TP-LSD:
Tri-Points Based Line Segment Detector

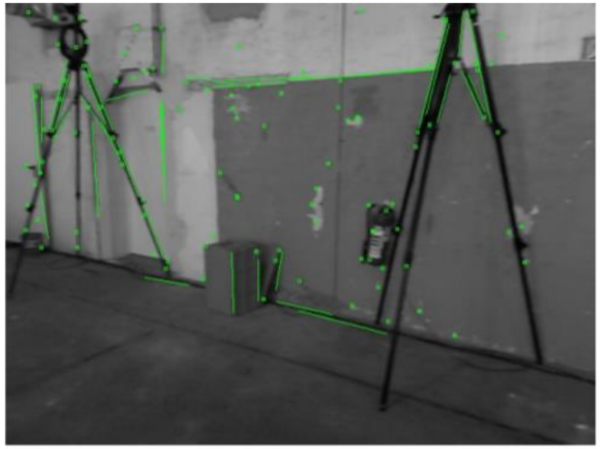
Siyu Huang[1], Fangbo Qin[2], Pengfei Xiong[1], Ning Ding[1], Yijia He[1]*, Xiao Liu[1]

[1] Megvii Technology [2] Institute of Automation, Chinese Academy of Sciences

Background: Applications of Line Segment Detection

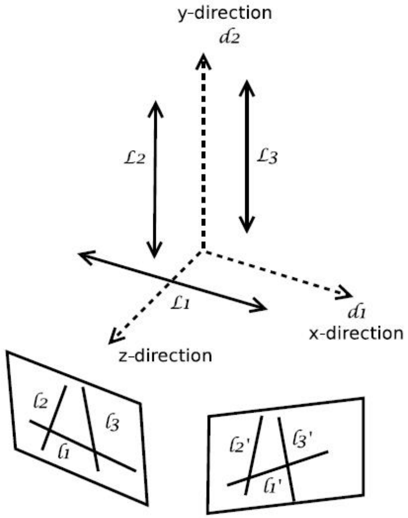
Simultaneous localization and mapping (SLAM)

(a) Textured scene



(b) Low-textured scene

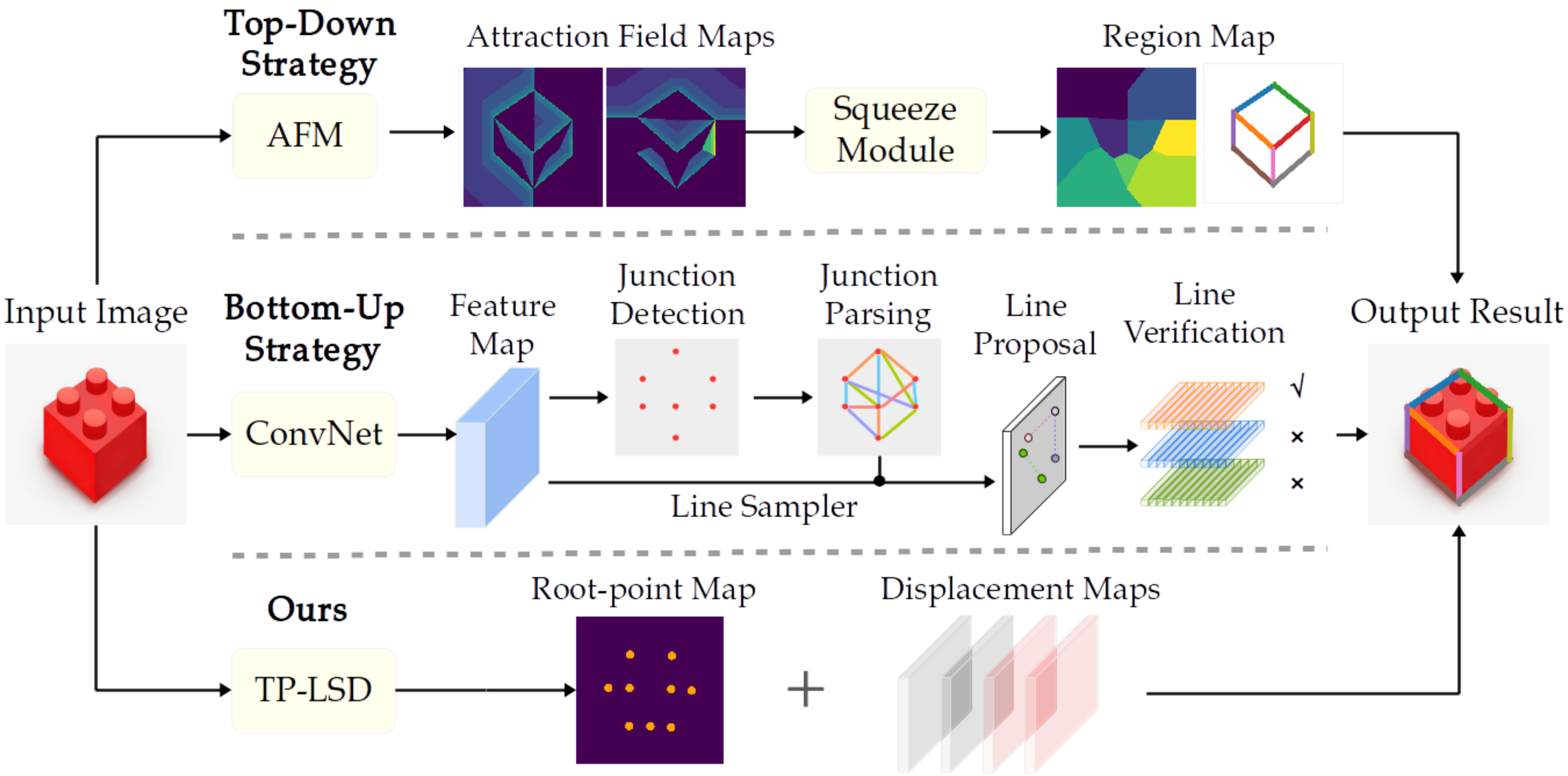
Pose Estimation



(a)

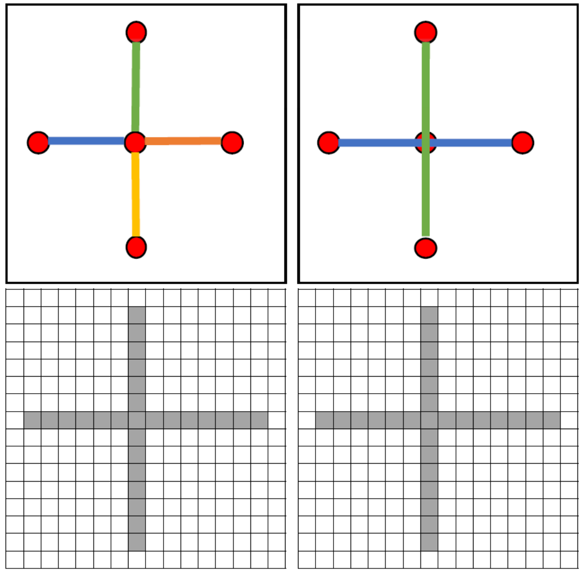
(b)

