PPGNet: Learning Point-Pair Graph for Line Segment Detection

Ziheng Zhang*, Zhengxin Li*, Ning Bi, Jia Zheng, Jinlei Wang, Kun Huang, Weixin Luo, Yanyu Xu and Shenghua Gao

School of Information Science and Technology, ShanghaiTech University, China

SVIP LAB ShanghaiTech Vision and Intelligent Perception LAB

CVPR LONG BEACH CALIFORNIA June 16-20, 2019

Highlights

√ **Novel Formulation**

Detect **line segments** by inferring a **point-pair graph** from an image.

√ **Novel Network**

We introduce the PPGNet, which **directly infer** point-pair **graph** from given **images**.

√ **Competitive Performance**

The experiments have shown the **effectiveness** and good **generalizability** of PPGNet.

√ **General Framework**

PPGNet is a **general framework** to infer a **graph** from an **image**.

Junction-line Graph Representation

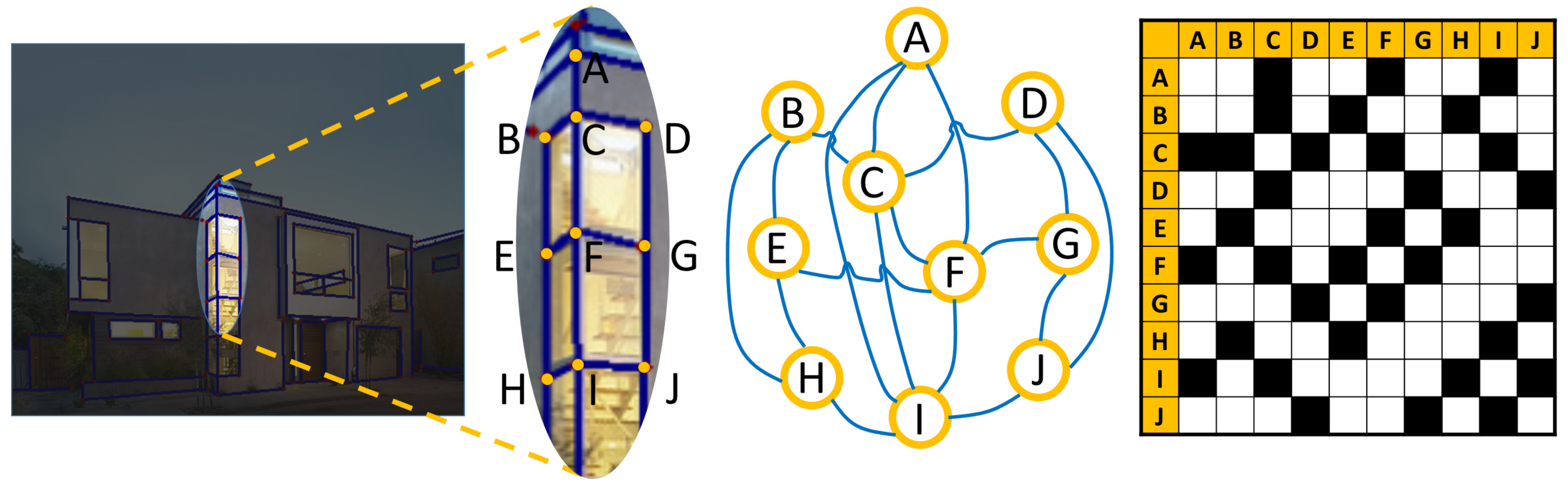

Figure 1. *Demonstration of junction-line graph representation G={V, E}. (a) an sample image patch with 10 junctions (V); (b) the graph which describes the connectivity of all junctions (G); (c) the adjacency matrix of all junctions (E).*

• End-Point vs. Graph-Based Representation

Both end-point (EPR) and graph-based (GBR) representation are able to describe all line segments in an image, and one can easily convert to another. However their differences emerge when we try to infer them from an image. EPR implicitly regards **line segments as objects**. As a consequence, for long line segments containing many junctions, which are very common in the real world (see above figure), EPR based methods need to handle highly overlapped line segment objects. On the other hand, GBR recognizes **line segments as relations** between junctions, thus naturally solve the problem.

