Conv-MPN: Convolutional Message Passing Neural Network for Structured Outdoor Architecture Reconstruction

Fuyang Zhang*, Nelson Nauata* and Yasutaka Furukawa
Simon Fraser University, BC, Canada
{fuyangz, nnauata, furukawa}@sfu.ca

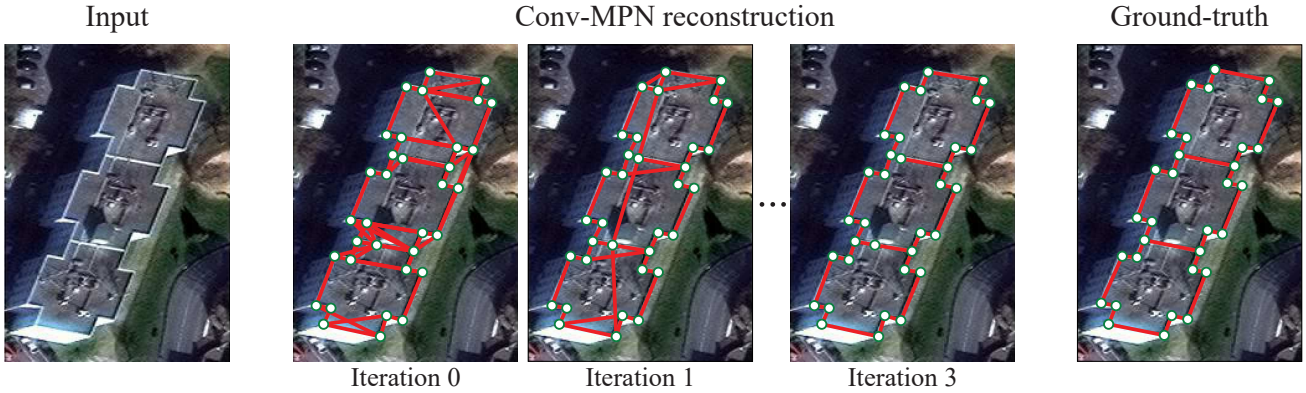

Figure 1. Conv-MPN, a novel message passing neural network, reconstructs outdoor buildings as planar graphs from a single image. The reconstructions after 0, 1, or 3 iterations of message passing are as shown.

Abstract

This paper proposes a novel message passing neural (MPN) architecture Conv-MPN, which reconstructs an outdoor building as a planar graph from a single RGB image. Conv-MPN is specifically designed for cases where nodes of a graph have explicit spatial embedding. In our problem, nodes correspond to building edges in an image. Conv-MPN is different from MPN in that 1) the feature associated with a node is represented as a feature volume instead of a 1D vector; and 2) convolutions encode messages instead of fully connected layers. Conv-MPN learns to select a true subset of nodes (i.e., building edges) to reconstruct a building planar graph. Our qualitative and quantitative evaluations over 2,000 buildings show that Conv-MPN makes significant improvements over the existing fully neural solutions. We believe that the paper has a potential to open a new line of graph neural network research for structured geometry reconstruction.

1. Introduction

Human vision evolved to master holistic image understanding, capable of detecting structural elements in an image and inferring their relationships. Look at a satellite image in Fig. 1. We can quickly see three building components, detect their building corners, and identify the common edges with the neighboring components.

The ultimate form of such structured geometry is the CAD representation, which enables a wide spectrum of applications such as rendering, effects mapping, simulation, or human interactions. Unfortunately, CAD model construction is still an open problem for computer vision, and is possible only by the hands of expert modelers.

Towards the automated construction of CAD geometry, the emergence of deep neural networks (DNNs) have brought revolutionary improvements to the detection of low-level primitives (e.g., corners). However, holistic understanding of high-level geometric structures (e.g., the inference of a graph) remains as a challenge for DNNs. The current state-of-the-art utilizes DNNs for low-level primitive detection, but employs optimization methods for high-level geometric structure inference [21, 16]. Optimization is powerful, but requires complex problem formulations and intensive engineering for injecting structural constraints.

This paper seeks to push the boundary of deep neural architecture for the task of structured geometry reconstruction. In particular, we propose a convolutional message passing neural network (Conv-MPN). Conv-MPN is a variant of a graph neural network (GNN), and learns to infer re-

*indicates equal contribution.

lationships of nodes by exchanging messages. Conv-MPN is specifically designed for cases where a node has an explicit spatial embedding, and makes two key distinctions from a standard message passing neural network (MPN): 1) the feature of a node is represented as a 3D volume as in CNNs instead of a 1D vector; and 2) convolutions encode messages instead of fully connected layers [24, 25] or matrix multiplications [5, 8]. This design allows Conv-MPN to exploit the spatial information associated with the nodes.

We have demonstrated the effectiveness of Conv-MPN on an outdoor architecture vectorization problem [21], where the input is a satellite RGB image and the output is a planar graph depicting both the internal and external architectural feature lines. This is a challenging problem for computer vision akin to the floorplan vectorization, which did not have an effective solution until recently [19].

The main challenge lies in the inference of a graph structure with an arbitrary topology. The outdoor architecture vectorization from a satellite image is even more challenging as Manhattan assumption does not hold due to the foreshortening effects. We would like to also emphasize the difference from the traditional building shape extraction problem [1], which represents a building as a set of pixels.