Background: Inference speed is limited by the two-step methods

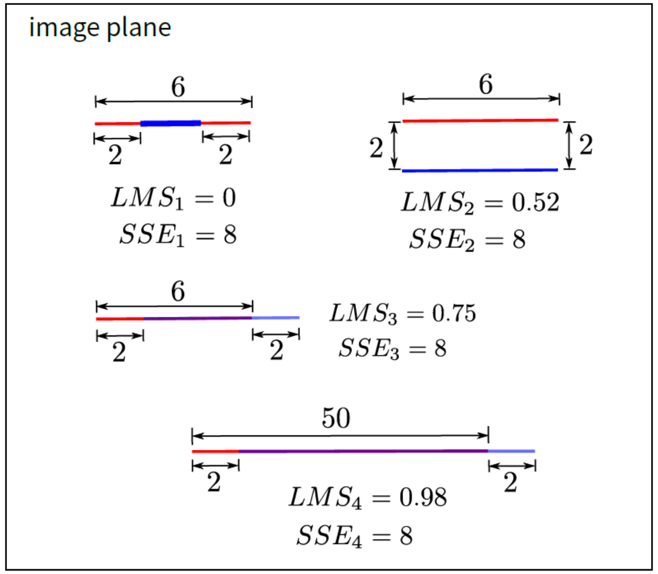


- Top-down strategy
  - Affected by regional textures
  - Lack explicit definition of endpoints
- Bottom-up strategy
  - Inaccurate junction predictions
  - Numerous parsing verification
- Ours one-step strategy
  - 6 times faster than existing methods

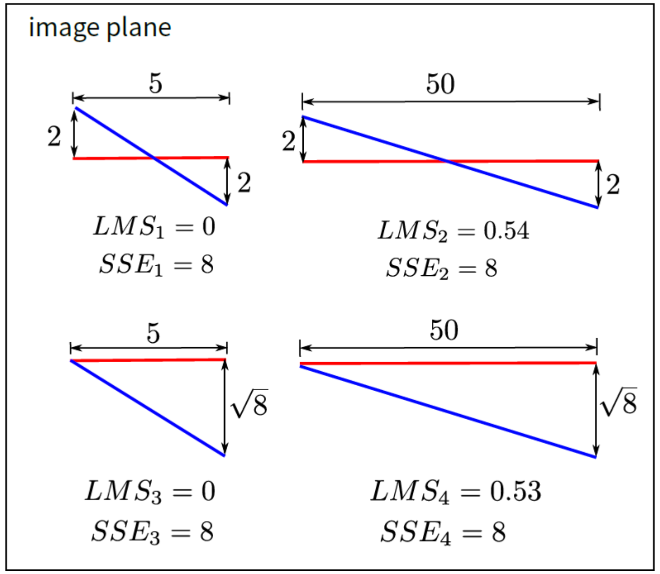
Background: Current metrics cannot reflect the various relationships between line segments



(a) incorrect connectivity[1]



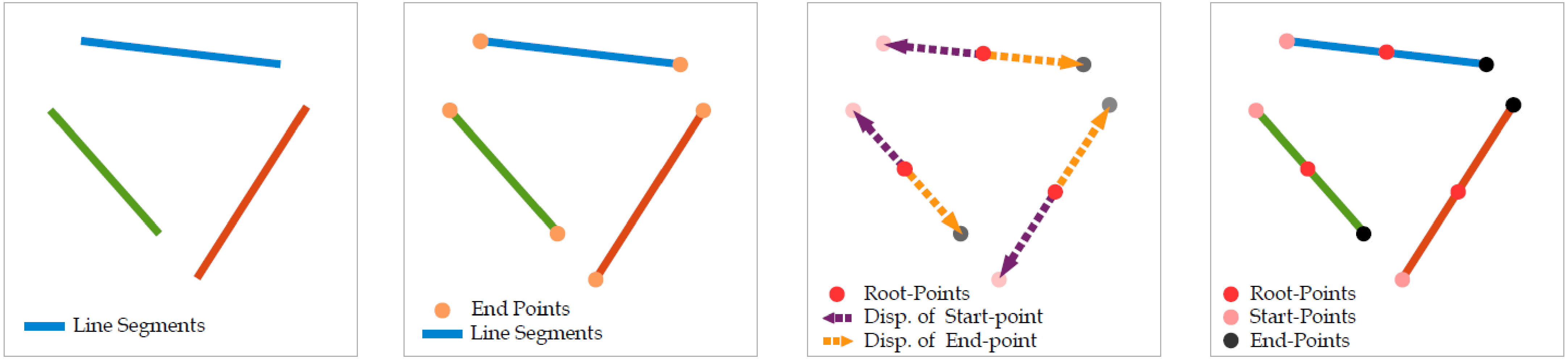
(b) Overlapped



(c) Intersection

[1] Zhou Y, Qi H, Ma Y. End-to-end wireframe parsing[C]//Proceedings of the IEEE International Conference on Computer Vision. 2019: 962-971.

Triplet-point representation

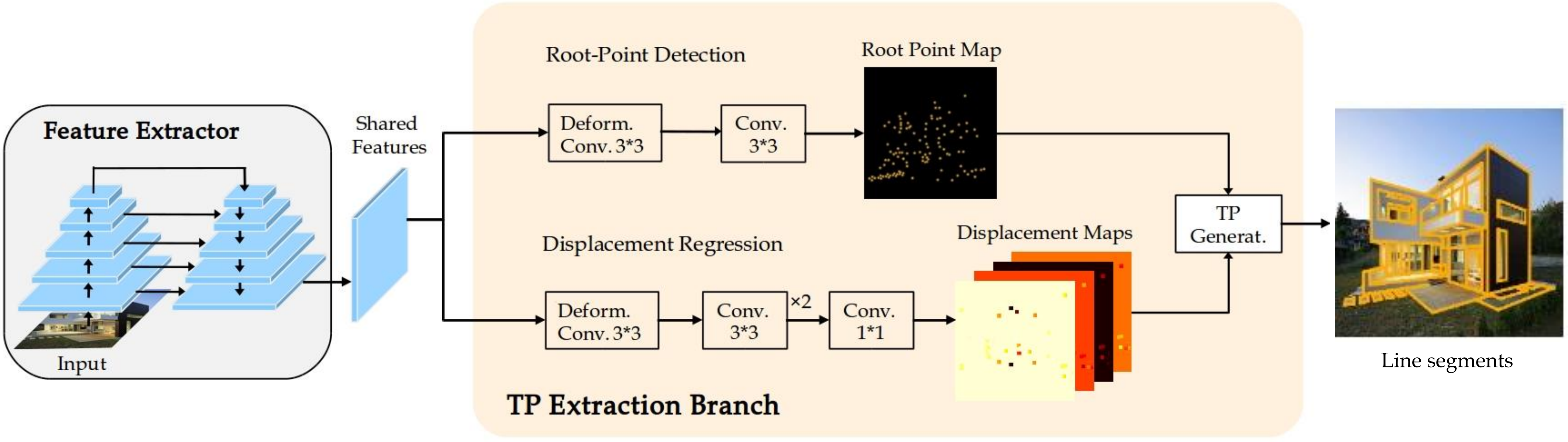


(a) Pixel based (b) End-points based (c) Tri-point based (d) Vectorized lines

$$(x_s, y_s) = (x_r, y_r) + d_s(x_r, y_r)$$
$$(x_e, y_e) = (x_r, y_r) + d_e(x_r, y_r)$$

