Vectorizing World Buildings: Planar Graph Reconstruction by Primitive Detection and Relationship Inference

Nelson Nauata and Yasutaka Furukawa

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

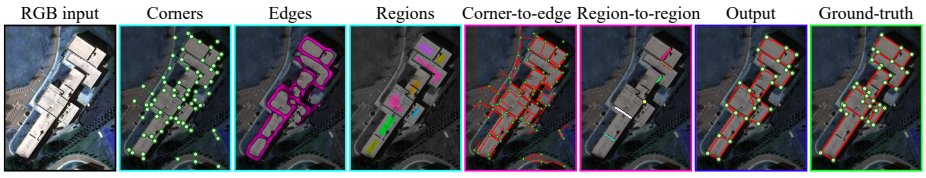

Fig. 1. The paper takes a RGB image, detects three geometric primitives (i.e., corners, edges, and regions), infers their relationships (i.e., corner-to-edge and region-to-region), and fuses the information via Integer Programming to reconstruct a planar graph.

Abstract. This paper tackles a 2D architecture vectorization problem, whose task is to infer an outdoor building architecture as a 2D planar graph from a single RGB image. We provide a new benchmark with ground-truth annotations for 2,001 complex buildings across the cities of Atlanta, Paris, and Las Vegas. We also propose a novel algorithm utilizing 1) convolutional neural networks (CNNs) that detects geometric primitives and infers their relationships and 2) an integer programming (IP) that assembles the information into a 2D planar graph. While being a trivial task for human vision, the inference of a graph structure with an arbitrary topology is still an open problem for computer vision. Qualitative and quantitative evaluations demonstrate that our algorithm makes significant improvements over the current state-of-the-art, towards an intelligent system at the level of human perception. We will share code and data.

Keywords: vectorization, remote sensing, deep learning, planar graph

1 Introduction

raster = in a grid

Human vision has a stunning perceptual capability in inferring geometric structure from raster imagery. What is remarkable is the holistic nature of our geometry perception. Imagine a task of inferring a building structure as a 2D graph from a satellite image (See Fig. 1). We learn structural patterns from examples quickly, utilize the learned patterns to augment the reconstruction process from incomplete data.

Computer Vision is still at its infancy in holistic reasoning of geometric structure. For low-level geometric primitives such as corners [20] or junctions [16], Convolutional Neural Networks (CNNs) have been an effective detector. Unfortunately, the task

Fig. 2. Sample satellite images used in our benchmark for the cities of Paris, Las Vegas and Atlanta (left-to-right). This image is part of the SpaceNet [9,1] corpus and is hosted as an Amazon Web Services (AWS) Public Dataset.

of high-level geometry reasoning, for example, the construction of CAD-quality geometry, is often only possible by the hands of expert modelers. CAD-level building reconstruction at a city-scale would enable richer architectural modeling and analysis across the globe, opening doors for broad applications in digital mapping, architectural study, or urban visualization/planning.

In an effort towards more holistic structured reconstruction techniques, this paper proposes a new 2D outdoor architecture vectorization problem, whose task is to reconstruct a 2D planar graph of outdoor building architecture from a single RGB image. While building segmentation from a satellite image has been a popular problem [1], their task is to extract only the external boundary as a 1D polygonal loop. Our problem seeks to reconstruct a planar graph of an arbitrary topology, including internal building feature lines that separate roof components and yield more high-fidelity building models (See Fig. 1). The inference of graph topology is the challenge in our problem, which is exacerbated by the fact that buildings on satellite images do not follow the Manhattan geometry due to the foreshortening effects through perspective projection.

exacerbated = esasperato

Our approach combines CNNs and integer programming (IP) to tackle the challenge. CNNs extract low- to mid-level topology information, in particular, detecting three types of geometric primitives (i.e., corners, edges, and regions) and inferring two types of pairwise primitive relationships (i.e., corner-to-edge and region-to-region relationships). IP consolidates all the information and reconstructs a planar graph.

We downloaded high-resolution satellite images from SpaceNet [9] corpus and annotated 2,001 complex buildings across the cities of Atlanta, Paris and Las Vegas as 2D polygonal graphs including internal and external architectural feature lines (See Fig. 2). Our qualitative and quantitative evaluations demonstrate significant improvements in our approach over the competing methods.

In summary, the contribution of the paper is two-fold: 1) A new outdoor architecture reconstruction problem as a 2D planar graph with a benchmark; 2) A hybrid algorithm combining primitive detectors, their relationship inference, and IP, which makes significant improvements over the existing state-of-the-art. We will share our code and data to promote further research.

2 Related work

Architectural reconstruction has a long history in Computer Vision. We first review building footprint extraction methods then focus our description on vector-graphics reconstruction techniques.

Building footprint extraction: In the SpaceNet Building Footprint Extraction challenge [1], a ground-truth building is represented as a set of pixels, ignoring the underlying vector graphics building structure. The winning method by Hamaguchi *et al*. [13] utilizes a multi-task U-Net for segmenting roads and buildings of different sizes, producing a binary building segmentation mask. Cheng *et al.* [8] utilizes CNNs for defining energy maps and optimizing polygon-based contours for building footprints. Acuna and Ling *et al*. [2] formulates the footprint extraction as the boundary tracing problem, finding a sequence of vertices forming a polygonal loop. Their method is designed for general object segmentation and tends to generate too many vertices. All these methods only extract the building footprint (i.e., external boundary) and ignores internal architectural feature lines.