We qualitative and quantitative evaluate the proposed approach on more than 2,000 complex building examples in the cities of Atlanta, Las Vegas, and Paris [21]. Conv-MPN makes significant improvements over all the existing neural solutions. We believe that this research has a potential to open a new line of graph neural network research for structured geometry reconstruction. Code and pretrained models can be found at https://github.com/zhangfuyang/Conv-MPN.

2. Related work

We first review structured reconstruction techniques based on the levels of graph structure to be inferred, then the use of message passing techniques on structured data.

Reconstruction with a fixed topology: With the fixed known topology, graph reconstruction amounts to simply detecting keypoints and classifying their semantic types, because their connections are already given. Convolutional neural networks have shown to be effective in solving human pose estimation [22, 31, 29] and hand tracking [34, 38, 30].

Low- to mid-level structured reconstruction: DNNs detect corners and classify the presence of their connections for wire-frame parsing [15, 37, 36]. However, the connection classification is performed for each edge independently, lacking higher level geometric reasoning that considers a graph as a whole. In remote sensing, most building extraction methods represent a building as a set of pixels [14] or a 1D polygonal loop [2, 6, 20, 7], limiting the output to a building external boundary as a 1D loop. In contrast, we seek to infer a graph of an arbitrary topology encoding both internal and external architectural feature lines.

Structured reconstruction (optimization): The state-of-the-art graph structure inference combines CNNs and optimization, in particular, integer programming (IP) [19, 18, 21]. CNNs detect low-level geometric primitives (e.g., corners) or infer pixel-wise graph information (e.g., edge likelihood). IP fuses all the information and infer graph structure, which is powerful but requires complex problem formulations and intensive engineering for injecting structural constraints.

Structured reconstruction (learning): A few methods learn to infer high-level geometric structure. Ritchie *et al*. [27] used DNNs to learn the arrangement of 2D strokes on a canvas with a simple shape-grammar similar to the L-system. Frans *et al*. [12]. proposed an unsupervised approach to the problem. However, their grammar is too rudimentary to represent building architecture. Zeng *et al*. [35] utilizes an architectural shape-grammar to reconstruct outdoor buildings from an ortho-rectified depthmap. However, their shape-grammar is again too restrictive: 1) Requiring ortho-rectification to utilize the Manhattan assumption; and 2) Modeling only small residential houses. This paper does not rely on a shape grammar, instead learns structural regularities from examples and utilize in the structure inference.

Message passing and convolution on graphs: Message passing has been an effective tool for high-level data reasoning [23, 3, 13, 17, 9, 32, 4]. A standard way is to extend the convolution operation over a grid of pixels to a graph of vertices [5, 17, 9, 10, 23]. Bruna *et al*. [5], Defferand *et al*. [9] and Kipf *et al*. [17] utilizes spectral analysis to define graph convolutions that act on an entire graph. The key difference is that our convolutions do not occur in the graph domain. Conv-MPN takes a graph with an explicit spatial embedding, represent a node as a feature volume and perform convolutions in the spatial domain. This framework allows Conv-MPN to exploit the spatial information associated with the nodes of a graph.

3. Preliminaries

This paper tackles the 2D architecture vectorization problem from a single satellite image, where a building is represented as a 2D planar graph. This section describes our data source and pre-processing steps (See Fig. 2).

Dataset: Our data source is a set of high-resolution satellite RGB images from SpaceNet [11] corpus, hosted through the Amazon Web Services (AWS) as a part of the SpaceNet Challenge [1]. In particular, we use an existing benchmark [21], which cropped 2,001 buildings into 256×256 square image patches for the cities of Atlanta, Paris and Las Vegas [21]. We use the same training and testing split

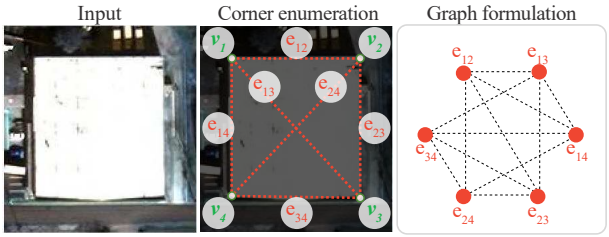

Figure 2. Preliminaries. Given a RGB image, we detect building corner candidates, enumerate bulding edge candidates, then formulate a graph for inference whose nodes are building edges.

(1601/400) as well as the metrics, which consists of the precision, recall, and f1-score for each of the corner, edge, and region primitives. Note that the satellite images are off-nadir, and buildings do not follow the Manhattan assumption due to the foreshortening effects.

Corner candidates enumeration: Given an input RGB image-patch $\mathcal{I}$ (256×256), we use Faster-RCNN with ResNet-50 as the backbone [26] to detect corner candidates, while treating each corner as a 8×8 bounding box with a corner at the center. The model is trained using SGD with the leaning rate set to 0.0001 and the batch-size of 1.

Graph formulation: Given building corner candidates, we enumerate building edge candidates by every pair of corners. Each edge candidate becomes a node in the graph for Conv-MPN inference in the next step (See Fig. 2 right). Nodes are connected when the corresponding building edges share the same building corner. Notice that we have a building planar graph and the graph for Conv-MPN inference. We explicitly write "building corner" or "building edge" when terms are confusing.