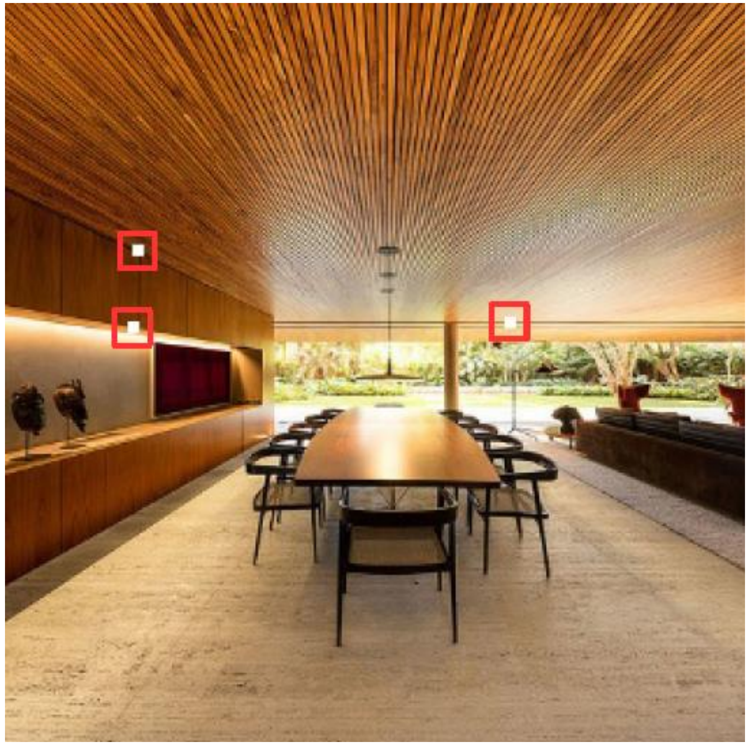
Notation: (x_s, y_s) denotes the root-point of a line segment, (x_s, y_s), (x_e, y_e) represent its start-point and end-point, respectively. $d_s(x_r, y_r)$, $d_e(x_r, y_r)$ denote the predicted 2D displacements.