Low-level reconstruction: Harris corner detection [26], Canny edge detection [3], and LSD line segment extractor [27] are popular traditional methods for low-level geometry detection. More recently, deep neural network (DNN) based approaches have been an active area of research [4]. By classifying the combination of incident edge directions, DNNs are also effective for the junction detection [16].

Mid-level reconstruction: Room layout estimation infers a graph of architectural feature lines from a single image, where nodes are room corners and edges are wall boundaries. Most approaches assume that the room shape is a 3D box, then solves an optimization problem with hand-engineered cost functions [15,25,17,6]. For a room beyond a box shape, Markov Random Field (MRF) infers detailed architectural structures [12] and Dynamic Programming (DP) searches for an optimal room shape [10,11], again via hand-engineered cost functions.

High-level reconstruction (knowledge): Given a prior knowledge about the overall geometric structure, corner detection alone suffices to reconstruct a complex graph structure. Human pose estimation is one of the most successful examples, where DNN is trained to detect human junctions with body types such as heads, right arms, and left

legs [5]. Their connections come from a prior knowledge (e.g. right hand is connected to right arm).

High-level reconstruction (optimization): A classical approach for CAD-quality 3D reconstruction is to inject domain knowledge as ad-hoc cost functions or processes in the optimization formulation [18]. The emergence of deep learning enabled robust solutions for low-level primitive detection. However, mid to high level geometric reasoning still relies on hand-crafted optimization [19,20].

Floor-SP [7] is the closest to our work, utilizing CNN-based corner, edge, and region detection with a sophisticated optimization technique to reconstruct floorplans. However, their method suffers from two limitations. First, Floor-SP does not allow any mistake in the region detection. [1] Second, Floor-SP requires principal directions and mostly Manhattan scenes, which is hardly true in our problem due to severe foreshortening effects. Our approach handles non-Manhattan scenes and utilizes region detection as soft-constraints.

High-level reconstruction (shape-grammar): A shape grammar defines rules of procedural shape generation [24]. Procedural reconstruction exploits the shape grammar in constraining the reconstruction process. Rectified building facade parsing is a good example, where heuristics and hand-engineered cost functions control the process [22,21]. More recently, DNNs learn to drive the procedural reconstruction for building facades [23] or top-down residential houses [29]. However, these shape grammars are too restrictive and do not scale to more complex large buildings in our problem.

3 2D architecture vectorization problem

This paper proposes a new building vectorization problem, where a building is to be reconstructed as a 2D planar graph from a single RGB image. We retrieved high-resolution satellite images from the SpaceNet [9] corpus, which are hosted as an Amazon Web Services Public Dataset [1] (See Fig. 2).

We annotated 2D planar graphs for 1010, 670, and 321 buildings from the cities of Atlanta, Paris, and Las Vegas, respectively. The average and the standard deviation of the number of corners, edges, and regions are 12.56/8.23, 14.15/9.53 and 2.8/2.19, respectively. Roughly 60% of the buildings have either 1 or 2 regions. 30% have 3 to 10 regions. The remaining 10% have more than 10 regions. We randomly chose 1601 training and 400 testing samples. We refer to the supplementary document for the complete distribution of samples per number of corners, edges and regions. Note a region is a space bounded by the edges, which is well-defined in our planar graphs. When multiple satellite images cover the same city region, we chose the one with the least off-Nadir angle to minimize the foreshortening effects.

For each building instance, we crop a tight axis-aligned bounding-box with 24 pixels margin, and paste to the center of a 256×256 image patch. The white color is padded at the background. We apply uniform shrinking if the bounding box is larger than 256×256. Figure 3 shows sample building annotations. We borrow the metrics introduced for

[1] Rooms are regions in their problem and can be detected easily. Our regions are roof segments and much less distinguishable.

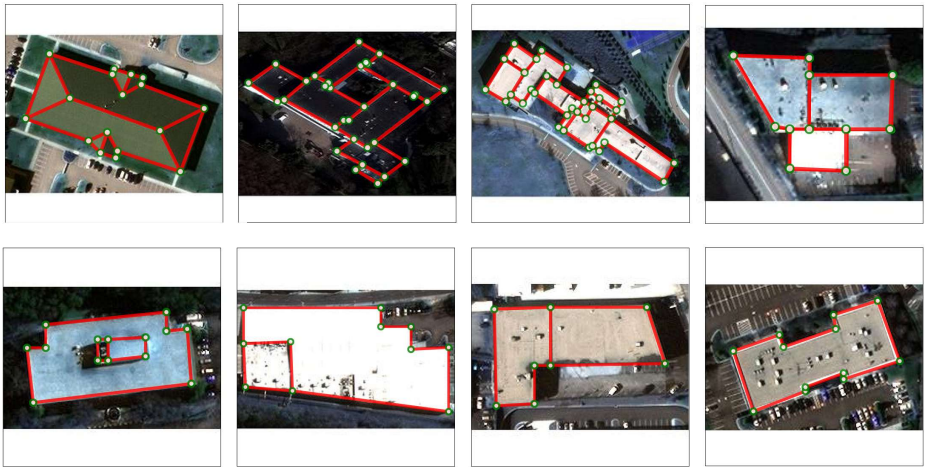

Fig. 3. Sample input RGB images and their corresponding planar graph annotations.

the floorplan reconstruction [7] (except for room++ metric), measuring the precision, recall, and f1-scores for the corner, edge, and region primitives. [2]

4 Algorithm

Our architecture vectorization algorithm consists of three modules: CNN-based primitive detection, CNN-based primitive relationship inference, and IP optimization (See Fig. 4). We now explain these three modules.

