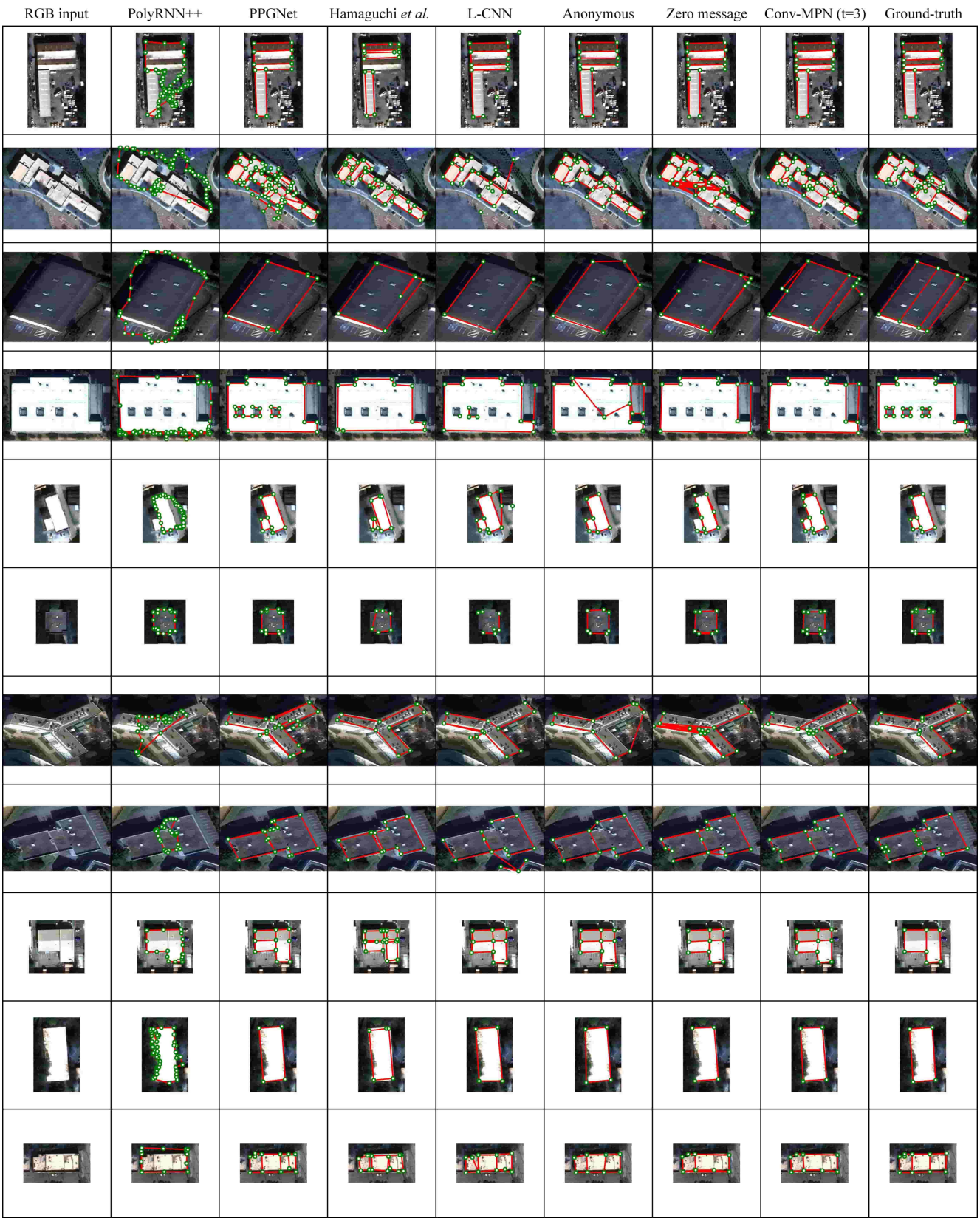

Figure 17. Additional qualitative results.

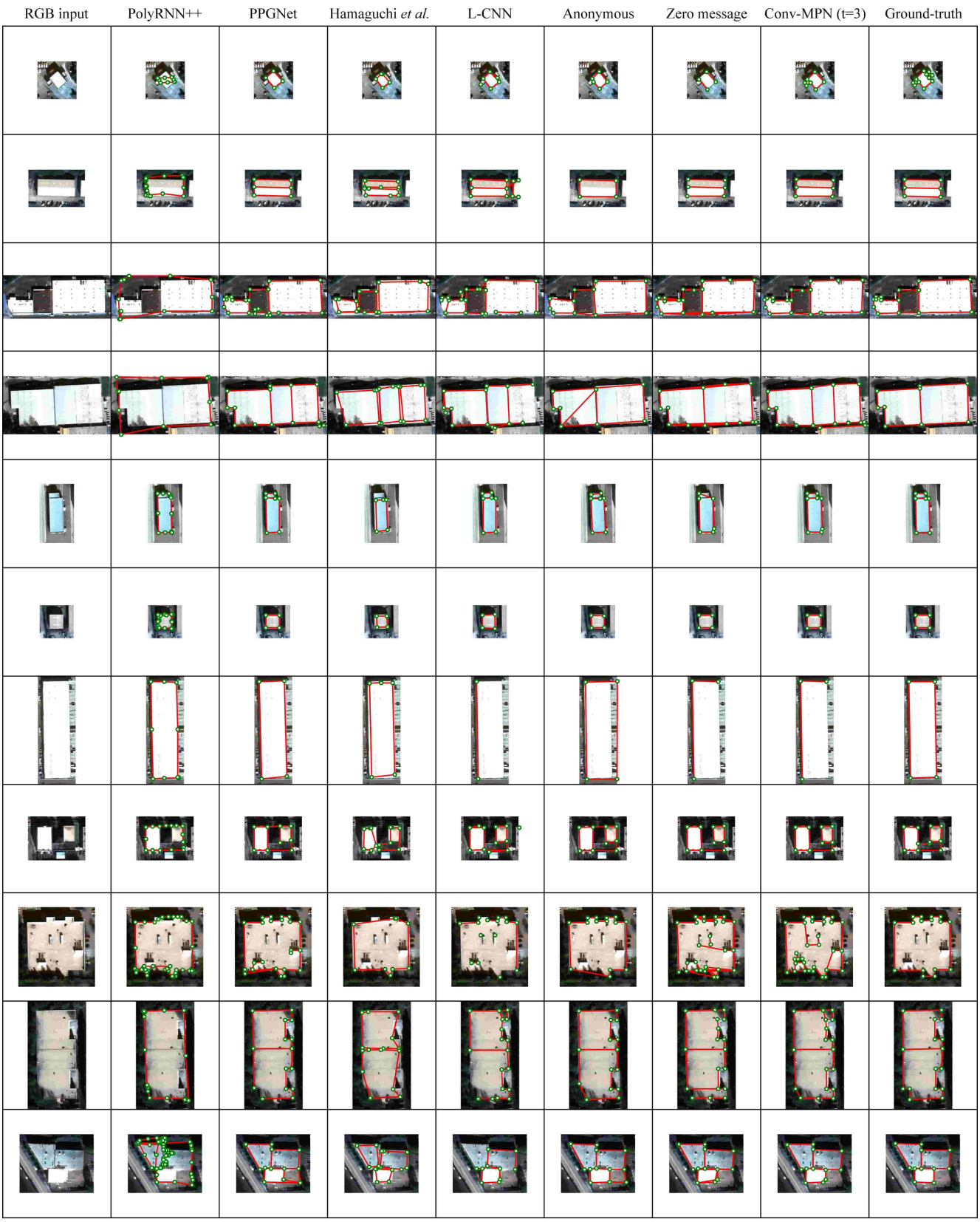

Figure 18. Additional qualitative results.

Figure 19. Additional qualitative results.

