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

Figure 2. The PPGNet architecture. First, the backbone computes shared features of size $C \times H/4 \times W/4$ for junction detection and adjacency matrix inference. Second, the Junction Detection Module outputs a list of junctions. Then, each junction pair is formed as two line segment candidates of different directions, over which features are evenly sampled into two feature matrices of size $C \times L$. 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.

Experiments & Results

![Precision-Recall curves of our PPGNet and state-of-the-art methods evaluated on (a) WireframeUrban dataset and (b) YorkUrban dataset.](image)

![Qualitative evaluation of our line segment detection method.](image)

**Highlights**

- Novel Formulation
  Dataset line segments as inferring a point-pair graph from an image.
- Novel Network
  We introduce the PPGNet, which directly infers point-pair graphs 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**

(a) (b) (c)

![Demonstration of junction-line graph representation G(V,E).](image)

- 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**

![The PPGNet](image)

**Loss Function**

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

$$L_{junc} = - \sum_j h_j \log H_j - (1 - h_j) \log (1 - H_j)$$

$$L_{adj} = - \sum_{ij} A_{ij} \log A_{ij} - (1 - A_{ij}) \log (1 - A_{ij})$$

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

$$L = \lambda_{junc} L_{junc} + \lambda_{adj} L_{adj}$$

where $ij$ and $jj$ are the elements of the prediction and ground truth of junctions, respectively, and $ij$ and $jj$ 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 Descent (SGD), with $\eta=0.2$, $\text{weight decay}=5 \times 10^{-5}$, and momentum=0.9, except for all normalization layers, of which $\text{weight decay}$ 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.

**Reference**


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