Learning Attraction Field Representation for Robust Line Segment Detection

Nan Xue[1,2], Song Bai[3], Fudong Wang[1], Gui-Song Xia[1]*, Tianfu Wu[2], Liangpei Zhang[1]

[1]Wuhan University, China   [2]NC State University, USA   [3]University of Oxford, UK

{xuenan, fudong-wang, guisong.xia, zlp62}@whu.edu.cn,
songbai.site@gmail.com, tianfu_wu@ncsu.edu

Abstract

This paper presents a region-partition based attraction field dual representation for line segment maps, and thus poses the problem of line segment detection (LSD) as the region coloring problem. The latter is then addressed by learning deep convolutional neural networks (ConvNets) for accuracy, robustness and efficiency. For a 2D line segment map, our dual representation consists of three components: (i) A region-partition map in which every pixel is assigned to one and only one line segment; (ii) An attraction field map in which every pixel in a partition region is encoded by its 2D projection vector w.r.t. the associated line segment; and (iii) A squeeze module which squashes the attraction field to a line segment map that almost perfectly recovers the input one. By leveraging the duality, we learn ConvNets to compute the attraction field maps for raw input images, followed by the squeeze module for LSD, in an end-to-end manner. Our method rigorously addresses several challenges in LSD such as local ambiguity and class imbalance. Our method also harnesses the best practices developed in ConvNets based semantic segmentation methods such as the encoder-decoder architecture and the a-trous convolution. In experiments, our method is tested on the WireFrame dataset [12] and the YorkUrban dataset [6] with state-of-the-art performance obtained. Especially, we advance the performance by 4.5 *percents on the WireFrame dataset. Our method is also fast with* 6.6 ∼ 10.4 *FPS, outperforming most of the existing line segment detectors. The source code of this paper is available at* https://github.com/cherubicXN/afm_cvpr2019.

1. Introduction

1.1. Motivation and Objective

Line segment detection (LSD) is an important yet challenging low-level task in computer vision. The resulting

---

*Corresponding author

(a) Attraction field map representation for line segments

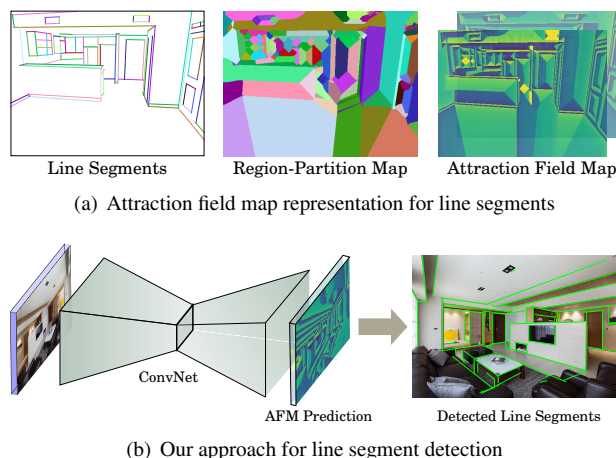(b) Our approach for line segment detection

Figure 1. Illustration of the proposed method. (a) The proposed attraction field dual representation for line segment maps. A line segment map can be almost perfectly recovered from its attraction filed map (AFM), by using a simple squeeze algorithm. (b) The proposed formulation of posing the LSD problem as the region coloring problem. The latter is addressed by learning ConvNets.

line segment maps provide compact structural information that facilitate many up-level vision tasks such as 3D reconstruction [6, 8], image partition [7], stereo matching [32], scene parsing [34, 33], camera pose estimation [27], and image stitching [25].

LSD usually consists of two steps: line heat map generation and line segment model fitting. The former can be computed either simply by the gradient magnitude map (mainly used before the recent resurgence of deep learning) [23, 31, 5], or by a learned convolutional neural network (ConvNet) [26, 18] in state-of-the-art methods [12]. The latter needs to address the challenging issue of handling unknown multi-scale discretization nuisance factors (*e.g.,* the classic zig-zag artifacts of line segments in digital images) when aligning pixels or linelets to form line segments in the line heat map. Different schema have been proposed, *e.g.,* the ϵ-meaningful alignment method proposed in [23] and the junction [24] guided alignment method proposed

in [12]. The main drawbacks of existing two-stage methods are in two-fold: lacking elegant solutions to solve the local ambiguity and/or class imbalance in line heat map generation, and requiring extra carefully designed heuristics or supervisedly learned contextual information in inferring line segments in the line heat map.

In this paper, we focus on learning based LSD framework and propose a single-stage method which rigorously addresses the drawbacks of existing LSD approaches. Our method is motivated by two observations,

- The duality between region representation and boundary contour representation of objects or surfaces, which is a well-known fact in computer vision.

- The recent remarkable progresses for image semantic segmentation by deep ConvNet based methods such as U-Net [22] and DeepLab V3+ [11].

So, the intuitive idea of this paper is that if we can bridge line segment maps and their dual region representations, we will pose the problem of LSD as the problem of region coloring, and thus open the door to leveraging the best practices developed in state-of-the-art deep ConvNet based image semantic segmentation methods to improve performance for LSD. By dual region representations, it means they are capable of recovering the input line segment maps in a nearly perfect way via a simple algorithm. We present an efficient and straightforward method for computing the dual region representation. By re-formulating LSD as the equivalent region coloring problem, we address the aforementioned challenges of handling local ambiguity and class imbalance in a principled way.