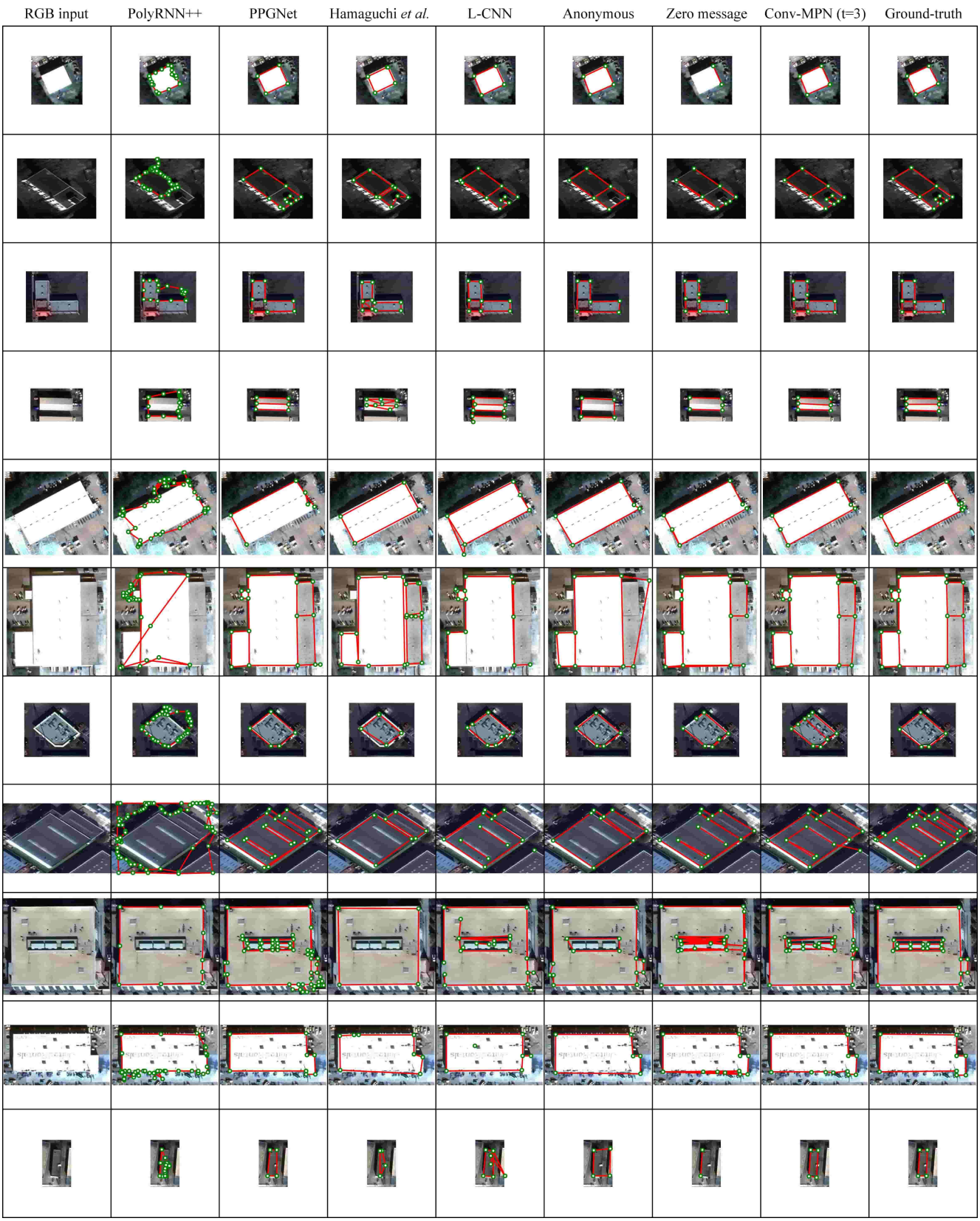

Figure 20. Additional qualitative results.

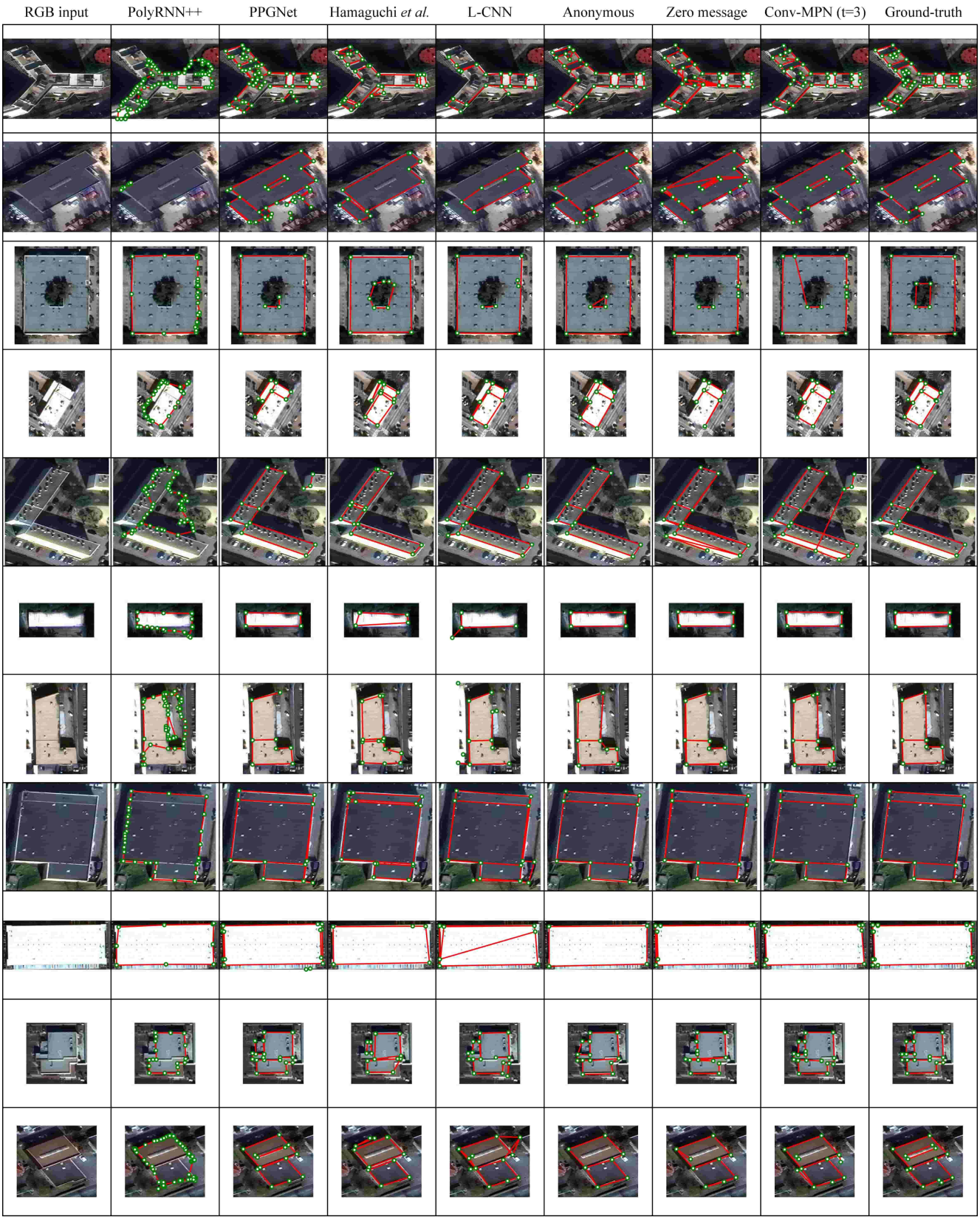

Figure 21. Additional qualitative results.