4. Conv-MPN architecture

The fundamental idea behind Conv-MPN is simple yet powerful. Standard graph neural networks (GNNs) [4] encode geometry information as a 1D vector numerically instead of a 3D feature volume spatially. MLP with 1D feature vectors cannot make effective geometric analysis, while convolution with 3D feature volumes enables natural spatial reasoning. Our idea is to take the standard MPN architecture then replace 1) a latent vector with a latent 3D volume for the feature representation; and 2) fully connected layers (or matrix multiplications) with convolutions for the message encoding. The section explains the Conv-MPN architecture specific to our problem setting, but it is straightforward to extend the framework to the entire GNN family.

4.1. Feature initialization

A node in the inference graph corresponds to a building-edge, which is to be represented as a 3D feature volume. We initialize the feature volume by passing a building RGB image concatenated with a binary building edge mask ($256 \times 256 \times 4$) through Dilated Residual Network (DRN) [33] (See Fig. 3). More specifically, we use the first three blocks of the DRN-C-26 architecture, followed by one 3×3 stride 2 convolution for downsampling to $64 \times 64 \times 32$. During training, we initialize the network parameters by the pretrained weights on the ImageNet [28]. [1]

4.2. Convolutional message passing

A standard form of feature vector update in MPN is to utilize multi layer perceptron (MLP) for encoding messages and mixing with the current feature:

$$f_v \leftarrow \text{MLP}\left(f_v; \sum_{w \in \mathbf{N}(v)} \text{MLP}(f_v; f_w)\right) \tag{1}$$

f_v denotes the feature vector associated with a node v, $\mathbf{N}(v)$ denotes the set of neighboring nodes, and ";" denotes the feature concatenation.

While Conv-MPN could simply replace MLP by CNN to form a feature update rule, that would require two CNN modules and hence more GPU memory. A node feature spreads across a volume and a simple pooling could keep all the information in a message without collisions. More precisely, instead of encoding a message for every pair of nodes, we just pool features across all the neighboring nodes to encode a message, followed by CNN to update a feature vector:

$$f_v \leftarrow \text{CNN}\left[f_v; \underset{w \in \mathbf{N}(v)}{\text{Pool}} f_w\right]. \tag{2}$$

We experimented max, sum, and mean poolings and the max pooling worked the best. We perform feature update up to 3 iterations due to the GPU memory limitation. The CNN module consists of 7 Conv-ReLU-BN blocks, which are not shared across different iterations. We use Conv-MPN(t=x) to denote our architecture with x iterations of convolutoinal message passing.

4.3. Building edge verification

After a few iterations of feature update, we put a CNN decoder to each node and output a confidence score, indicating if the corresponding building edge is true or not (See Fig. 3). The decoder first pass the feature into a 5 Conv-ReLU-BN blocks to convert feature into $64 \times 64 \times 128$. After that, feature are downsampled into $2 \times 2 \times 128$ via max-pooling. In the end, the feature is flattened into 512 dimensional feature vector, followed by a single fully connected layer to regress a confidence score.

[1] We do not keep latent features at the graph edges unlike standard MPN for memory consideration.

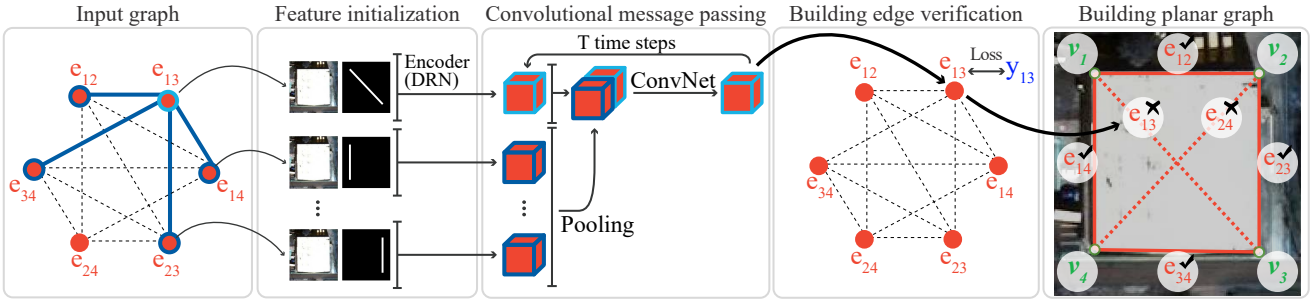

Figure 3. Conv-MPN Architecture. Given a graph, DRN encoder initializes a feature volume for each node. Convolutional message passing update feature volumes for T times. A building edge verification module uses a simple CNN decoder to estimate the confidence of a node (i.e., a building edge candidate).

4.4. Edge classification loss

We use the weighted binary cross-entropy loss:

$$\mathcal{L} = -\sum \mathcal{H} \log \hat{\mathcal{H}} - \lambda(1-\mathcal{H}) \log(1-\hat{\mathcal{H}}). \quad (3)$$

$\mathcal{H}$ and $\hat{\mathcal{H}}$ are the ground-truth and the prediction of the building edge confidence. $\lambda = 3$ is used to increase the weight on positive samples.

5. Experiments

We have implemented the proposed system in PyTorch. The learning rate is initialized to 5×10^{-4}, while we decay the rate by 0.8 when the testing loss does not decrease in 4 epochs. We terminate the training process when the testing loss does not decrease in 20 epochs.