The development of TP-LSD

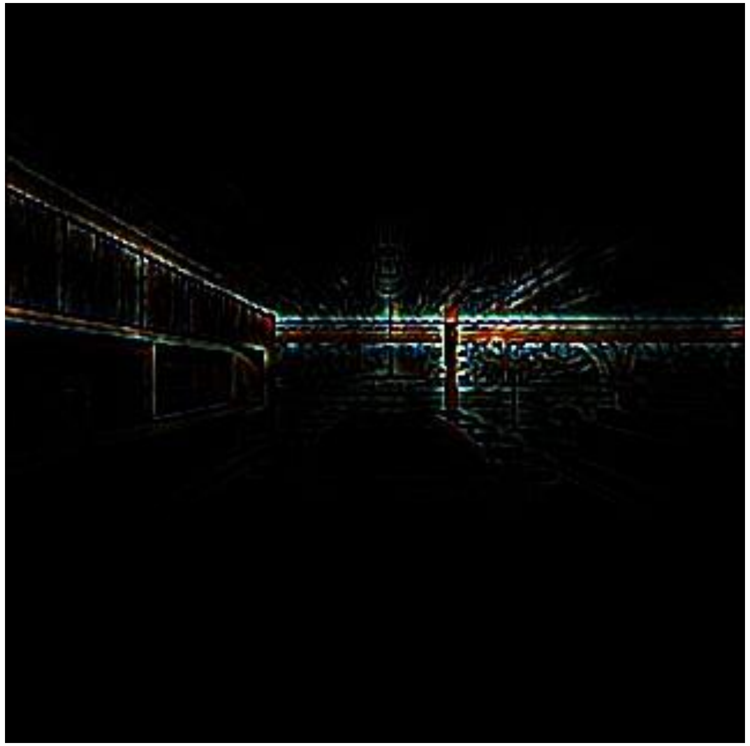


Model 1: TP-LSD with only TP extraction branch

The development of TP-LSD

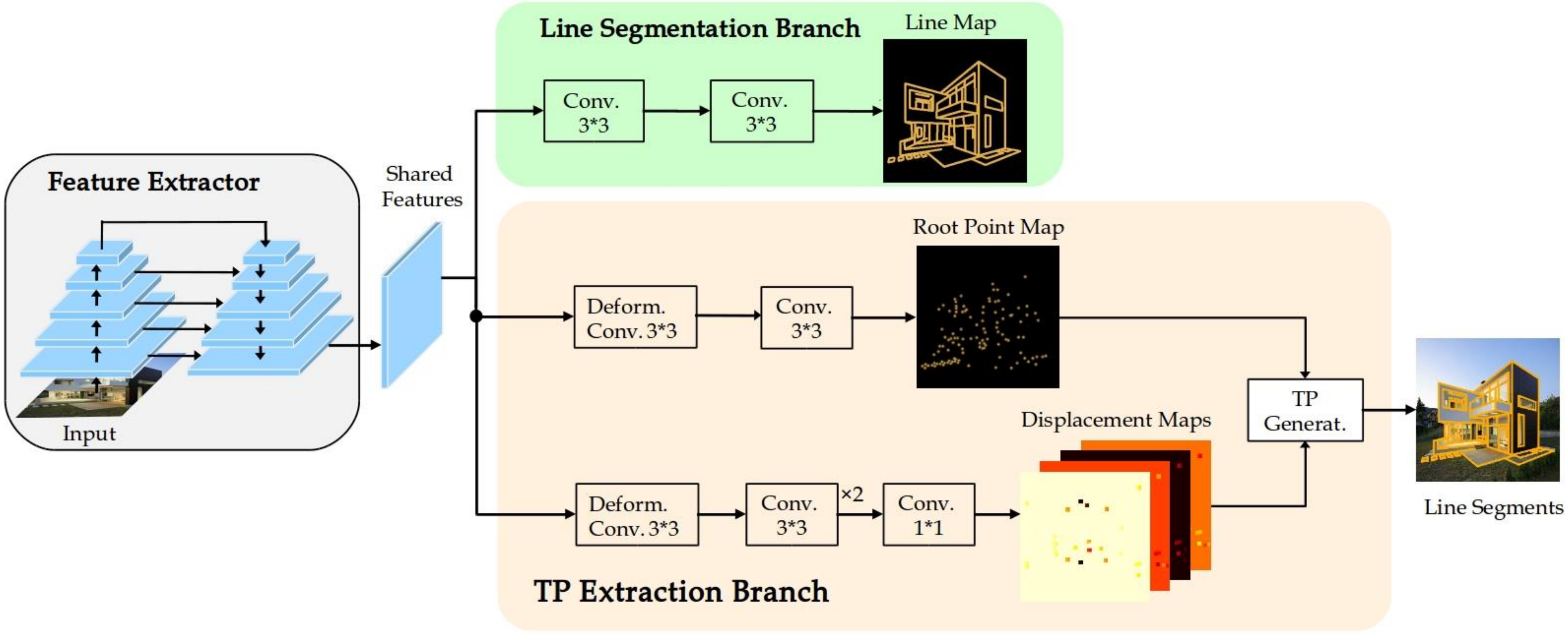


CNN visualization



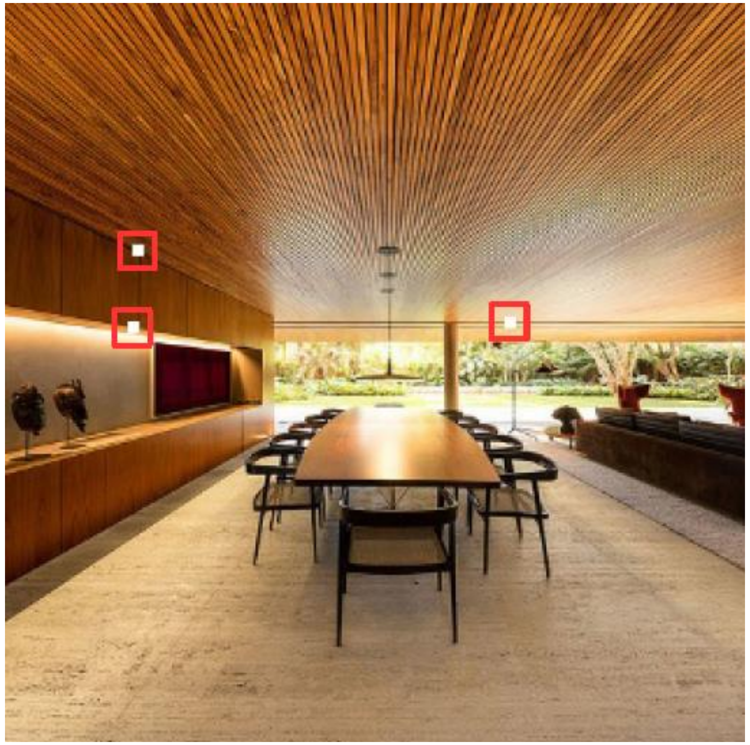
Model 1: TP-LSD with only TP extraction branch

The development of TP-LSD

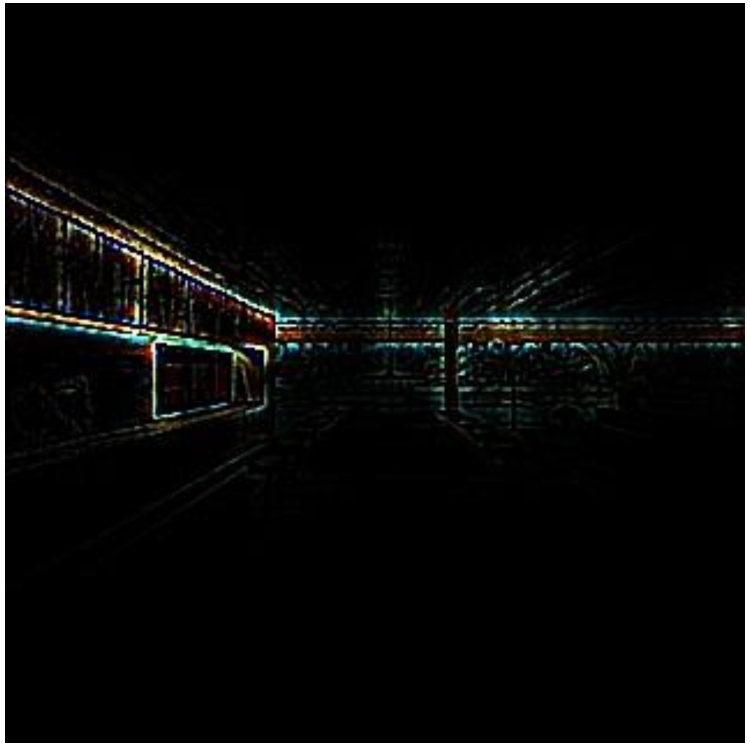


Model 2: TP-LSD with TP extraction branch and line segmentation branch

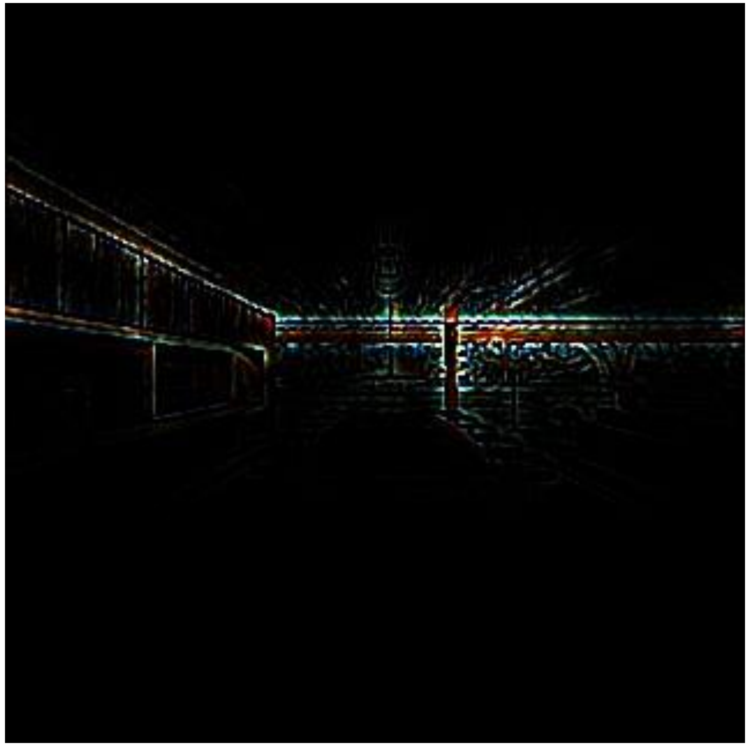
The development of TP-LSD



CNN visualization



CNN visualization
(without line segmentation branch)