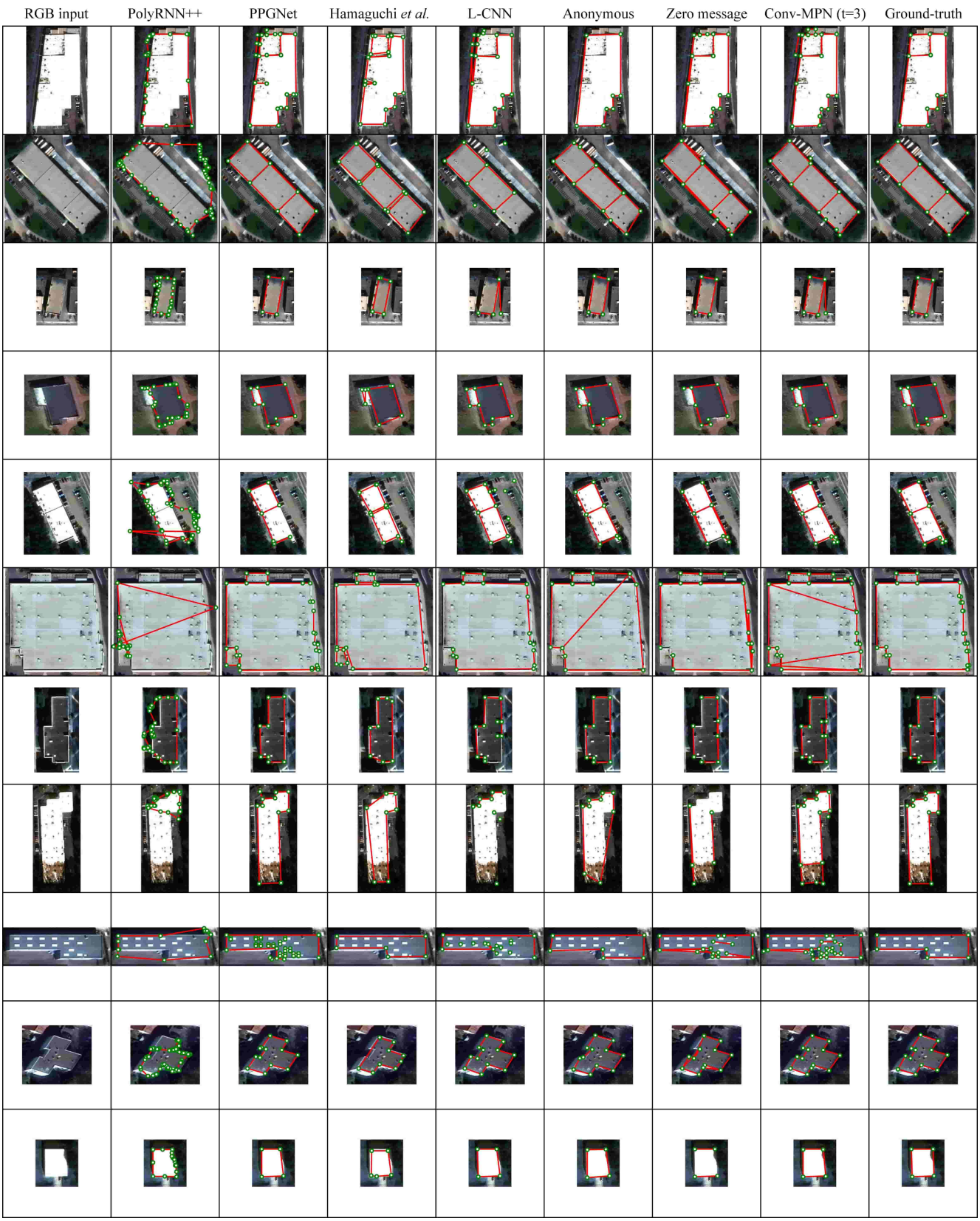

Figure 22. Additional qualitative results.

Figure 23. Additional qualitative results.

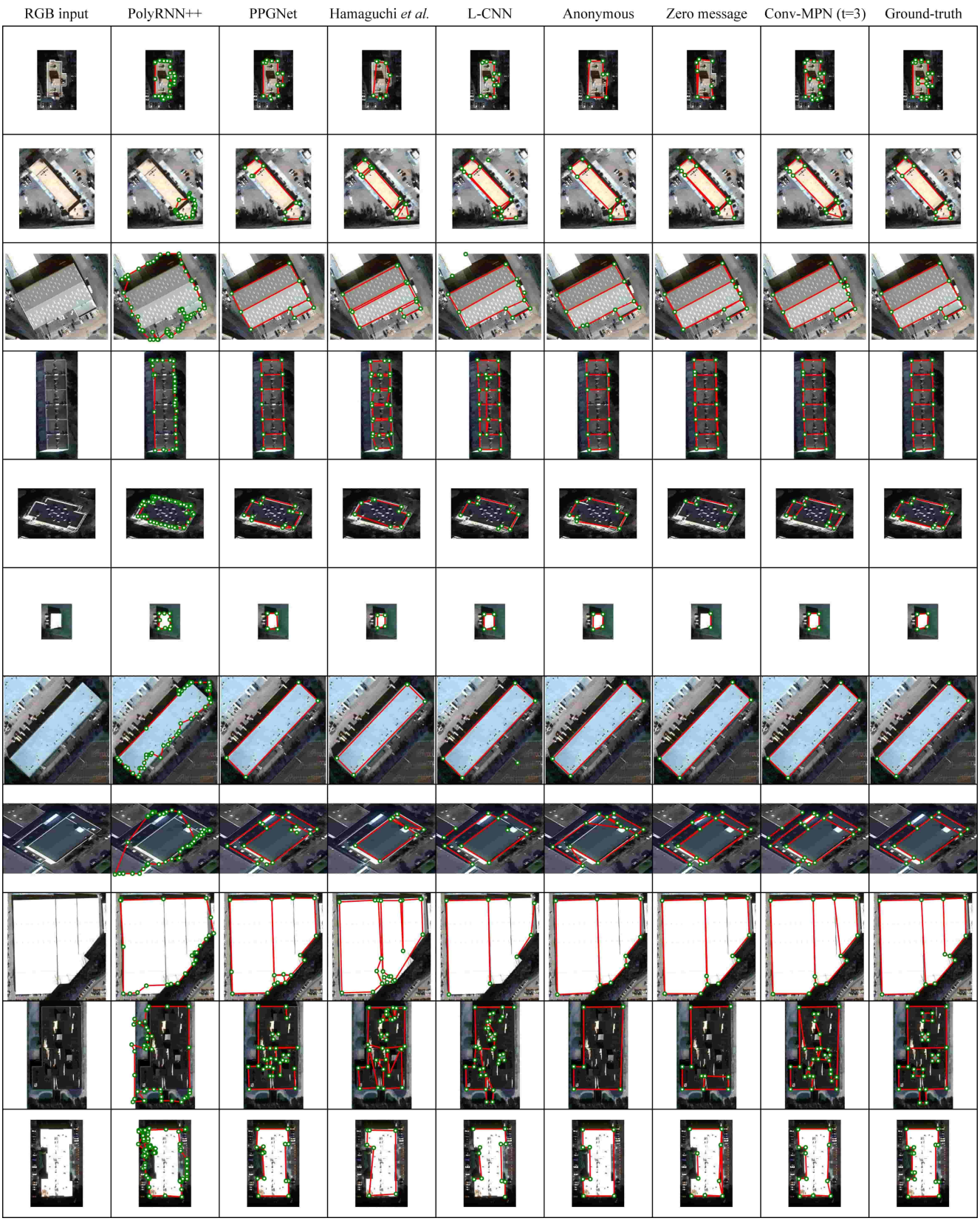

Figure 24. Additional qualitative results.