4.1 Primitive detection

We follow Floor-SP [7] and obtain corner candidates, an edge confidence image, and region candidates by standard CNN architecture (See Fig. 5), in particular, Fully Convolutional Network (FCN) for corners [16], Dilated Residual Networks (DRN) [28] for edges, and Mask-RCNN [14] for regions. Corner detections are thresholded at 0.2, where the confidence scores will also be used in the optimization. Edge information is estimated as a pixel-wise confidence score without thresholding. Every pair of corner candidates is considered to be an edge candidate. Region detections are thresholded at 0.5. We refer the full architectural specification to the supplementary document.

4.2 Primitive relationship classification

We classify two types of pairwise primitive relationships by CNNs (See Fig. 6).

[2] In short, a corner is declared to be correct if there exists a ground-truth corner within a certain distance. An edge is declared to be correct if both corners are declared to be correct. A region is declared to be correct if there exists a ground-truth region with more than 0.7 IOU. Our only change is to tighten the distance tolerance on the corner detection from 10 pixels to 8 pixels.

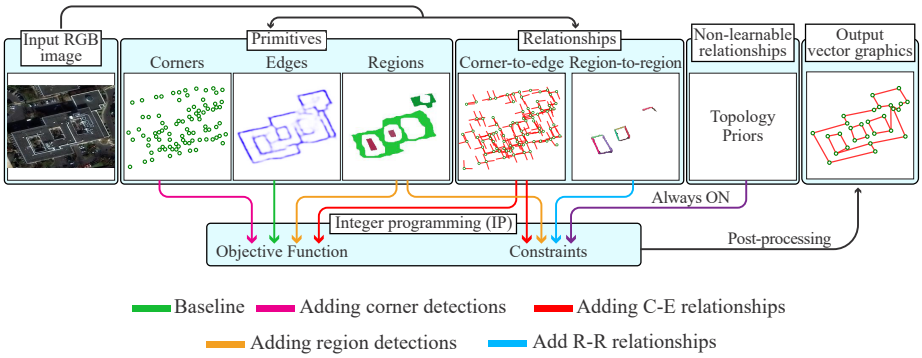

Fig. 4. System overview. Our pipeline detects geometric primitives, infers their relationships, and fuses all the information into integer programming to reconstruct a planar graph.

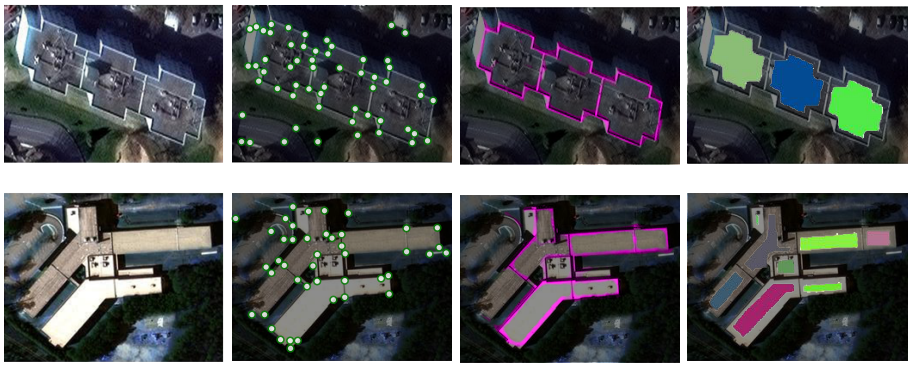

Fig. 5. Primitive detection. From left to right, the figure shows an input RGB image and its corner, edge (as pixel-wise confidence map) and region detections.

Corner-to-edge relationships: For every pair of a corner and an incident edge, we compute the confidence score by utilizing the junction-type inference technique by Huang *et al.*without any changes [16]. In short, we discretize 360 degrees around each detected corner into 15 angular bins, and add a module at the end of the corner detection head to estimate the presence of an edge in each bin. A corner to edge confidence score is simply set to the edge presence score in the corresponding bin. This score will be used by the objective function and the corner-to-edge relationship constraints in the optimization.

Region-to-region relationships: Given a RGB image and a pair of regions, we use Mask R-CNN [14] to find their common boundary as a pixel-wise segmentation mask. More precisely, we represent the input as a 5-channel image, where the two regions are represented as binary masks. The output is a set of common edges of the two regions, each of which is represented as a segmentation instance. When 2 regions do not have a shared boundary, Mask R-CNN should not output any segments. Detected segments are thresholded at 0.5, which are often reliable and the confidence scores will not be used

Fig. 6. Primitive relationships. From left to right, the figure shows an input RGB image and its corner-to-edge relationships visualized as junctions and room-to-room relationships visualized as common boundaries.

in the next optimization. The common boundaries will be used by the region-to-region relationship constraints in the optimization.

4.3 Geometric primitive assembly via IP

Integer Programming (IP) fuses detected primitives and their relationship information into a planar graph, where the inspiration of our formulation comes from the floorplan vectorization works by Liu et al. [19].

Objective function: Indicator variables are defined for each primitive: (1) I_{cor} for a corner $c \in \mathcal{C}$; (2) I_{edg} for an edge $e \in \mathcal{E}$; and (3) I_{reg} for a region $r \in \mathcal{R}$. After the optimization, we collect the set of primitives whose indicator variables are 1 as a building reconstruction. We also have an indicator variable I_{dir} for a corner to an incident edge direction relationship. The variable becomes 1, if a corner has an incident edge along the direction (with binning).

The objective function consists of the three terms:

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

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. With abuse of notation, c' and c'' denotes the end-points of an edge e.

The first objective term states that if an edge and its two end-points have high confidence scores (i.e., their product is at least 0.5^3), there is an incentive to select that edge. The second term suggests to select a corner and its incident edge direction if their confidence scores are high. The third objective term suggests to select as many regions as possible.