Conv-MPN is GPU memory intensive due to the use of 3D feature volumes. A single NVIDIA TitanX GPU with 24G memory is used for all the experiments except for Conv-MPN (t=3), which requires two TitanX GPUs. We set the batch size to be 1, but accumulate gradients and update the parameters every 8 batches to suppress noisy gradients.

During training, the inference graph becomes too large to fit in a GPU memory for large buildings with many building corner candidates. For training Conv-MPN (and GNN in the comparative evaluations), we used 1215 buildings that have at most 15 building corner candidates. During testing, inference requires less memory and we simply apply the trained network on large buildings, which surprisingly works well for Conv-MPN in our experiments. Training Conv-MPN (t=1), Conv-MPN (t=2), and Conv-MPN (t=3) takes roughly 20, 30, and 40 hours.

5.1. Main results

Figure 4 shows representative planar graph reconstructions by Conv-MPN. The method is able to recover complex building structure beyond the Manhattan geometry without relying on any hand-crafted constraints or priors.

Next, we conduct comparative evaluations against five competing methods: PolyRNN++ [2], PPGNet [36], Hamaguchi *et al.* [14], L-CNN [37], and Nauata *et al.* [21] (See Table 1 and Fig. 5). Here, we provide brief summary of the five methods.

- PolyRNN++ traces the building external boundary in a recurrent fashion [2] and produces a 1D polygonal loop.
- PPGNet [36] uses CNN to detect corners and classifies their connections. However, the connection (i.e., edge) classification is independent of other connections, lacking in higher level geometric reasoning.
- Hamaguchi *et al.* [14] won the SpaceNet Building Footprint Extraction challenge [1]. The method uses CNN to produce binary masks of building footprints [14]. We convert the segmentation into a polygonal loop, and use OpenCV implementation of the Ramer-Douglas-Peucker algorithm with a threshold of 10 to simplify the curve.
- L-CNN [37] proposes an end-to-end neural network that detects corners and classifiers their connections. Like PPGNet, L-CNN also performs connection classification for each edge independently.
- Nauata *et al.* [21] is the current state-of-the-art for the problem, which detects 3 types of geometric primitives, classifies 2 types of pairwise primitive relationships, and uses integer programming to combine all the information into a building planar graph.

Nauata *et al.* relies on integer programming with hand-crafted objectives and structural constraints. The first 4 methods and Conv-MPN seek to learn geometric regularities or priors from examples instead.

Table 1 shows that Conv-MPN achieves the best higher-order (region) metrics among prior-free solutions. For the corner and edge metrics, Conv-MPN is not always the best. In particular, L-CNN outperforms Conv-MPN slightly on the edge metrics. However, as shown in Fig. 6, the graph structure from L-CNN is often broken, as edges are estimated independently. Figures 5 and 6 demonstrate that the region metrics best reflect our perceptual quality of the planar graph structure, in which Conv-MPN makes significant

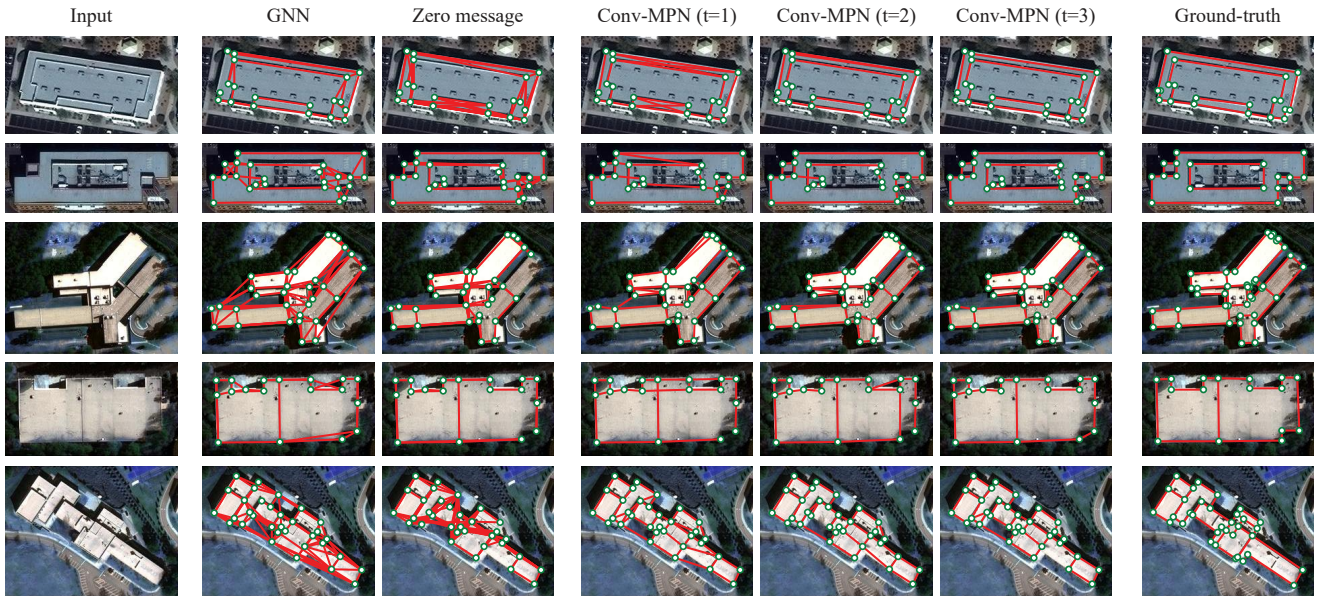

Figure 4. From left to right, an input RGB image, GNN, Zero message, Conv-MPN reconstructions after 1, 2, or 3 iterations of convolutional message passing and ground-truth.