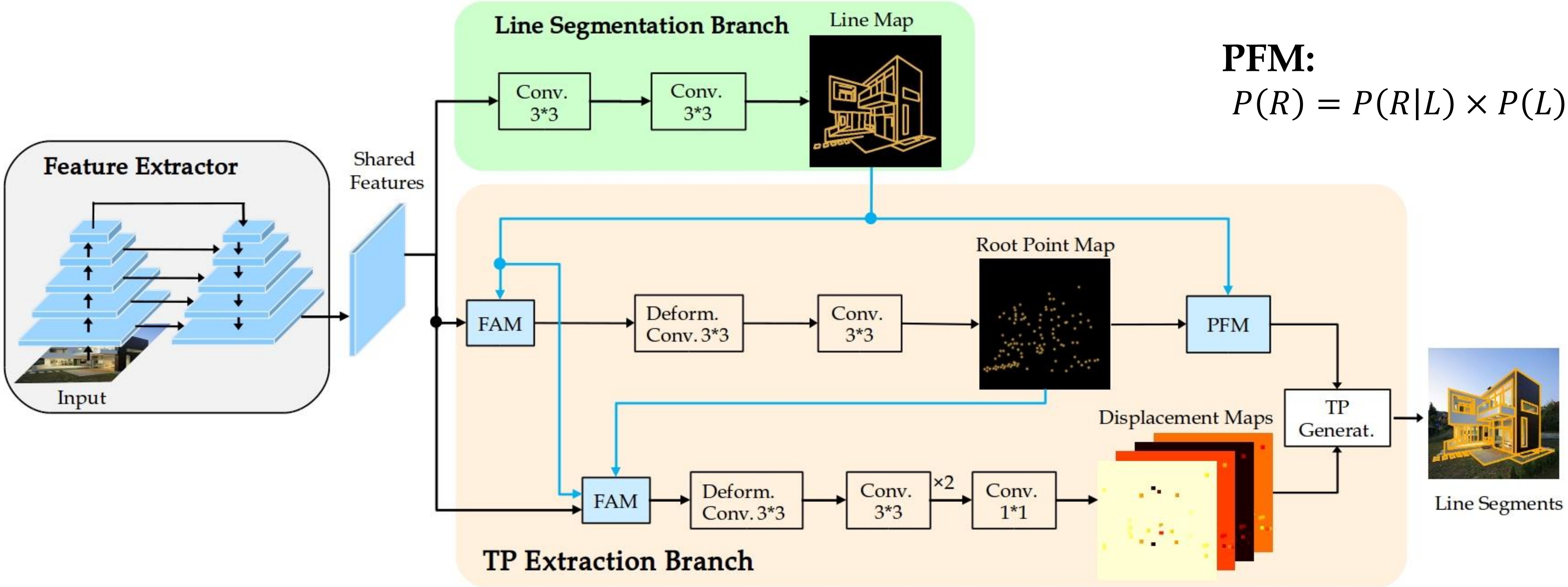


Model 2: TP-LSD with TP extraction branch and line segmentation branch

The development of TP-LSD

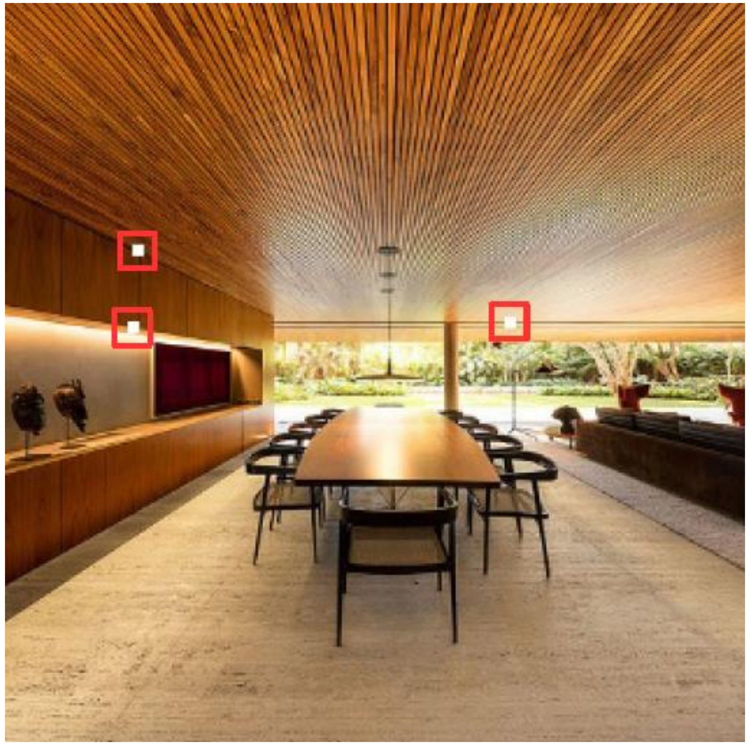
PFM:

$$P(R) = P(R|L) \times P(L)^{\alpha}$$

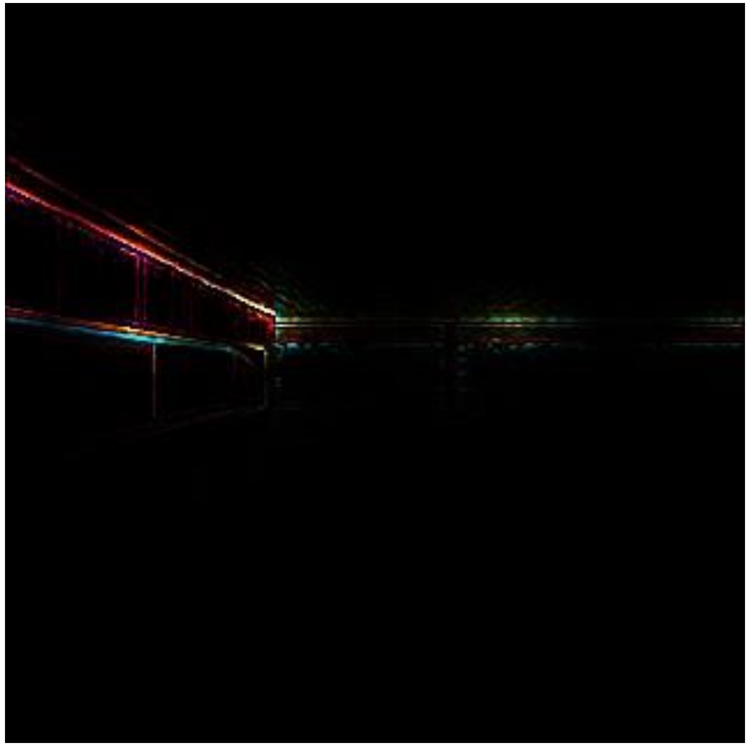


Model 3: TP-LSD with feature aggregating module (FAM) and point filter module (PFM)