The maximization of the function is subject to four constraints, which are intuitive but require complex mathematical formulations. We here focus on explaining the ideas, while referring the details to the supplementary material. Note that we describe constraints as hard constraints, but turn them into soft constraints via slack variables before solving the problem for robustness. Lastly, after reconstructing a graph with IP, we apply a simple post-processing and eliminate a corner when it has two incident edges that are colinear with an error tolerance of 5 degrees.

Topology prior constraints: There are domain-specific constraints. First, if an edge is active, its two end-points must also be active. Second, no dangling edges are allowed, and every corner must have at least two incident edges. Third, no two edges can intersect, which ensures the planarity of the reconstructed graphs.

Region primitive constraints: Suppose a region is selected. All the edges that intersect with the region should be off. Similarly, the region must be surrounded by edges. We take a point at the region boundary and cast a ray outwards the region. We collect all the edges that intersect with the ray and enforce that at least one edge must be on. We sample such points at every 2 pixels around the region boundary.

Region-to-region relationship constraints: This constraint is similar in spirit to the region primitive constraint but is more powerful. Suppose a pair of regions have a common boundary prediction as a segmentation mask. The constraint enforces that at least one edge is selected nearby the mask. To be precise, we fit a line segment to the boundary segment, consider an orthogonal line segment (16 pixels in length) at the center. We collect all the edge primitives that intersect with the last line segment, and enforce that one edge will be chosen.

Corner-to-edge relationship constraints: If the incident indicator is on, the corresponding corner must be on, and one of the edges in the corresponding directional bin must be on. If two edges are incident to the same corner and within 5 degrees in angular distance, both edges cannot be on at the same time. If corner-to-edge compatibility score from the relationship inference is below 0.2, we do not allow any edges in that direction bin to be on.

5 Experimental results

We have implemented the proposed DNNs in PyTorch and the IP optimization in Python with Gurobi (a quadratic integer programming solver). We have used a workstation with Intel Xeon processors (2.2GHz) and NVidia GTX 1080 GPU with 11GB of RAM. The training usually takes 2 days for the primitive detectors and relationship classifiers. At

Table 1. Quantitative evaluations. P_C, P_E, and P_R denote corner, edge, and region primitive information, respectively. R_{CE} and R_{RR} denote corner-to-edge and region-to-region relationship information, respectively. The cyan, orange, and magenta indicate the top 3 results.

Model	Corner			Edge			Region		
	Prec.	Recall	f1	Prec.	Recall	f1	Prec.	Recall	f1
PolyRNN++ [2]	49.6	43.7	46.4	19.5	15.2	17.1	39.8	13.7	20.4
PPGNet [30]	78.0	69.2	73.3	55.1	50.6	52.8	32.4	30.8	31.6
Hamaguchi *et al.*[13]	58.3	57.8	58.0	25.4	22.3	23.8	51.0	36.7	42.7
L-CNN [31]	66.7	86.2	75.2	51.0	71.2	59.4	25.9	41.5	31.9
Floor-SP [7]	55.0	51.4	53.1	29.0	26.9	27.9	39.0	32.5	35.5
Ours (P_E)	75.0	41.5	53.4	52.4	15.6	24.1	66.7	0.5	1.0
Ours ($P_E + P_C$)	85.3	57.9	69.0	66.8	29.8	41.2	81.6	6.9	12.6
Ours ($P_E + P_C + R_{CE}$)	81.3	66.1	72.9	62.5	38.8	47.9	71.7	15.6	25.6
Ours ($P_E + P_C + R_{CE} + P_R$)	91.7	61.6	73.7	68.0	44.2	53.6	71.8	46.6	56.5
Ours ($P_E + P_C + R_{CE} + P_R + R_{RR}$)	91.1	64.6	75.6	68.1	48.0	56.3	70.9	53.1	60.8

test time, the network inference takes a fraction of a second, while IP normally takes less than 5 minutes, but can take up to 1 hour for some complex buildings.

5.1 Comparative evaluations

Table 1 shows our main result, comparing our approach against five competing methods: PolyRNN++ [2], PPGNet [30], Hamaguchi [13], L-CNN [31] and Floor-SP [7].

• PolyRNN++ traces the building external boundary in a recurrent fashion [2]. We fine-tuned all their released pretrained models, in particular, "Recurrent Decoder plus Attention", "Reinforcement Learning", "Evaluator Network", and "Gated Graph Neural Network". However, we found that fine-tuning only "Recurrent Decoder plus Attention" achieved the best results and is used in our evaluation.
• PPGNet [30] and L-CNN [31] were reproduced by simply taking the official code and training on our data.
• Hamaguchi [13] won the SpaceNet Building Footprint Extraction challenge [1]. The authors graciously trained their model and produced results using our data. Since their method produces pixel-wise binary masks of building footprints [13], which performs poorly in our metrics, we utilized the OpenCV implementation of the Ramer-Douglas-Peucker algorithm with a threshold of 10 to simplify the boundary curve.
• Floor-SP [7] is a state-of-the-art floorplan reconstruction system. Their algorithm is sensitive to the principal direction extraction (PDE), which becomes challenging against severe foreshortening effects in our problem. We tried to improve their PDE implementation without much success and used their default code, which extracts a mixture of 4 Manhattan frames (8 directions).