Table 1. **Comparative evaluations**: Table shows precision and recall values, when the edge confidence threshold is set to 0.5. The cyan, orange, and magenta colors indicates the first, second, and third best results, respectively among the prior-free methods. Nauata *et al*. is the concurrent state-of-the-art method. This method is not prior-free and uses integer optimization with hand-crafted objectives and structural constraints.

Model	Corner			Edge			Region		
	Preci.	Recall	F1-score	Preci.	Recall	F1-score	Preci.	Recall	F1-score
PolyRNN++ [2]	49.6	43.7	46.4	19.5	15.2	17.1	39.8	13.7	20.4
PPGNet [36]	78.0	69.2	73.3	55.1	50.6	52.8	32.4	30.8	31.6
Hamaguchi *et al*. [14]	58.3	57.8	58.0	25.4	22.3	23.8	51.0	36.7	42.7
L-CNN [37]	66.7	86.2	75.2	51.0	71.2	59.4	25.9	41.5	31.9
Conv-MPN (t=3) [Ours]	77.9	80.2	79.0	56.9	60.7	58.7	51.1	57.6	54.2
Nauata *et al*. [21]	91.1	64.6	75.6	68.1	48.0	56.3	70.9	53.1	60.8

improvements. Note that Conv-MPN stays behind Nauata *et al*. [21] on the region F1-score, which requires hand-crafted objectives and structural constraints in a complex IP optimization formulation. We would like to emphasize again that Conv-MPN learns such priors and constraints all from examples automatically, which is a phenomenal feat and makes a big improvements against all the other prior-free solutions.

5.2. Ablation study

We verify the contributions of Conv-MPN architecture, in particular, on the effects of 1) feature volume representation and 2) message passing. Figures 6 and 7 provide the quantitative and qualitative comparisons, respectively.

Feature volume representation: We compare against a vanilla GNN, where we take the Conv-MPN architecture and replace $(64 \times 64 \times 32)$ feature volume by a 512 dimensional vector. The feature initialization, message passing, and line verification modules are modified accordingly to match up the feature dimensions (refer to the supplementary document for the details). We conduct message passing once both on Conv-MPN and GNN for clear comparison.

Figure 7 shows that GNN provides competitive results for the edge recall, but performs poorly on the other metrics. In particular, the performance gap is significant for the regions, which requires high-level geometry reasoning and demonstrates the power of our feature representation.

Message passing: We compare against two Conv-MPN variants that do not conduct message passing. The first variant (denoted as "per-edge classifier") simply does not