The PPGNet

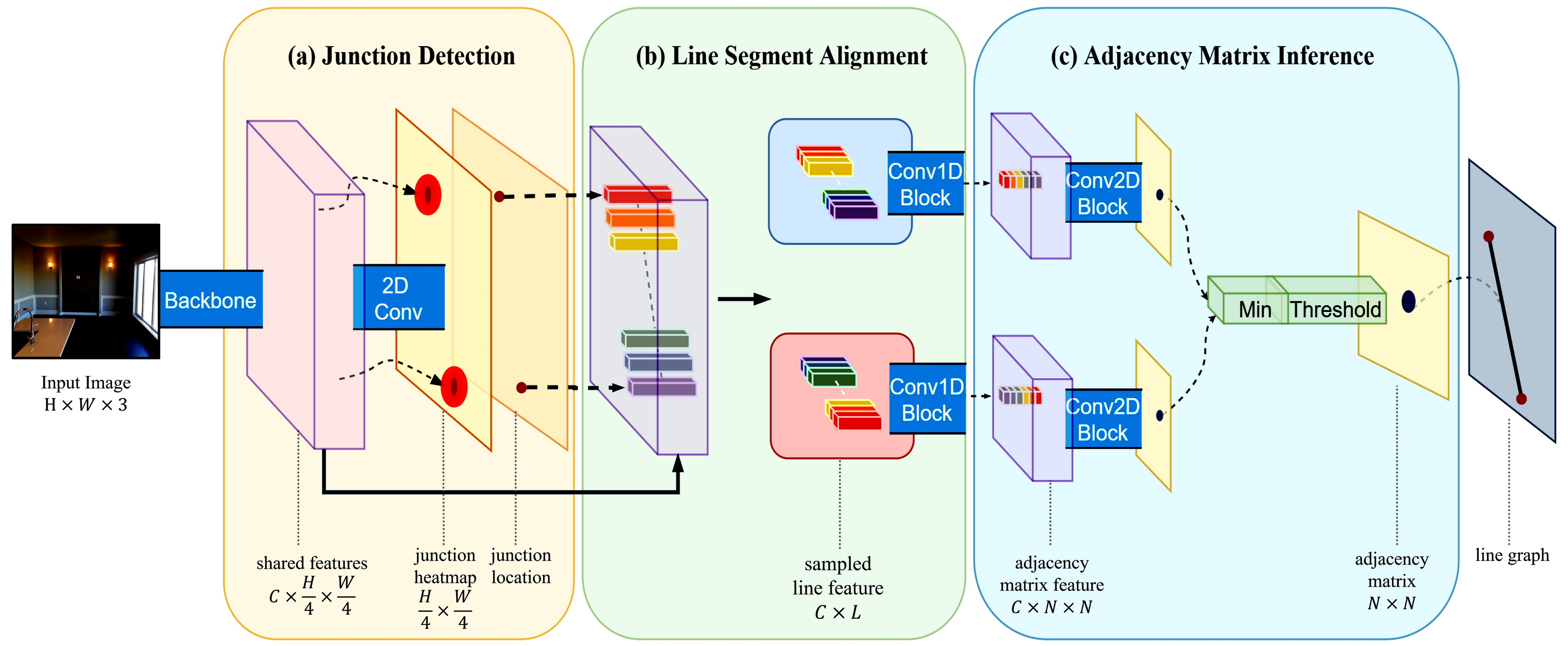

Figure 2. *The PPGNet architecture. First, the backbone computes shared features of size C × H/4 × W/4 for Junction detection and adjacency matrix inference. Second, the Junction Detection Module output a list of N junctions. Third, each junction pair is formed as two line segment candidates of different directions, over which features are evenly sampled into two feature matrix of size C × L. After that, we apply 1D convolution over each feature matrix, which outputs a feature vector of size C. Fourth, each feature vector is used by the Adjacency Matrix Inference Module to infer the connectivity of the corresponding junction pairs .*

• Loss Function

Both junction heatmap and adjacency matrix are supervised using binary cross entropy loss:

$$\mathcal{L}_{junc} = -\sum_i \tilde{H}_i \log H_i + (1-\tilde{H}_i)\log(1-H_i)$$
$$\mathcal{L}_{adj} = -\sum_i \tilde{A}_i \log A_i + (1-\tilde{A}_i)\log(1-A_i)$$

and the final loss is the weighted sum of two losses:

$$\mathcal{L} = \lambda_{junc}\mathcal{L}_{junc} + \lambda_{adj}\mathcal{L}_{adj}$$

, where $\tilde{H}_i$ and H_i are the elements of prediction and ground truth of junctions, respectively, $\tilde{A}_i$ and A_i are the elements of the prediction and the ground truth of the adjacency matrix, respectively.

• Training & Evaluation Details

All modules are jointly optimized using Stochastic Gradient Decent (SGD), with *lr*=0.2, *weight_decay*= 5×10^{-4}, and *momentum*=0.9, except for all normalization layers, of which *weight_decay* is set to zero. The backbone network is initialized with parameters pretrained for segmentation task on the MIT ADE20K dataset [3], and other modules are initialized with *kaiming initialization* [2], as the common practice. During the training phase, AMIM infers adjacency matrix for **ground truth junctions** instead of junctions predicted by JDM because we do not have corresponding ground truth adjacency matrix for supervision. During evaluating phase, junctions and adjacency matrix are jointly estimated by our PPGNet.