All the models were trained and tested on the same split. In the table, the last row (our system with all the features) has the best f1-scores for the corner and the region

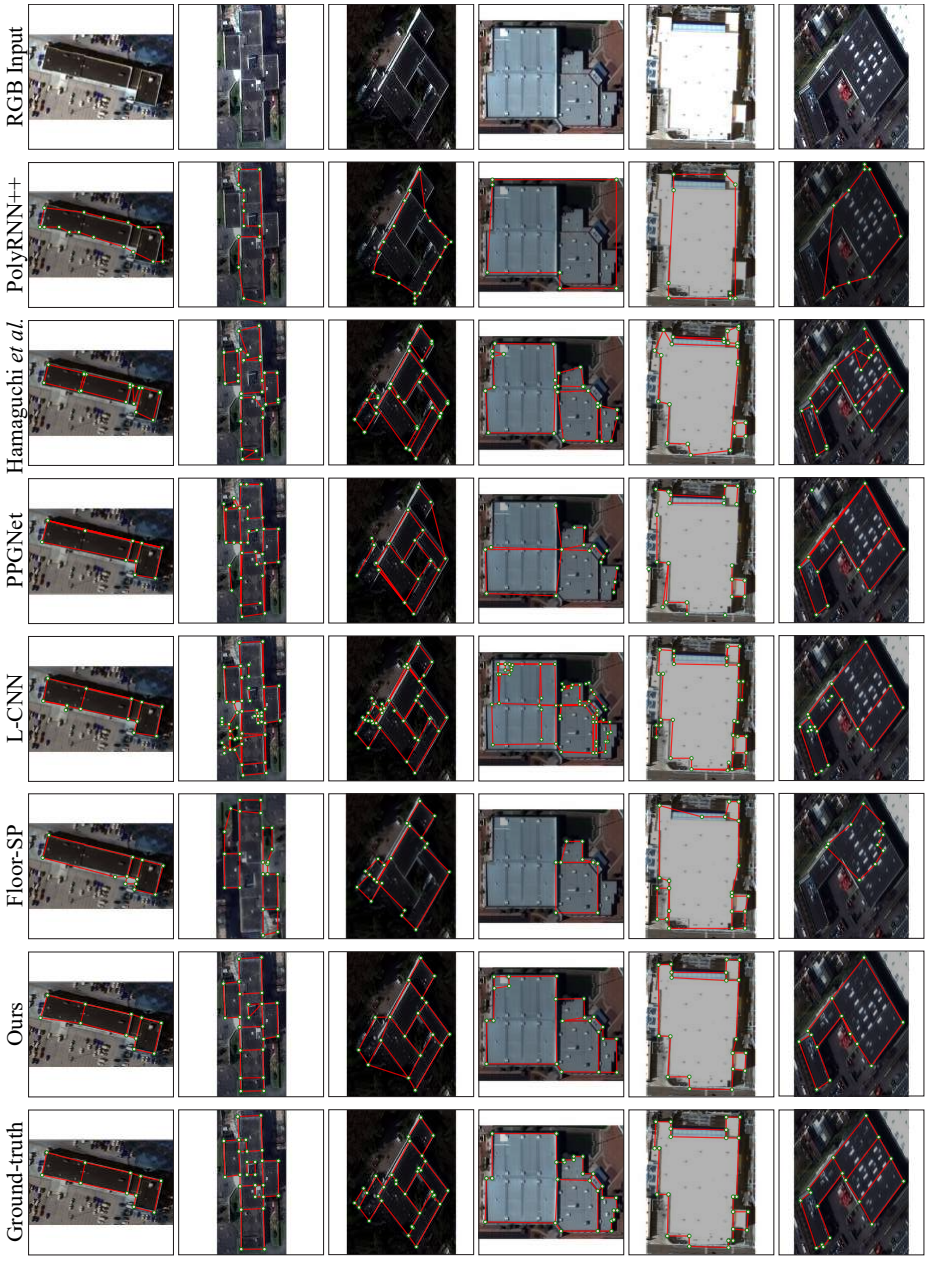

Fig. 7. Comparisons against five competing methods. The top row is the input image. The last row is the ground-truth. The supplementary document contains results for the entire testing set.

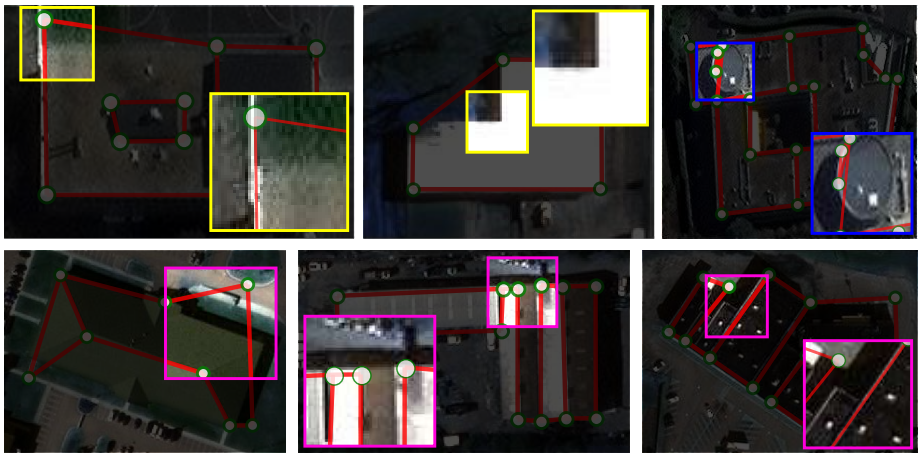

Fig. 8. Three major failure modes from our method: missed corner detections (yellow), curved buildings (blue), and weak image signals (magenta).

metrics, and the second best f1-score for the edge metric. Overall, our model makes steady improvements over the competing methods when more features are added, especially on the region metric, which is the most challenging and consistent with the visual reconstruction quality.