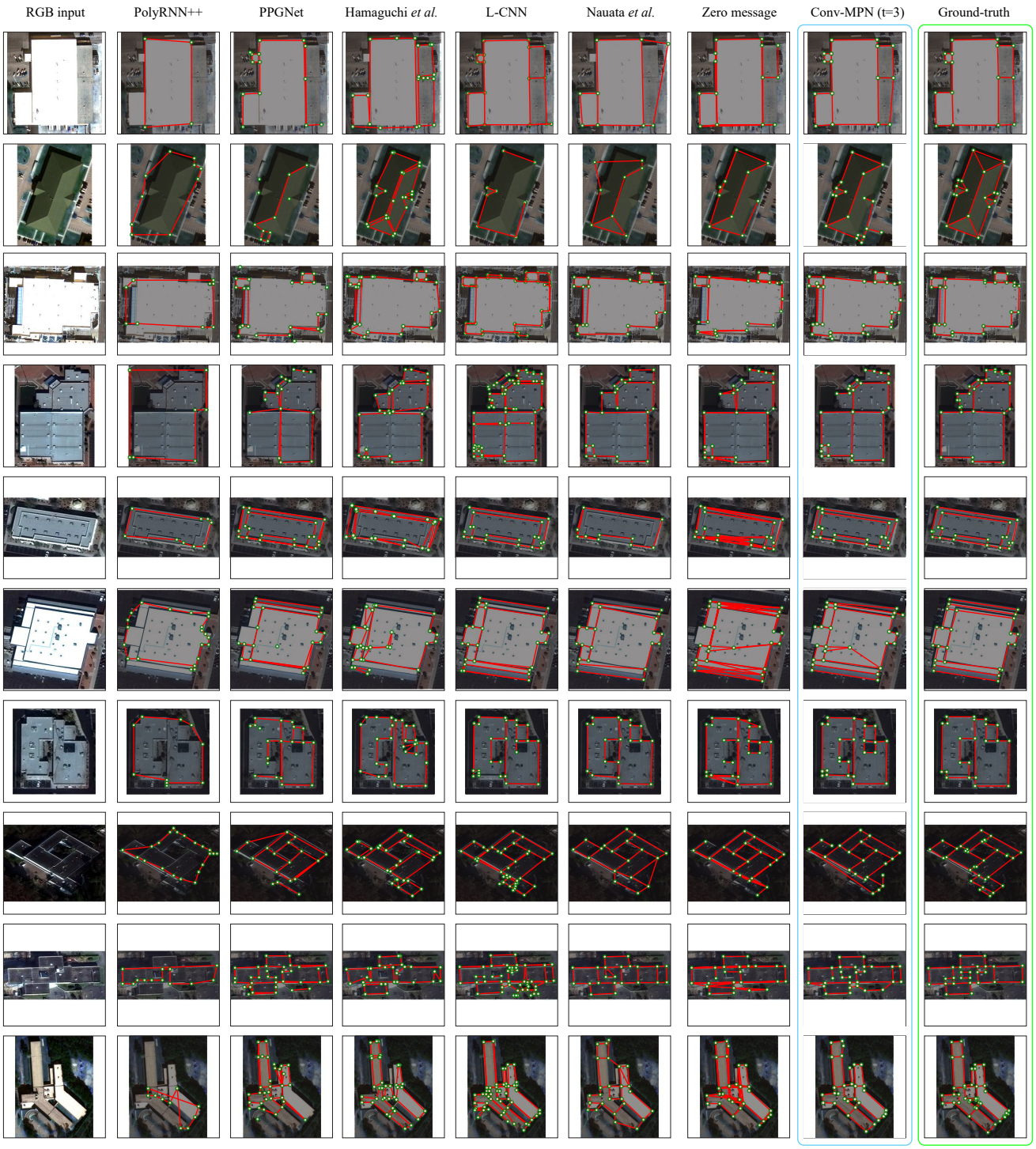

Figure 5. Comparative evaluations against competing methods. PolyRNN++ [2], PPGNet [36], Hamaguchi *et al.* [14], and L-CNN [37] are prior-free existing methods, all utilizing DNNs. Nauata *et al.* [21] is not prior-free. Zero message is a variant of our Conv-MPN without any message passing. Conv-MPN is our prior-free system.

exchange messages by cutting the inter-node connections. The second variant (denoted as "zero message") is equivalent to Conv-MPN (t=1), except that it always overwrites the pooled neighbor features with a value of 0.

Figure 7 shows that Conv-MPN (t=1) is superior to "per-edge classifier" and "zero message" in most metrics. In

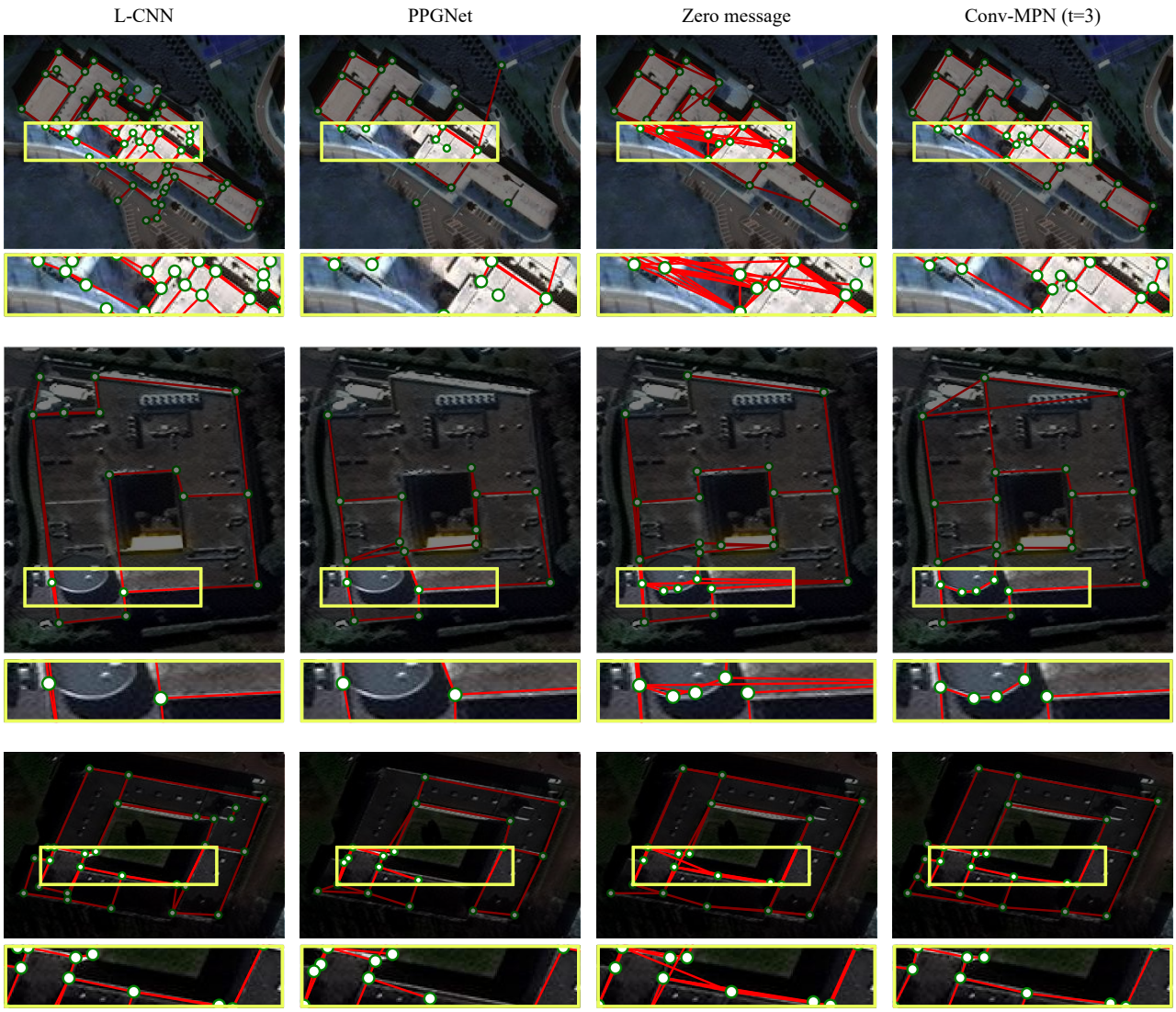

Figure 6. Close-up comparisons. From left to right, L-CNN[37], PPGNet[36], Zero message, and Conv-MPN(t=3). In the zooming area, we show the common mistakes that Conv-MPN can help to prevent. Typically, Conv-MPN helps removing the edge intersections, thin triangles and connecting missing edges.