The development of TP-LSD

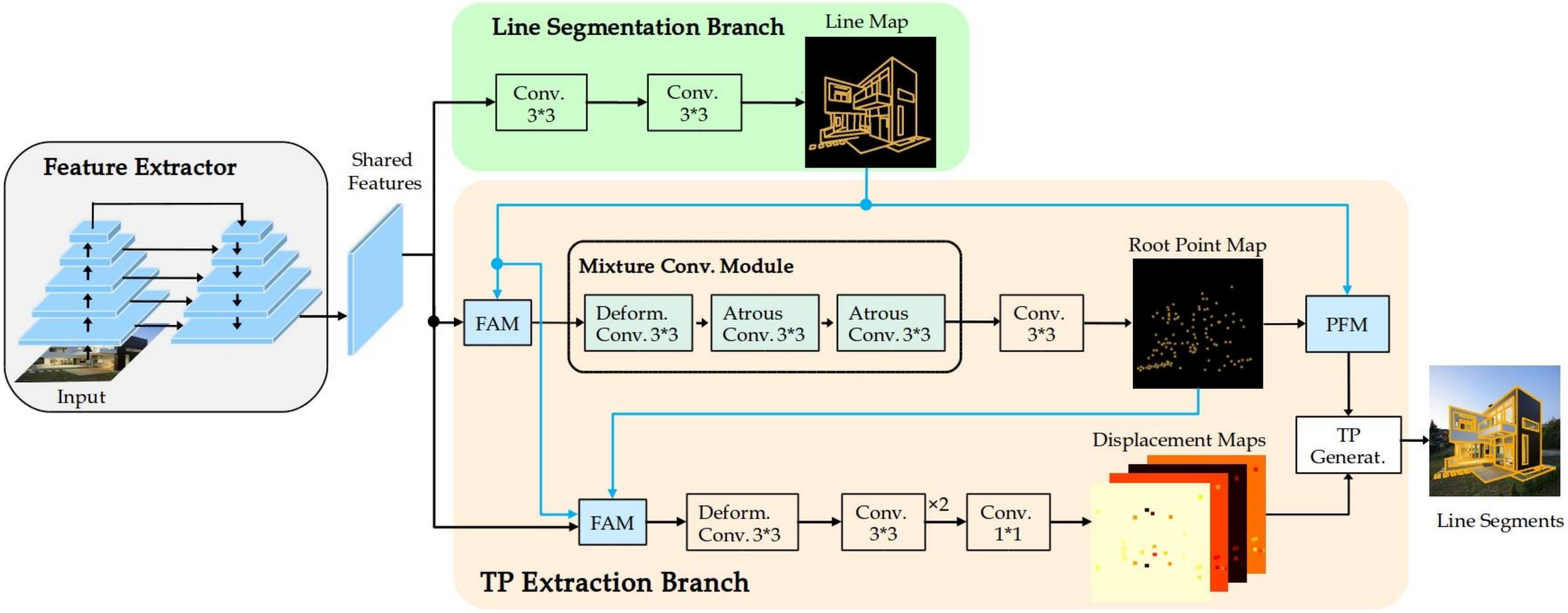


CNN visualization



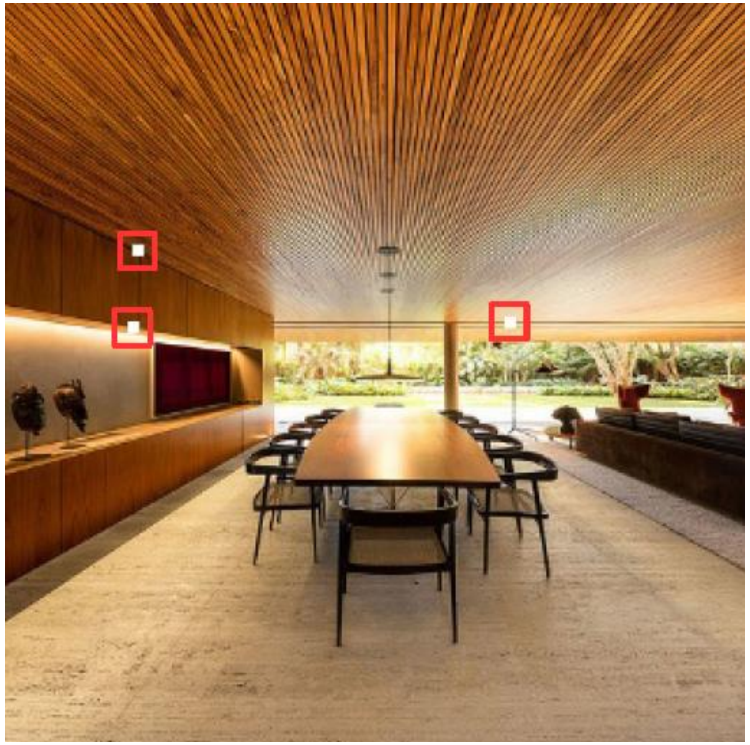
Model 3: TP-LSD with feature aggregating module (FAM) and point filter module (PFM)

The development of TP-LSD

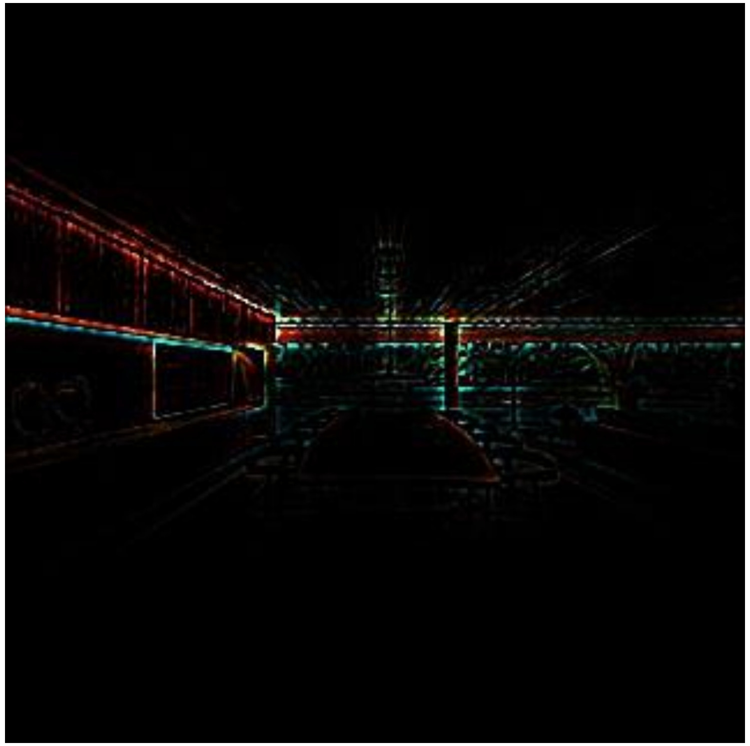


Model 4: TP-LSD with mixture convolution module (MCM)