Experiments & Results

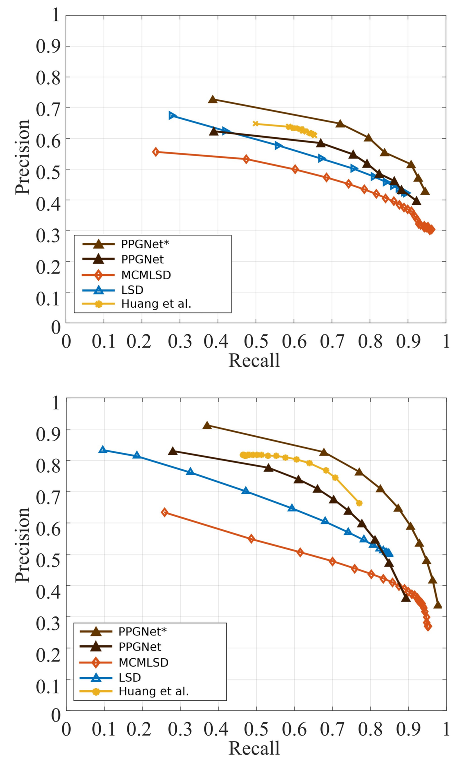

Figure 3. *Precision-Recall curves of our PPGNet and state of the art methods evaluated on (a) Wireframe dataset and (b) York-Urban dataset.*

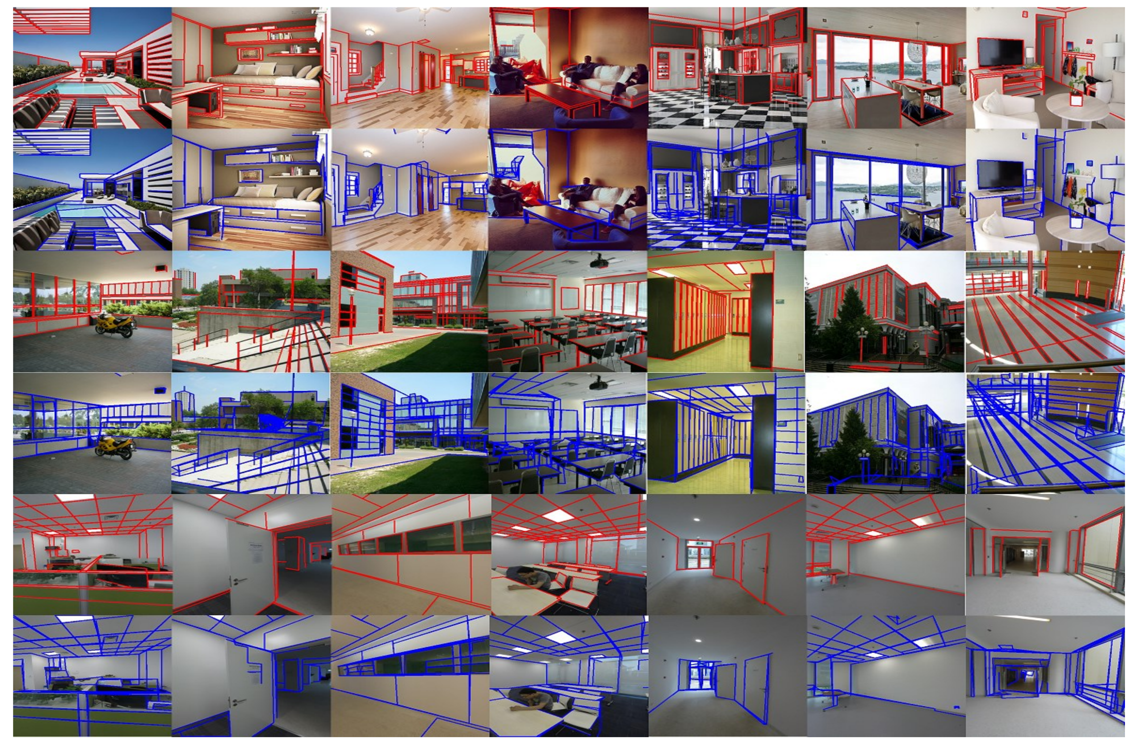

Figure 4. *Qualitative evaluation of our line segment detection method. 1st row: ground truth (Wireframe); 2nd row: prediction (Wireframe); 3rd row: ground truth (York Urban); 4th row: prediction (York Urban); 5th row: ground truth (Our dataset); 6th row: prediction (Our dataset)*

Reference

[1] Zhang Ziheng, 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, pp. 7105-7114. 2019.

[2] He Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification." In Proceedings of the IEEE international conference on computer vision, pp. 1026-1034. 2015.

[3] Zhou Bolei, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. "Semantic understanding of scenes through the ade20k dataset." International Journal of Computer Vision 127, no. 3 (2019): 302-321.

Find paper, code, data and this poster itself HERE!