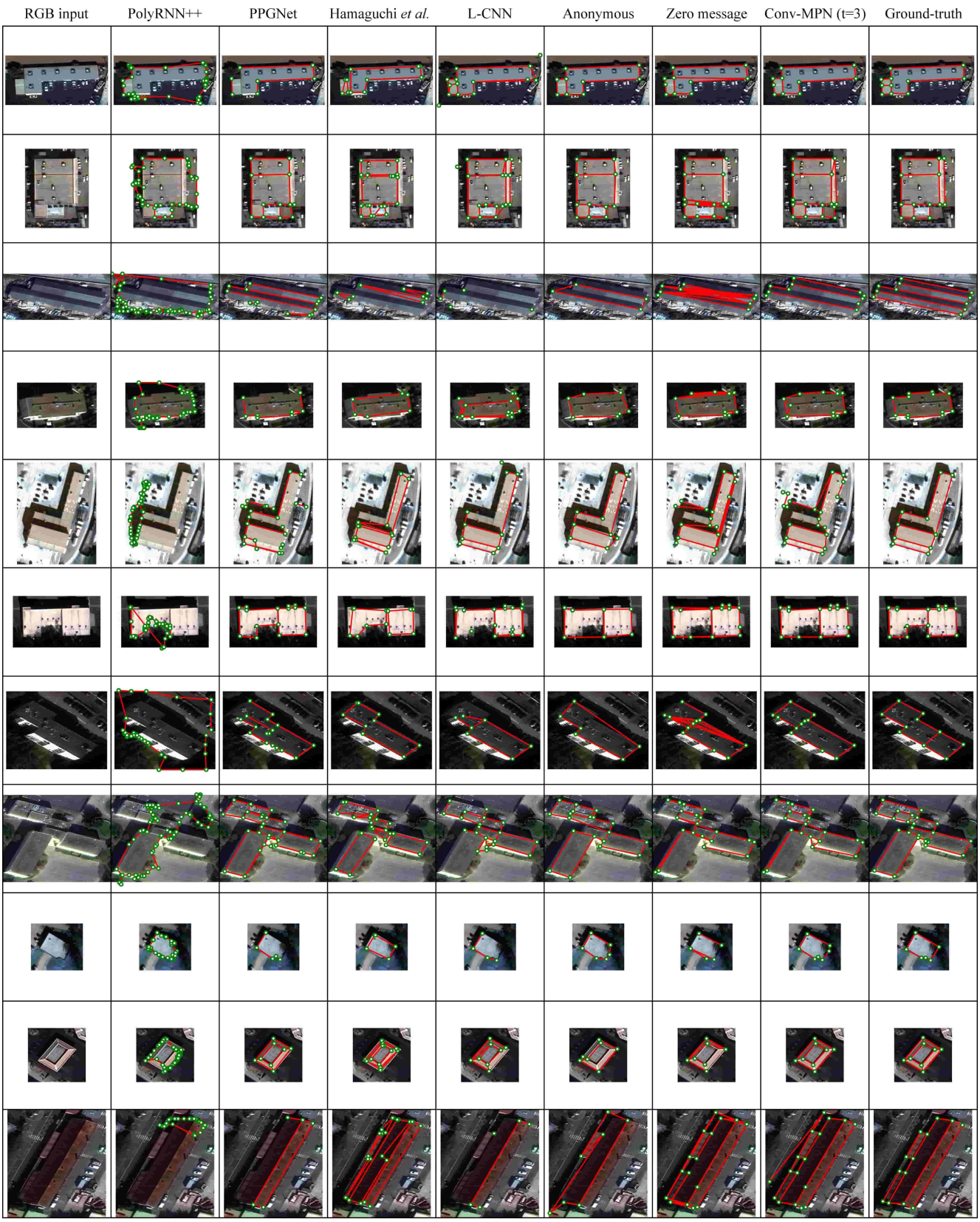

Figure 25. Additional qualitative results.

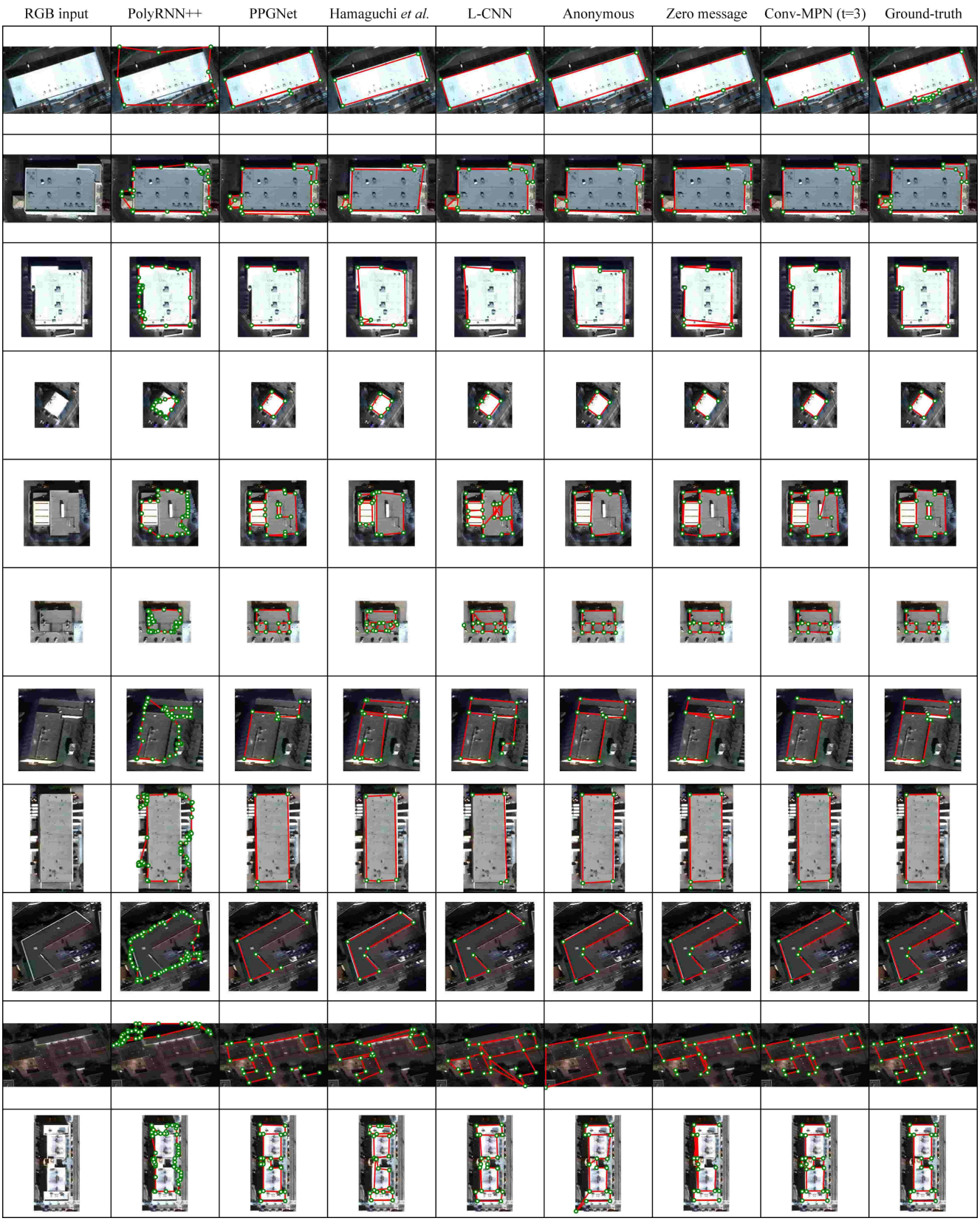

Figure 26. Additional qualitative results.

Figure 27. Additional qualitative results.