The development of TP-LSD

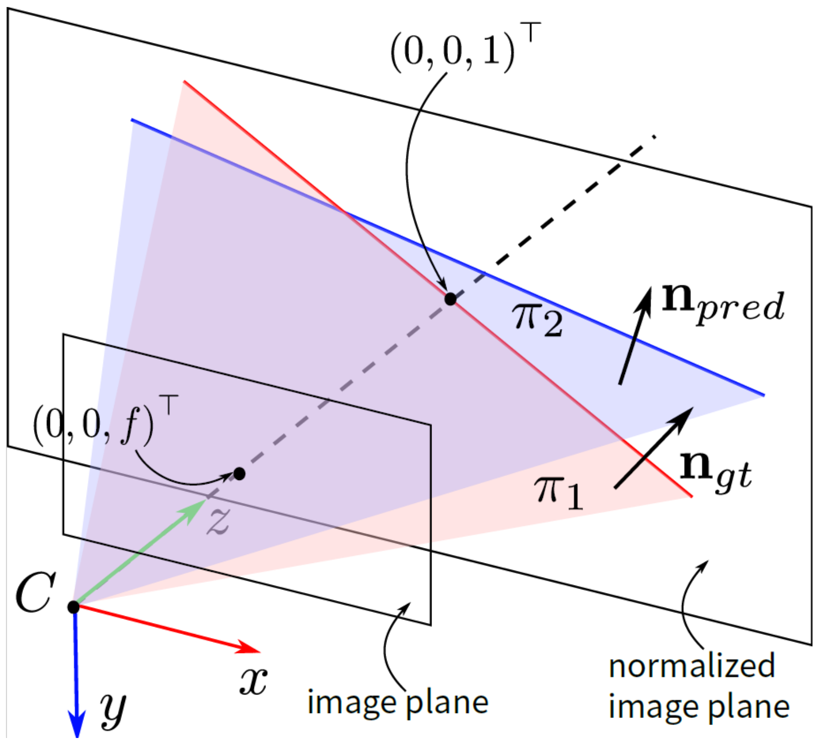


CNN visualization



Model 4: TP-LSD with mixture convolution module (MCM)

Overview of Line Matching Score



$$LMS = Score_\theta \times Score_l$$

$$Score_\theta = \begin{cases} 1 - \frac{\theta(\mathbf{n}_{gt}, \mathbf{n}_{pred})}{\eta_\theta}, & \text{if } \theta(\mathbf{n}_{gt}, \mathbf{n}_{pred}) < \eta_\theta \\ 0, & \text{otherwise} \end{cases}$$

$$\eta_1 = \frac{\mathcal{L}_{pred} \cap \mathcal{L}_{gt}}{\mathcal{L}_{gt}}, \quad \eta_2 = \frac{\mathcal{L}_{pred} \cap \mathcal{L}_{gt}}{\mathcal{L}_{pred}\,|\cos(\alpha)|}$$

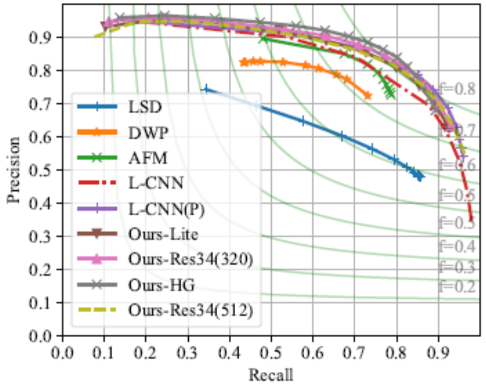
$$Score_l = \begin{cases} \frac{\eta_1 + \eta_2}{2}, & \text{if } \eta_1 \geq \eta_l, \text{ and } \eta_2 \geq \eta_l \\ 0, & \text{otherwise} \end{cases}$$

Analysis of the metric on real image

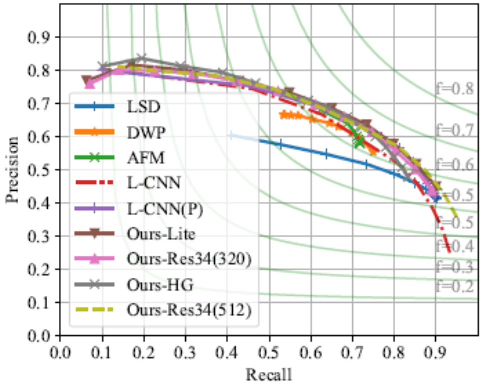
(a) GT, GT-lines marked by red (b) sAP^{10} matching result (c) LAP matching result

In (b, c), the mismatched and matched line segments are marked by blue and red, respectively.

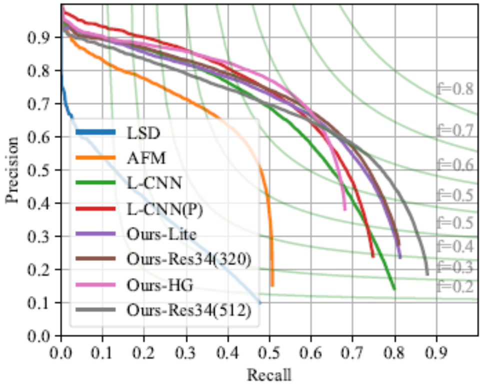
Evaluation on Public Benchmarks



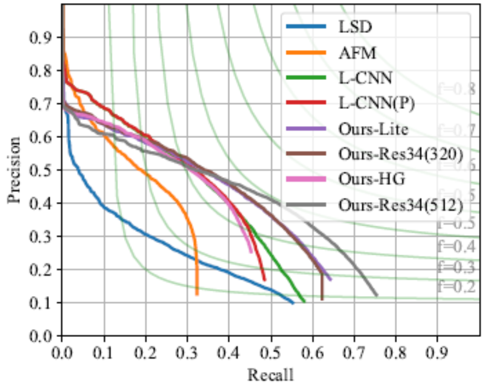
(a) Pixel based metric; Wireframe



(b) Pixel based metric; YorkUrban



(c) LMS metric; Wireframe



(d) LMS metric; YorkUrban

Precision-recall curves of line segment detection
The models are trained on Wireframe dataset and tested on both Wireframe and YorkUrban datasets

[1] Huang K, Wang Y, Zhou Z, et al. Learning to parse wireframes in images of man-made environments[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 626-635.

[2] Denis P, Elder J H, Estrada F J. Efficient edge-based methods for estimating manhattan frames in urban imagery[C]//European conference on computer vision. Springer, Berlin, Heidelberg, 2008: 197-210.