particular, the performance gap in the region metrics are again significant, indicating that Conv-MPN effectively exchanges information via the convolutional message passing.

Figure 7 also shows how Conv-MPN improve reconstructions over multiple iterations of the convolutional message passing (See Figure. 4 for qualitative evaluations). The performance improvement is consistent and strong from no iterations to 1 and 2 iterations, where per-edge-classifier can be considered as Conv-MPN (t=0). Due to the memory limitation, Conv-MPN (t=3) is the largest model we trained, which shows the best results, where the performance improvements start to saturate.

5.3. Failure cases

Conv-MPN is far from perfect, where Figure 8 shows failure examples. The first major failure mode comes from missing corners. If a building corner is not detected, Conv-MPN will automatically miss all the incident structure. The second major failure mode is large buildings with 30 corner candidates or more, which do not appear in the training set due to the memory limitation.

6. Conclusion

This paper presents a novel message passing neural architecture Conv-MPN for structured outdoor architecture

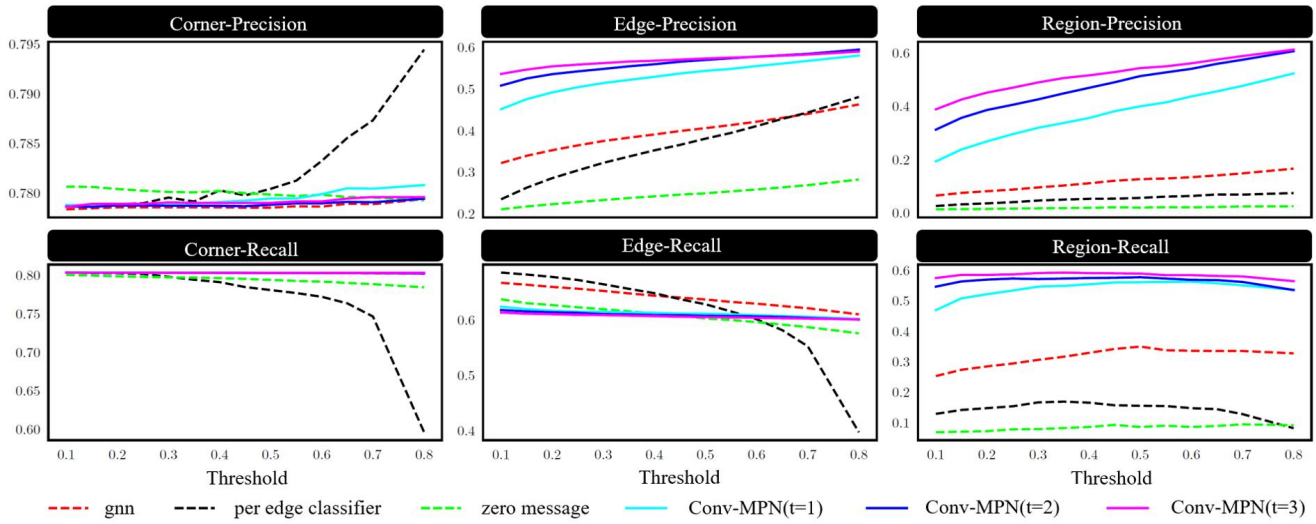

Figure 7. The precision and recall for the corners, edges and regions, while changing the edge confidence thresholds in the range $[0.1, 0.8]$ with an increment of 0.05. We plot the precision and recall separately for clarity. Note that y-axes for different plots are not in the same scale for better visualization.

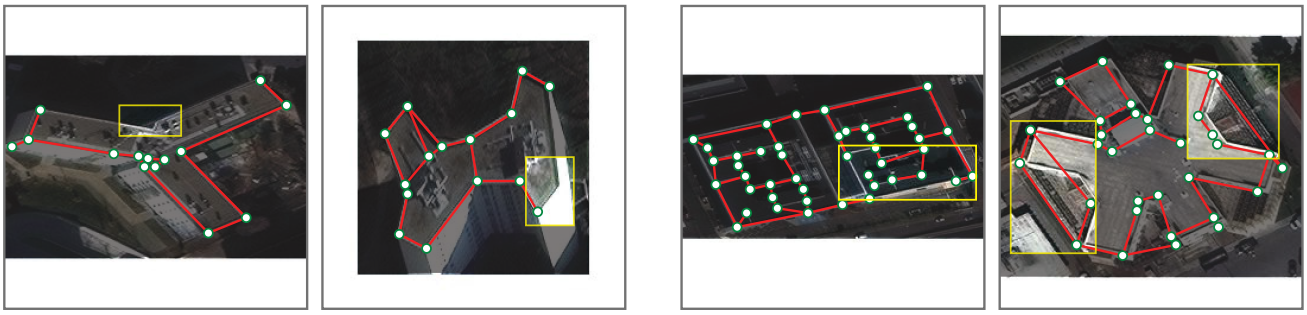

Figure 8. Failure cases. The left two examples suffer from missing corners by mask R-CNN. The right two examples show complex buildings, which Conv-MPN does not generalize well.

reconstruction. Our idea is simple yet powerful. Conv-MPN represents the feature associated with a node as a feature volume and utilizes CNN for message passing, while retaining the standard message passing neural architecture. Qualitative and quantitative evaluations verify the effectiveness of our idea and demonstrates significant performance improvements over the existing prior-free solutions. The main drawback is the extensive memory consumption, which is one of our future work to address.