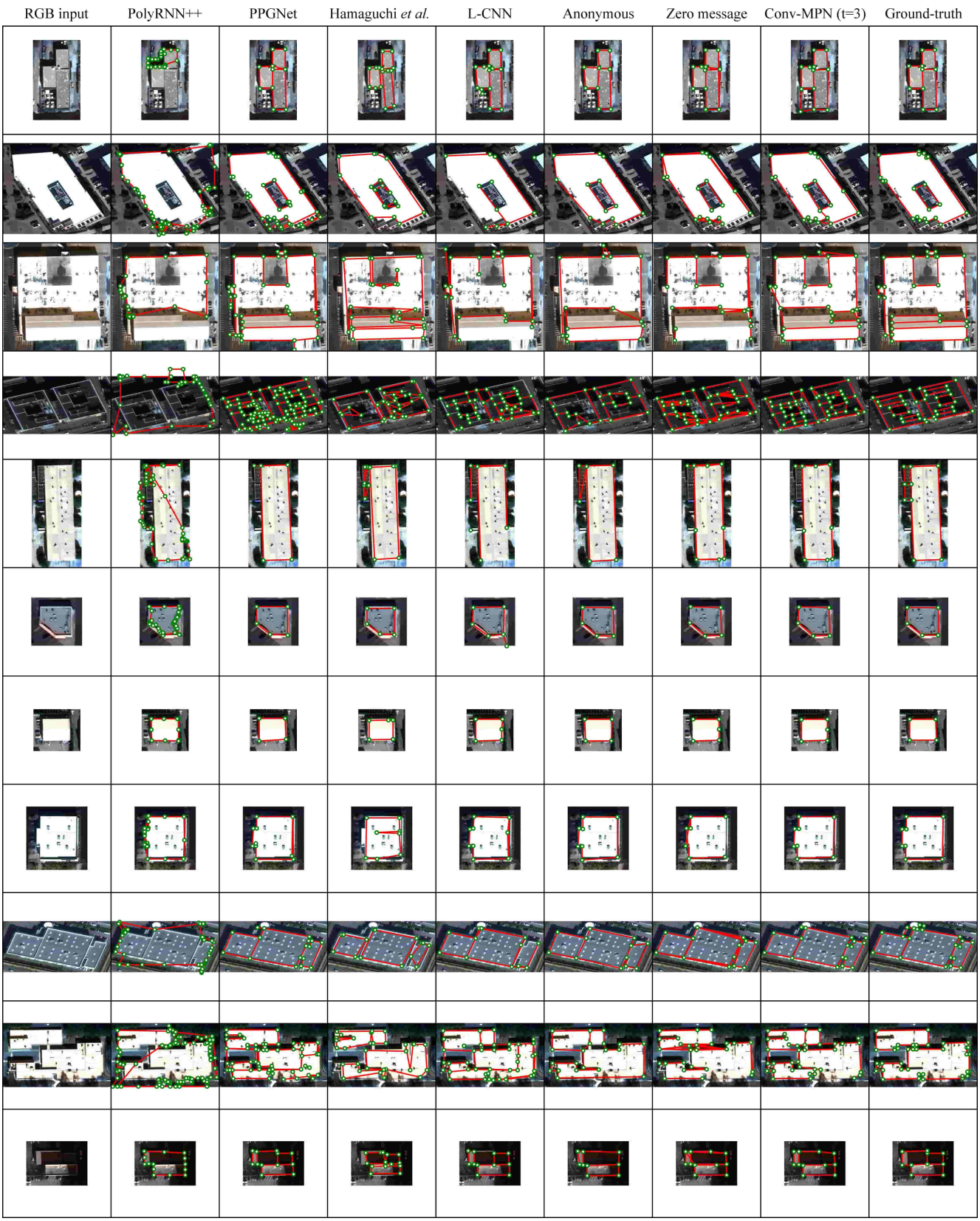

Figure 28. Additional qualitative results.

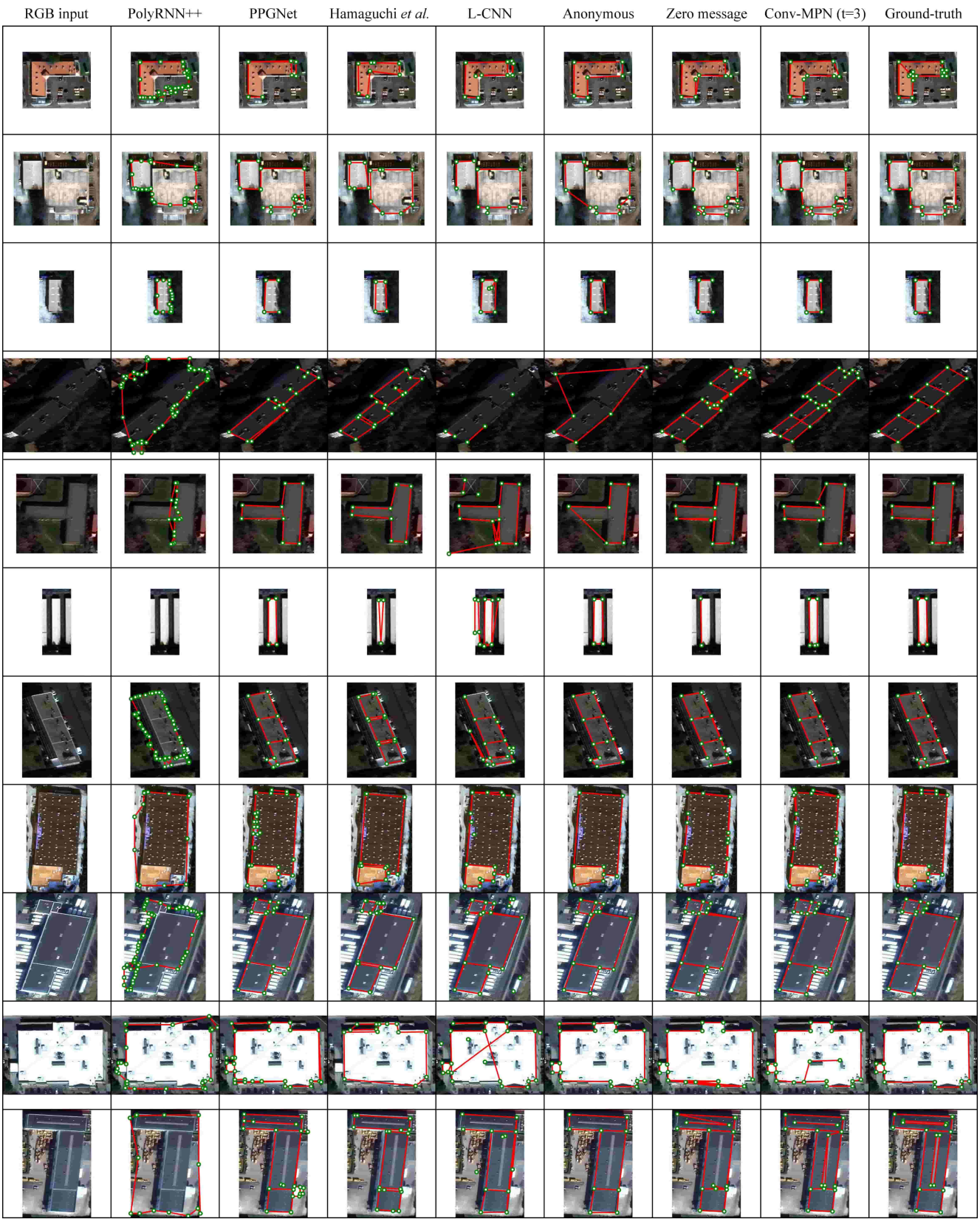

Figure 29. Additional qualitative results.