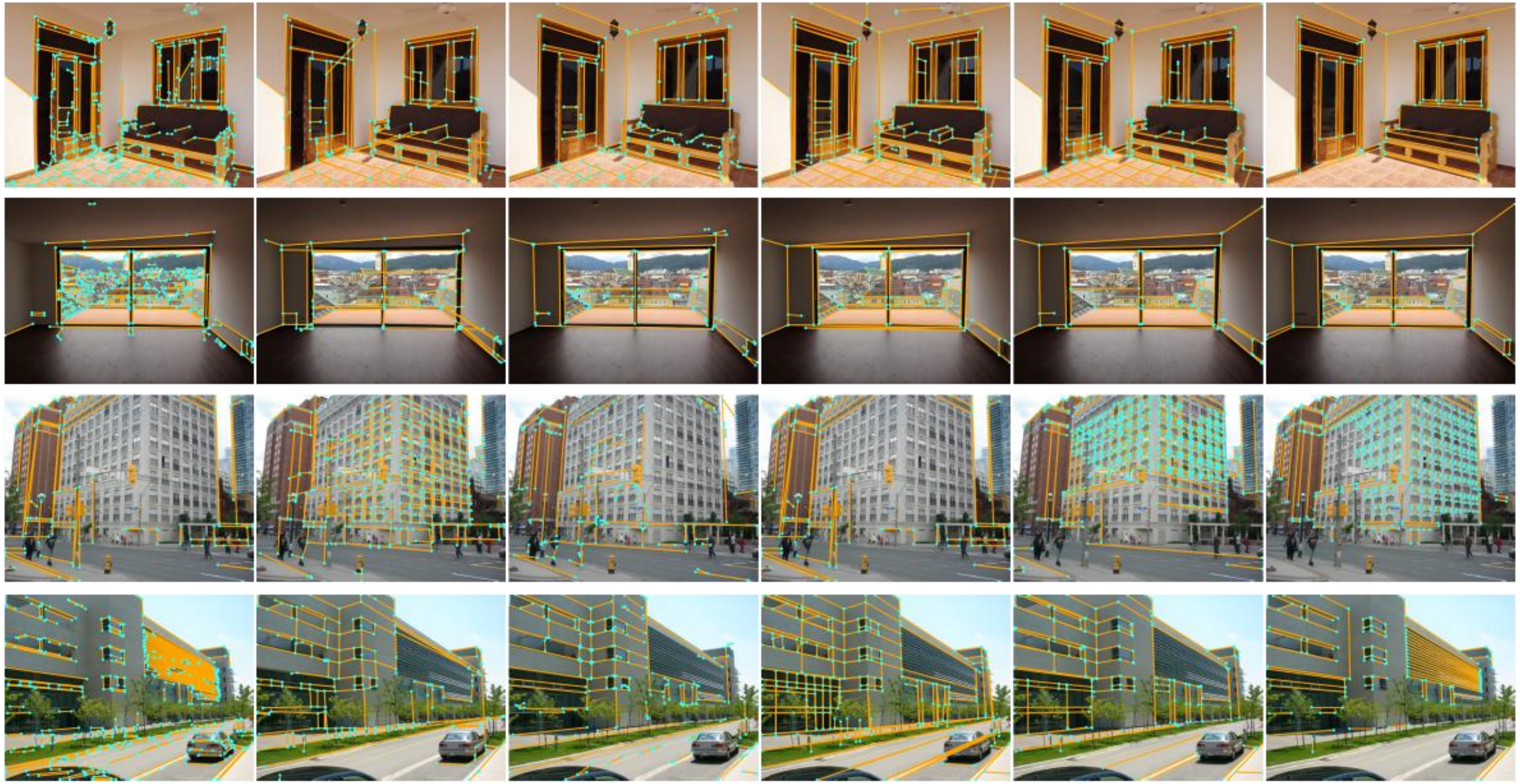
Evaluation on Public Benchmarks

Table 1: Evaluation results of different line segment detection methods. "/“ means that the score is too slow to be meaningful. The best two scores are shown in red and blue.

Method	Input Size	Wireframe dataset				YorkUrban dataset				FPS
		F^H	sAP^5	sAP^{10}	LAP	F^H	sAP^5	sAP^{10}	LAP	
LSD [6]	320	0.641	6.7	8.8	18.7	0.606	7.5	9.2	16.1	100
DWP [8]	512	0.727	/	/	6.6	0.652	/	/	3.1	2.2
AFM [22]	320	0.773	18.3	23.9	36.7	0.663	7.0	9.1	17.5	12.8
L-CNN [28]	512	0.775	58.9	62.8	59.8	0.646	25.9	28.2	32.0	11.1
L-CNN(P) [28]	512	0.817	52.4	57.3	57.9	0.675	20.9	23.1	26.8	5.2
TP-LSD-Lite	320	0.804	56.4	59.7	59.7	0.681	24.8	26.8	31.2	78.2
TP-LSD-Res34	320	0.816	57.5	60.6	60.6	0.674	25.3	27.4	31.1	42.2
TP-LSD-HG	512	0.820	50.9	57.0	55.1	0.673	18.9	22.0	24.6	53.4
TP-LSD-Res34	512	0.806	57.6	57.2	61.3	0.672	27.6	27.7	34.3	18.1

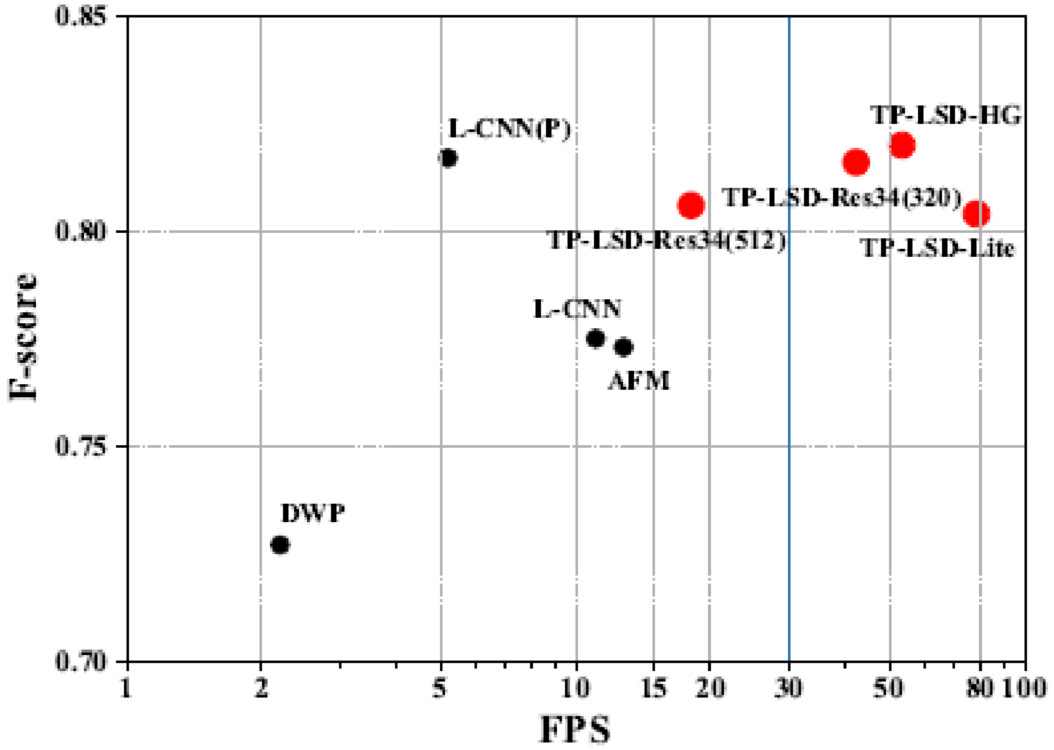
[1] Huang K, Wang Y, Zhou Z, et al. Learning to parse wireframes in images of man-made environments[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 626-635.
[2] Denis P, Elder J H, Estrada F J. Efficient edge-based methods for estimating Manhattan frames in urban imagery[C]//European conference on computer vision. Springer, Berlin, Heidelberg, 2008: 197-210.

Evaluation on Public Benchmarks



LSD DWP AFM L-CNN TP-LSD GT

Conclusion



- **Tri-points representation**
  - Proposed to simplify the detection pipeline
- **TP-LSD**
  - A faster and compact model for line segment detection
- **LAP (Line Matching Average Precision)**
  - A more distinctively evaluation metric for the relative spatial relationship between line segments
- **Performance**
  - SOTA result on the Wireframe and the YorkUrban dataset
  - Improve the inference speed by 6 times, up to 78FPS

Thanks