The current popular approach to structured reconstruction is to inject domain knowledge as hand-crafted objectives or constraints into an optimization formulation. Conv-MPN learns all such priors from examples, then infer a planar graph structure form a single image. We believe that this paper has a potential to open a new line of graph neural network research for structured geometry reconstruction. We will share our code and data to promote further research.

Acknowledgement: This research is partially supported by NSERC Discovery Grants, NSERC Discovery Grants Accelerator Supplements, and DND/NSERC Discovery Grant Supplement. This research is also supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior / Interior Business Center (DOI/IBC) contract number D17PC00288. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

References

[1] SpaceNet on Amazon Web Services (AWS). "Datasets." The SpaceNet Catalog. Last modified April 30, 2018. https://spacenetchallenge.github.io/datasets/datasetHomePage.html, 2018. Online; accessed 19 October 2018. 2, 4

[2] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 859–868, 2018. 2, 4, 5, 6

[3] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1993–2001, 2016. 2

[4] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. 2, 3

[5] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013. 2

[6] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a polygon-rnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5230–5238, 2017. 2

[7] Dominic Cheng, Renjie Liao, Sanja Fidler, and Raquel Urtasun. Darnet: Deep active ray network for building segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7431–7439, 2019. 2

[8] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3844–3852. Curran Associates, Inc., 2016. 2

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016. 2

[10] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015. 2

[11] Adam Van Etten, Dave Lindenbaum, and Todd M. Bacastow. Spacenet: A remote sensing dataset and challenge series. *CoRR*, abs/1807.01232, 2018. 2

[12] Kevin Frans and Chin-Yi Cheng. Unsupervised image to sequence translation with canvas-drawer networks. *arXiv preprint arXiv:1809.08340*, 2018. 2

[13] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017. 2

[14] Ryuhei Hamaguchi and Shuhei Hikosaka. Building detection from satellite imagery using ensemble of size-specific detectors. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 223–2234. IEEE, 2018. 2, 4, 5, 6

[15] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. Learning to parse wireframes in images of man-made environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 626–635, 2018. 2

[16] Jiaye Wu Yasutaka Furukawa Jiacheng Chen, Chen Liu. Floor-sp: Inverse cad for floorplans by sequential room-wise shortest path. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 1

[17] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2

[18] Chen Liu, Jiaye Wu, and Yasutaka Furukawa. Floornet: A unified framework for floorplan reconstruction from 3d scans. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 201–217, 2018. 2

[19] Chen Liu, Jiajun Wu, Pushmeet Kohli, and Yasutaka Furukawa. Raster-to-vector: Revisiting floorplan transformation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2195–2203, 2017. 2

[20] Diego Marcos, Devis Tuia, Benjamin Kellenberger, Lisa Zhang, Min Bai, Renjie Liao, and Raquel Urtasun. Learning deep structured active contours end-to-end. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8877–8885, 2018. 2

[21] Nauata Nelson and Furukawa Yasutaka. Vectorizing world buildings: Planar graph reconstruction by primitive detection and relationship classification. *arXiv preprint arXiv:1912.05135*, 2019. 1, 2, 4, 5, 6

[22] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. 2

[23] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023, 2016. 2

[24] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[25] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 2

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 3

[27] Daniel Ritchie, Anna Thomas, Pat Hanrahan, and Noah Goodman. Neurally-guided procedural models: Amortized

inference for procedural graphics programs using neural networks. In *Advances in neural information processing systems*, pages 622–630, 2016. 2

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 3

[29] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019. 2

[30] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10965–10974, 2019. 2

[31] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 466–481, 2018. 2

[32] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. 2

[33] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017. 3

[34] Shanxin Yuan, Guillermo Garcia-Hernando, Björn Stenger, Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Pavlo Molchanov, Jan Kautz, Sina Honari, Liuhao Ge, et al. Depth-based 3d hand pose estimation: From current achievements to future goals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2636–2645, 2018. 2

[35] Huayi Zeng, Jiaye Wu, and Yasutaka Furukawa. Neural procedural reconstruction for residential buildings. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 737–753, 2018. 2

[36] Ziheng Zhang, Zhengxin Li, Ning Bi, Jia Zheng, Jinlei Wang, Kun Huang, Weixin Luo, Yanyu Xu, and Shenghua Gao. Ppgnet: Learning point-pair graph for line segment detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7105–7114, 2019. 2, 4, 5, 6, 7

[37] Yichao Zhou, Haozhi Qi, and Yi Ma. End-to-end wireframe parsing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 962–971, 2019. 2, 4, 5, 6, 7

[38] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4903–4911, 2017. 2