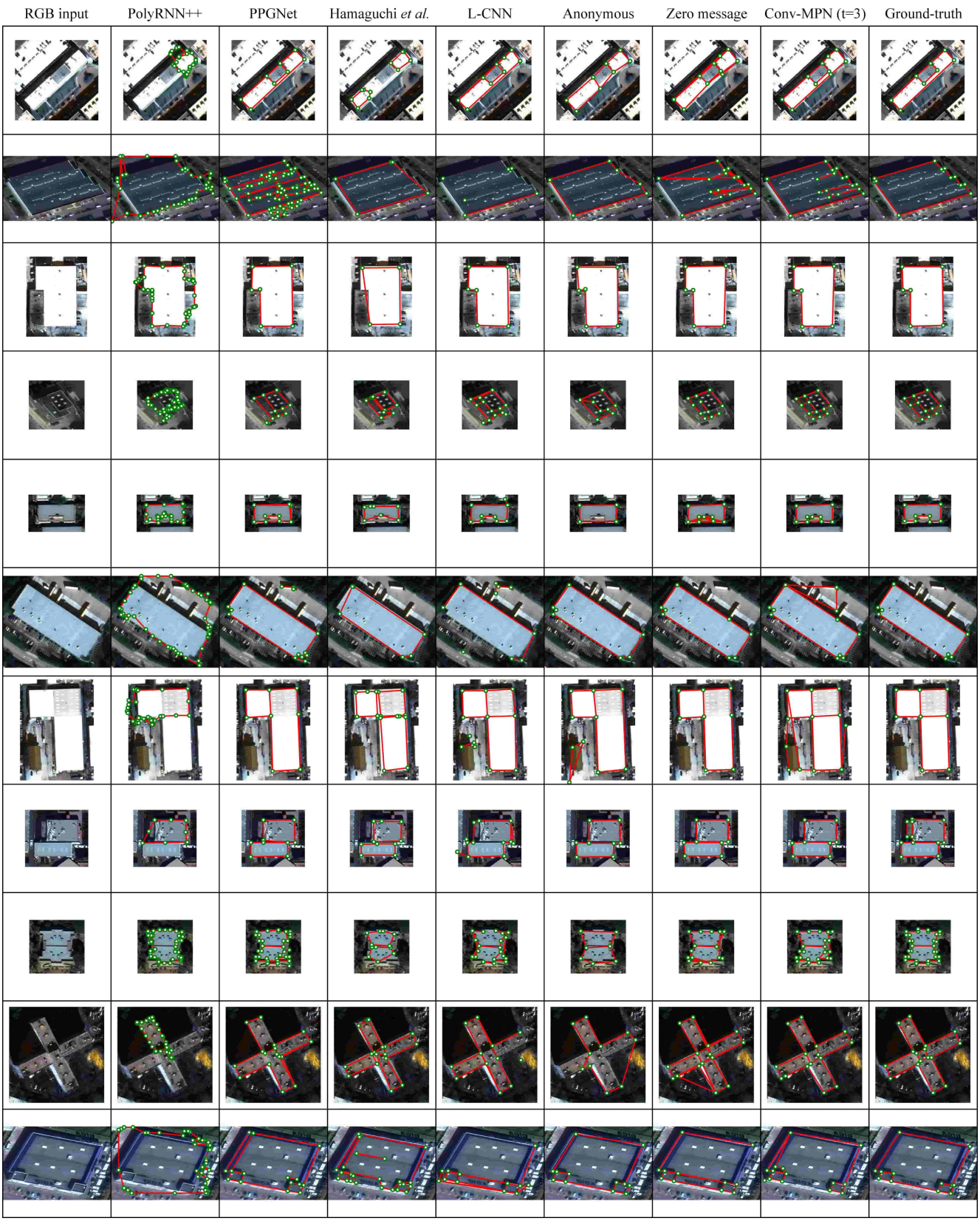

Figure 30. Additional qualitative results.

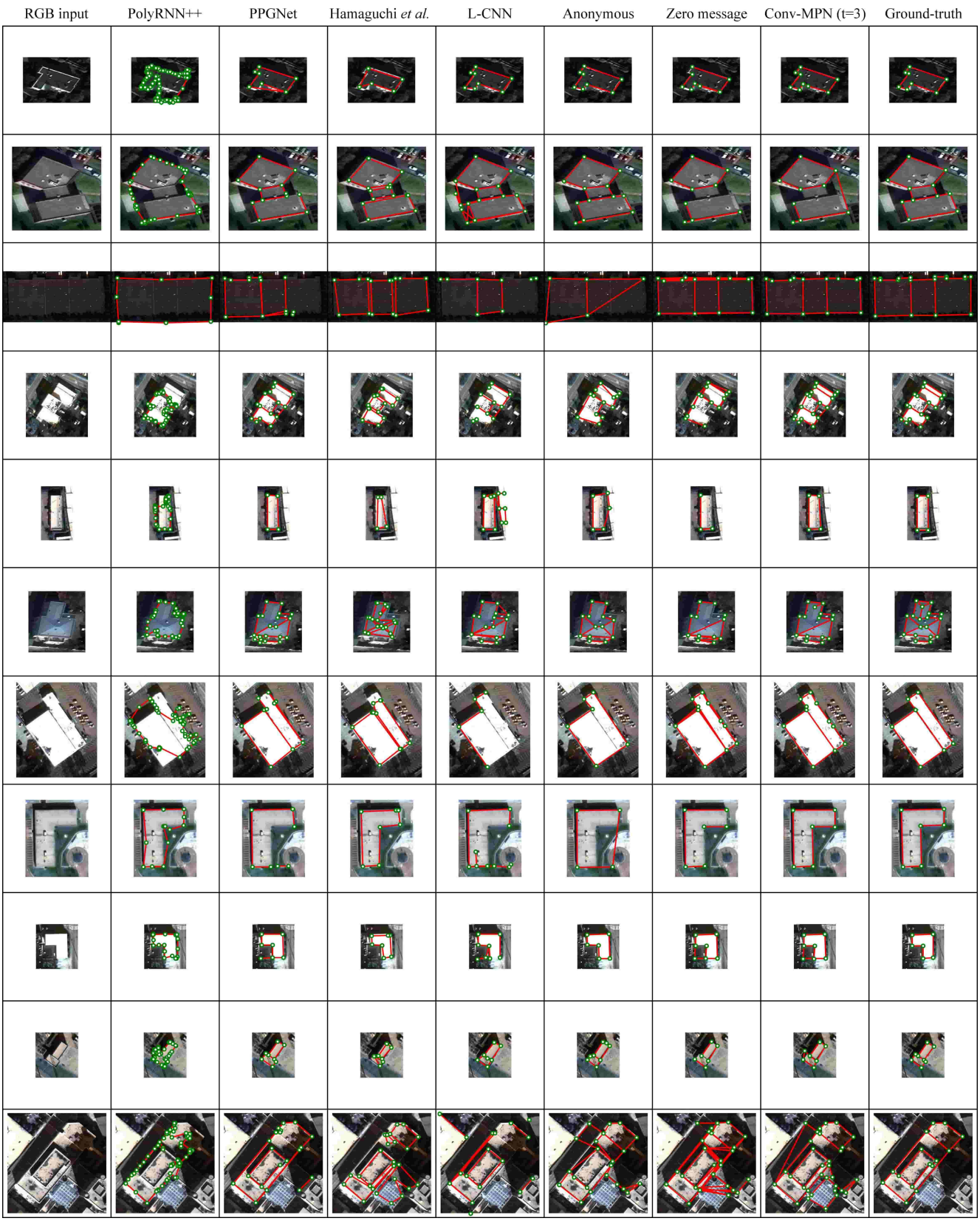

Figure 31. Additional qualitative results.

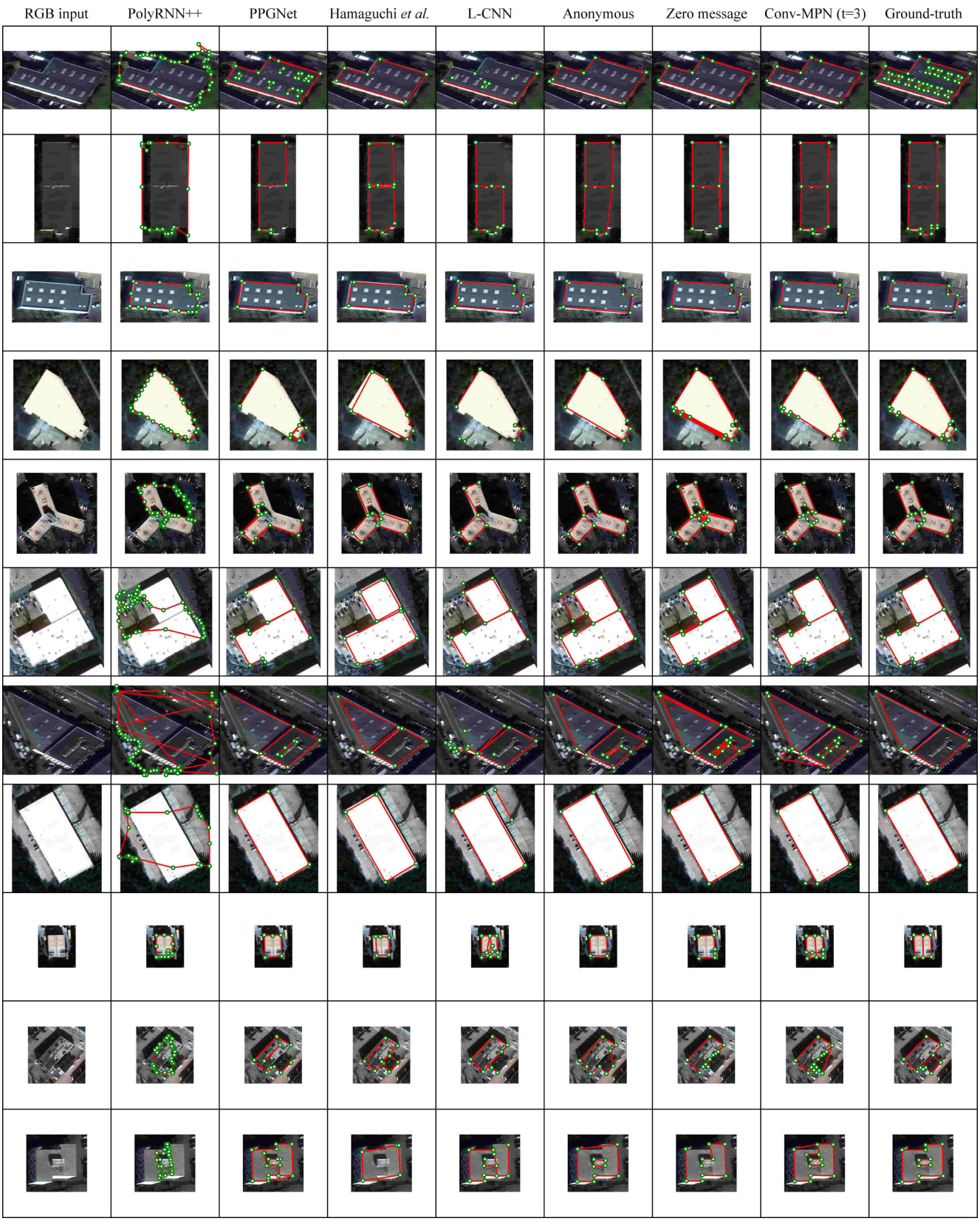

Figure 32. Additional qualitative results.