PPGNet and L-CNN achieve compelling f1-scores for the corners and the edges. L-CNN even outperforms our method for the edge f1-score. However, this metric is not a good indicator as illustrated in Figures 7 and 9. The figures show that the reconstruction results by PPGNet and L-CNN "look" reasonable at first sight. However, close examinations reveal that their results suffer from thin triangles, self-intersecting edges (i.e., the graph is not actually planar), and colinear edges. Their limitation comes from the fact that they infer edges independently. Their region metrics are far behind ours, which requires more holistic structure reasoning.

Floor-SP, on the other hand, performs poorly on all the metrics. There are two reasons. First, their shortest path algorithm at the core requires accurate principal directions, whose extraction is difficult without the Manhattan constraints in our problem. Second, they assume region detections to be 100% correct and cannot recover from region detection mistakes. As a result, the method often generates too many extraneous corners or completely miss parts of the graphs.

5.2 Ablation study

The bottom half of Table 1 and Fig. 10 verify the contributions of various components in our system: the three primitive detections and two relationship inference. We use symbols $\mathrm{P}_C, \mathrm{P}_E$, and P_R to denote if the corner, edge, and region primitive detections are used by our system. Similarly, R_{CE} and R_{RR} denote if the corner-to-edge and region-to-region relationships are used by our system.

Edge detections only ($\mathbf{P}_E$): Our first baseline utilizes only the edge detection results, that is, seeking to maximize ($\sum_{e\in\mathcal{E}}(c^e - 0.5)I_{edg}(e)$). In short, this baseline accepts all the edges whose score are above 0.5.

Adding corner detections ($\mathbf{P}_E, \mathbf{P}_C$): The second baseline ($\mathrm{P}_E + \mathrm{P}_C$) adds the corner detection results, seeking to maximize $\sum_{e\in\mathcal{E}}(e_{conf}c'_{conf}c''_{conf} - 0.5^3)I_{edg}(e)$. This baseline accepts all the edges, whose scores based on the corner and the edge detection are above 0.5^3. This effectively suppresses the corner and edge false positives, noticeably improving the precision and recall for both primitives.

Adding C-E relationships ($\mathbf{P}_E, \mathbf{P}_C, \mathbf{R}_{CE}$): This baseline adds the corner-to-edge relationship constraints and the corresponding objective in Eq. 1 to the formulation, enforcing the corner and edge variables to follow the predicted relationships. This change alone doubles the region f1-score.

Adding region detections ($\mathbf{P}_E, \mathbf{P}_C, \mathbf{R}_{CE}, \mathbf{P}_R$): With the addition of region detections, this baseline has the complete objective function, while the region constraints are also added. This baseline allows IP to conduct high-level geometry reasoning, and brings significant boost to the region metrics.

Adding R-R relationships ($\mathbf{P}_E, \mathbf{P}_C, \mathbf{R}_{CE}, \mathbf{P}_R, \mathbf{R}_{RR}$): Finally, our system with all the features achieve the best results, successfully reconstructing complex large buildings.

5.3 Failure cases

There are three major failure modes as illustrated in Fig. 8. First, our algorithm cannot recover from corners missed by the corner detector. Missing corners lead to missing incident graph structure or corrupted geometry. Second, our algorithm assumes piecewise linear structure and cannot handle curved buildings, while the system tries to approximate the shape as shown in the figure. Third, our system also fails when the image signal becomes weak and the detected primitive and/or relationship information also become weak.

6 Conclusion

This paper introduces a novel outdoor architecture vectorization problem with a benchmark, whose task is to reconstruct a building architecture as a 2D planar graph from a single image. The paper also presents an algorithm that uses CNNs to detect geometric primitives and infer their relationships, where IP fuses all the information into a planar graph through holistic geometric reasoning. The proposed method makes significant improvements over the existing state-of-the-art. The growing volume of remote sensing data collected by space and airborne assets facilitates myriad of scientific, engineering, and commercial applications in geographic information systems (GIS). We believe that this paper makes an important step towards the construction of an intelligent GIS system at the level of human perception. 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.

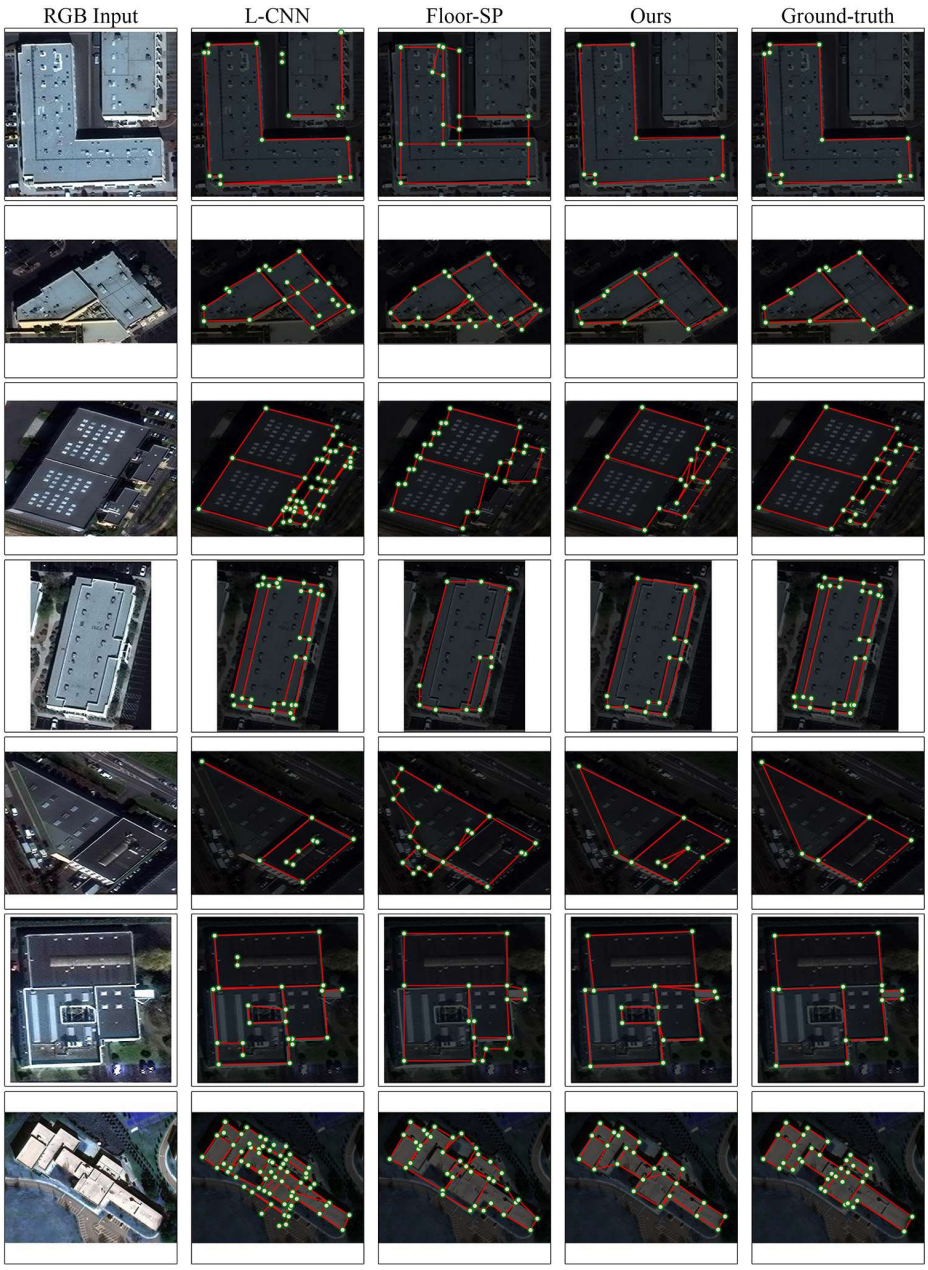

Fig. 9. More detailed comparisons against L-CNN [31] and Floor-SP [7].

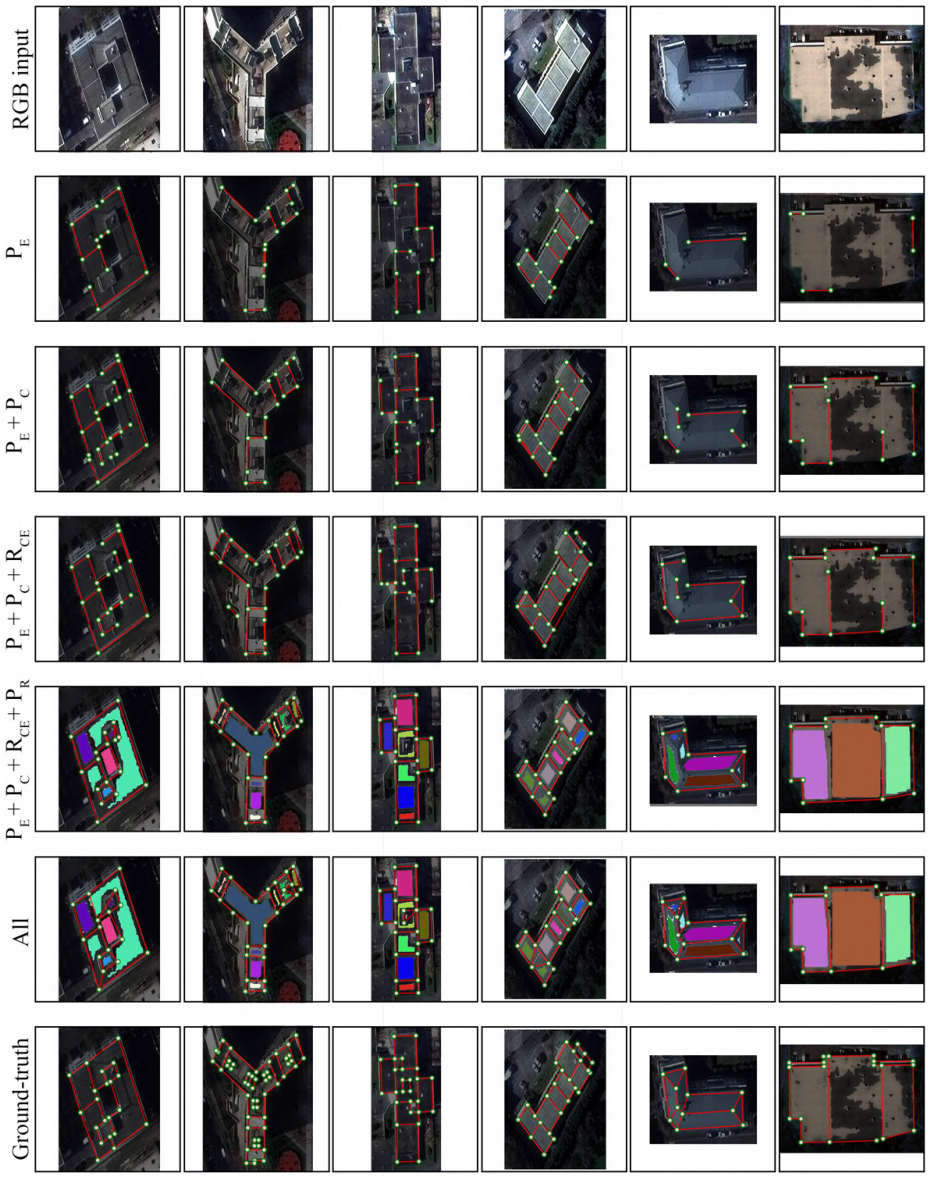

Fig. 10. Qualitative evaluation for ablation study. Figure shows results for input RGB image displayed in the first row for ablation experiments presented in Section 5.2 (same order) for multiple target buildings (columns) followed by ground-truth in the last row.