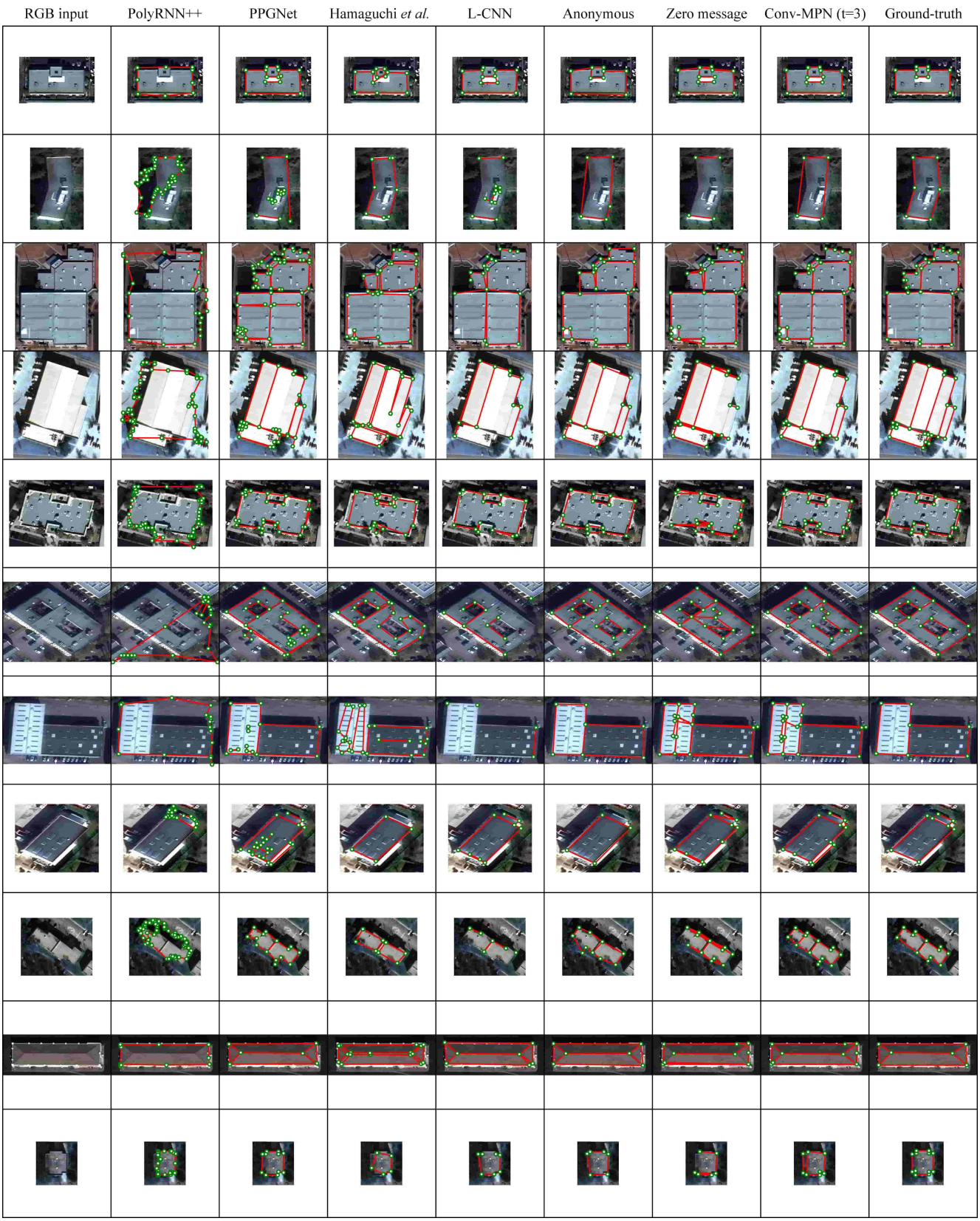

Figure 33. Additional qualitative results.

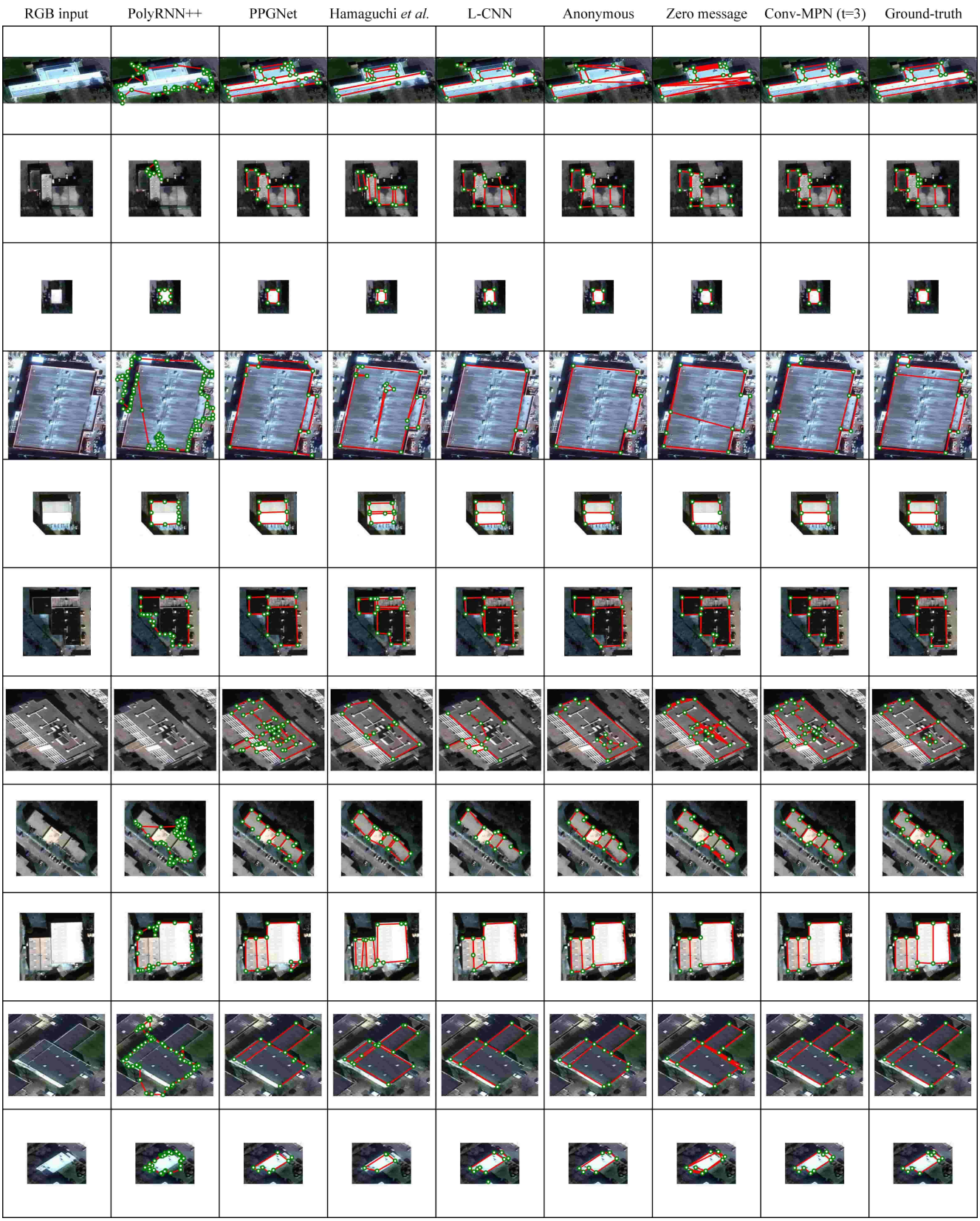

Figure 34. Additional qualitative results.

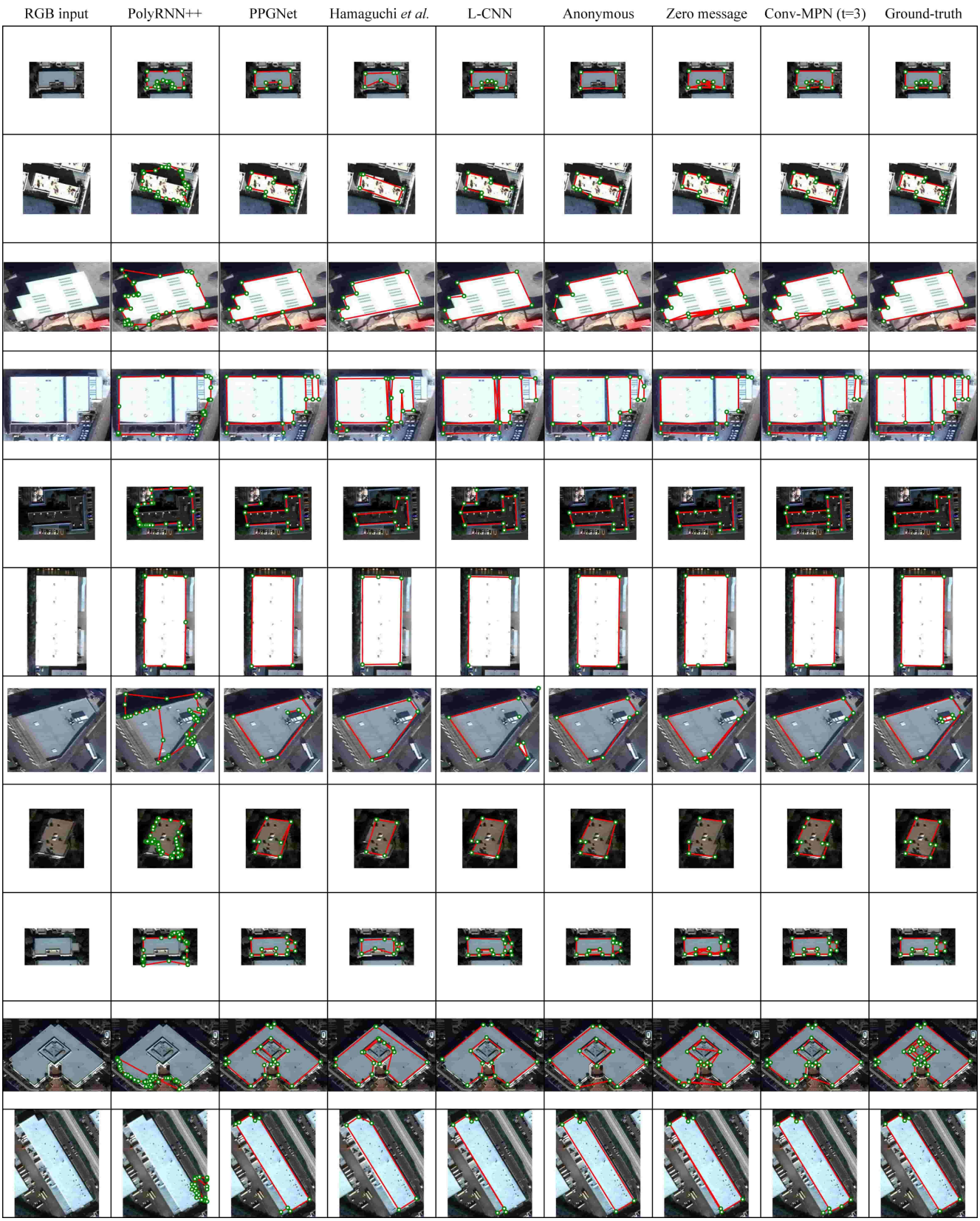

Figure 35. Additional qualitative results.

Figure 36. Additional qualitative results.

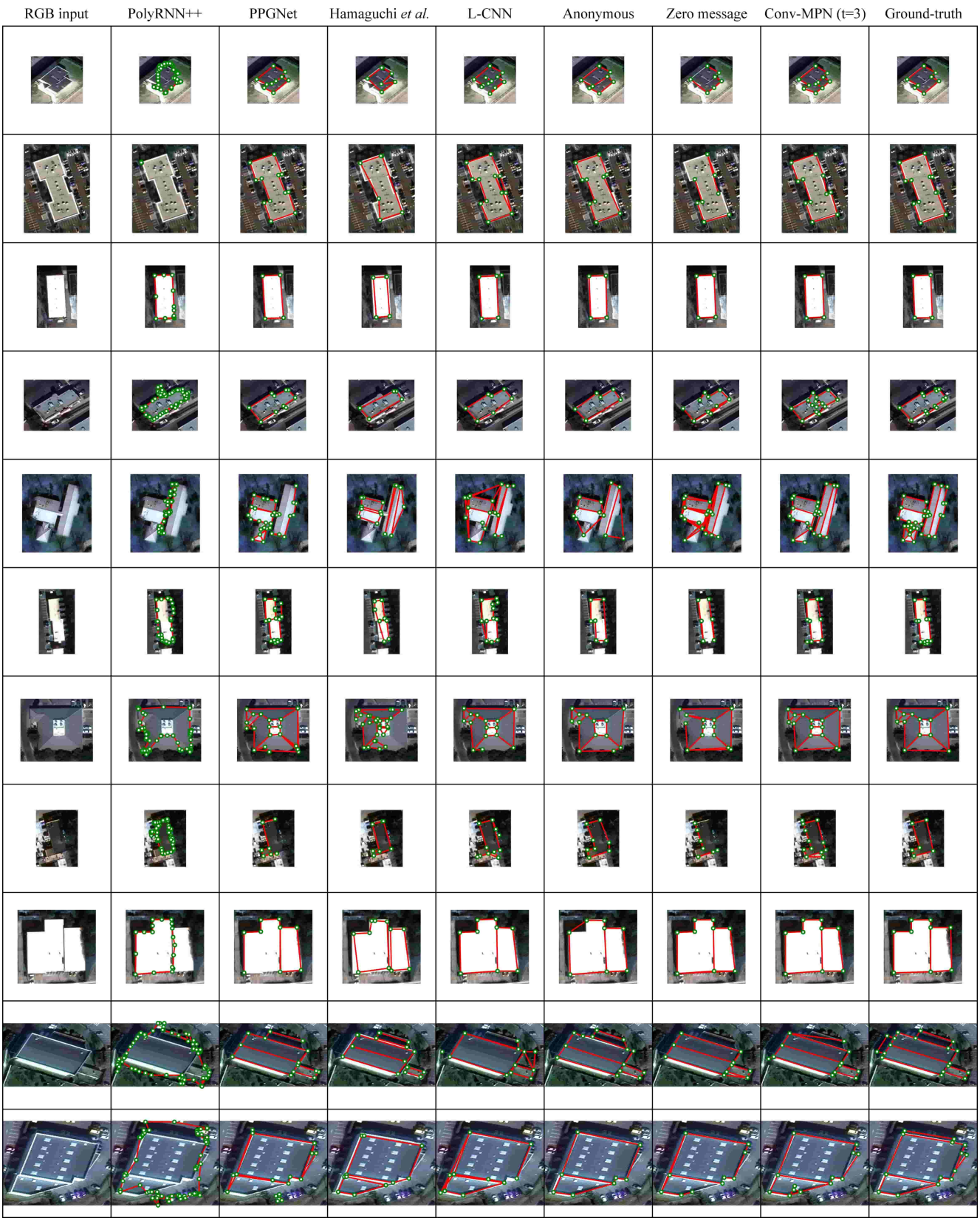

Figure 37. Additional qualitative results.