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. Acuna, D., Ling, H., Kar, A., Fidler, S.: Efficient interactive annotation of segmentation datasets with polygon-rnn++ (2018)
3. Canny, J.: A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence (6), 679–698 (1986)
4. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: Openpose: realtime multi-person 2d pose estimation using part affinity fields. arXiv preprint arXiv:1812.08008 (2018)
5. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7291–7299 (2017)
6. Chao, Y.W., Choi, W., Pantofaru, C., Savarese, S.: Layout estimation of highly cluttered indoor scenes using geometric and semantic cues. In: International Conference on Image Analysis and Processing. pp. 489–499. Springer (2013)
7. Chen, J., Liu, C., Wu, J., Furukawa, Y.: Floor-sp: Inverse cad for floorplans by sequential room-wise shortest path. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2661–2670 (2019)
8. Cheng, D., Liao, R., Fidler, S., Urtasun, R.: Darnet: Deep active ray network for building segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7431–7439 (2019)
9. Etten, A.V., Lindenbaum, D., Bacastow, T.M.: Spacenet: A remote sensing dataset and challenge series. CoRR (2018), http://arxiv.org/abs/1807.01232
10. Flint, A., Mei, C., Murray, D., Reid, I.: A dynamic programming approach to reconstructing building interiors. In: European Conference on Computer Vision. pp. 394–407. Springer (2010)
11. Flint, A., Murray, D., Reid, I.: Manhattan scene understanding using monocular, stereo, and 3d features. In: 2011 International Conference on Computer Vision. pp. 2228–2235. IEEE (2011)
12. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Manhattan-world stereo. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1422–1429. IEEE (2009)
13. Hamaguchi, R., Hikosaka, S.: Building detection from satellite imagery using ensemble of size-specific detectors. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 223–2234. IEEE (2018)
14. 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)
15. Hedau, V., Hoiem, D., Forsyth, D.: Recovering the spatial layout of cluttered rooms. In: Computer vision, 2009 IEEE 12th international conference on. pp. 1849–1856. IEEE (2009)
16. 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)
17. Lee, C.Y., Badrinarayanan, V., Malisiewicz, T., Rabinovich, A.: Roomnet: End-to-end room layout estimation. arXiv preprint arXiv:1703.06241 (2017)
18. Lin, H., Gao, J., Zhou, Y., Lu, G., Ye, M., Zhang, C., Liu, L., Yang, R.: Semantic decomposition and reconstruction of residential scenes from lidar data. ACM Transactions on Graphics (TOG) **32**(4), 66 (2013)
19. Liu, C., Wu, J., Kohli, P., Furukawa, Y.: Raster-to-vector: Revisiting floorplan transformation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2195–2203 (2017)

20. Liu, C., Wu, J., Furukawa, Y.: Floornet: A unified framework for floorplan reconstruction from 3d scans. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 201–217 (2018)
21. Liu, H., Zhang, J., Zhu, J., Hoi, S.: Deepfacade: A deep learning approach to facade parsing. pp. 2301–2307 (08 2017). https://doi.org/10.24963/ijcai.2017/320
22. Martinović, A., Mathias, M., Weissenberg, J., Van Gool, L.: A three-layered approach to facade parsing. In: Proceedings of the 12th European Conference on Computer Vision - Volume Part VII. pp. 416–429. ECCV'12, Springer-Verlag, Berlin, Heidelberg (2012)
23. Nishida, G., Bousseau, A., Aliaga, D.G.: Procedural modeling of a building from a single image. In: Computer Graphics Forum. vol. 37, pp. 415–429. Wiley Online Library (2018)
24. Parish, Y.I., Müller, P.: Procedural modeling of cities. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. pp. 301–308. ACM (2001)
25. Schwing, A.G., Hazan, T., Pollefeys, M., Urtasun, R.: Efficient structured prediction for 3d indoor scene understanding. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 2815–2822 (2012)
26. Szeliski, R.: Computer vision: algorithms and applications. Springer Science & Business Media (2010)
27. Von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: Lsd: a line segment detector. Image Processing On Line **2**, 35–55 (2012)
28. Yu, F., Koltun, V., Funkhouser, T.A.: Dilated residual networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 636–644 (2017)
29. Zeng, H., Wu, J., Furukawa, Y.: Neural procedural reconstruction for residential buildings. In: The European Conference on Computer Vision (ECCV) (September 2018)
30. Zhang, Z., Li, Z., Bi, N., Zheng, J., Wang, J., Huang, K., Luo, W., Xu, Y., Gao, S.: Ppgnet: Learning point-pair graph for line segment detection. arXiv preprint arXiv:1905.03415 (2019)
31. Zhou, Y., Qi, H., Ma, Y.: End-to-end wireframe parsing. arXiv preprint arXiv:1905.03246 (2019)