1.2. Method Overview

Figure 1 illustrates the proposed method. Given a 2D line segment map, we represent each line segment by its geometry model using the two end-points[1]. In computing the dual region representation, there are three components (detailed in Section 3).

- A region-partition map. It is computed by assigning every pixel to one and only one line segment based on a proposed point to line segmentation distance function. The pixels associated with one line segment form a region. All regions represent a partition of the image lattice (*i.e.,* mutually exclusive and the union occupies the entire image lattice).

- An attraction field map. Each pixel in a partition region has one and only one corresponding projection point on the geometry line segment (but the reverse is often

a one-to-many mapping). In the attraction field map, every pixel in a partition region is then represented by its attraction/projection vector between the pixel and its projection point on the geometry line segment [2].

- A light-weight squeeze module. It follows the attraction field to squash partition regions in an attraction field map to line segments that almost perfectly recovers the input ones, thus bridging the duality between region-partition based attraction field maps and line segment maps.

The proposed method can also be viewed as an intuitive expansion-and-contraction operation between 1D line segments and 2D regions in a simple projection vector field: The region-partition map generation jointly expands all line segments into partition regions, and the squeeze module degenerates regions into line segments.

With the duality between a line segment map and the corresponding region-partition based attraction field map, we first convert all line segment maps in the training dataset to their attraction field maps. Then, we learn ConvNets to predict the attraction field maps from raw input images in an end-to-end way. We utilize U-Net [22] and a modified network based on DeepLab V3+ [11] in our experiments. After the attraction field map is computed, we use the squeeze module to compute its line segment map.

In experiments, the proposed method is tested on the WireFrame dataset [12] and the YorkUrban dataset [6] with state-of-the-art performance obtained comparing with [12, 5, 1, 23]. In particular, we improve the performance by 4.5% on the WireFrame dataset. Our method is also fast with $6.6 \sim 10.4$ FPS, outperforming most of line segment detectors.

2. Related Work and Our Contributions

The study of line segment detection has a very long history since 1980s [2]. The early pioneers tried to detect line segments based upon the edge map estimation. Then, the perception grouping approaches based on the *Gestalt Theory* are proposed. Both of these methods concentrate on the hand-crafted low-level features for the detection, which have become a limitation. Recently, the line segment detection and its related problem edge detection have been studied under the perspective of deep learning, which dramatically improved the detection performance and brings us of great practical importance for real applications.

2.1. Detection based on Hand-crafted Features

In a long range of time, the hand-crafted low-level features (especially for image gradients) are heavily used for

---

[1]We will have discrepancy for some intermediate points of a line segment between their annotated pixel locations and the geometric locations when the line segment is not strictly horizontal or vertical.

[2]They are the same point when the pixel is on the geometry line segment, and thus we will have a zero vector. We observed that the total number of those points are negligible in our experiments.

line segment detection. These approaches can be divided into edge map based approaches [9, 14, 28, 29, 30, 1] and perception grouping approaches [3, 23, 5]. The edge map based approaches treat the visual features as a discriminated feature for edge map estimation and subsequently applying the Hough transform [2] to globally search line configurations and then cutting them by using thresholds. In contrast to the edge map based approaches, the grouping methods directly use the image gradients as local geometry cues to group pixels into line segment candidates and filter out the false positives [23, 5].

Actually, the features used for line segment detection can only characterize the local response from the image appearance. For the edge detection, only local response without global context cannot avoid false detection. On the other hand, both the magnitude and orientation of image gradients are easily affected by the external imaging condition (*e.g.* noise and illumination). Therefore, the local nature of these features limits us to extract line segments from images robustly. In this paper, we break the limitation of locally estimated features and turn to learn the deep features that hierarchically represent the information of images from low-level cues to high-level semantics.

2.2. Deep Edge and Line Segment Detection

Recently, HED [26] opens up a new era for edge perception from images by using ConvNets. The learned multi-scale and multi-level features dramatically addressed the problem of false detection in the edge-like texture regions and approaching human-level performance on the BSDS500 dataset [20]. Followed by this breakthrough, a tremendous number of deep learning based edge detection approaches are proposed [18, 15, 17, 16, 19, 11]. Under the perspective of binary classification, the edge detection has been solved to some extent. It is natural to upgrade the traditional edge map based line segment detection by alternatively using the edge map estimated by ConvNets. However, the edge maps estimated by ConvNets are usually over-smoothed, which will lead to local ambiguities for accurate localization. Further, the edge maps do not contain enough geometric information for the detection. According to the development of deep learning, it should be more reasonable to propose an end-to-end line segment detector instead of only applying the advances of deep edge detection.

Most recently, Huang *et al.* [12] have taken an important step towards this goal by proposing a large-scale dataset with high quality *line segment annotations* and approaching the problem of line segment detection as two parallel tasks, *i.e.*, edge map detection and junction detection. As a final step for the detection, the resulted edge map and junctions are fused to produce line segments. To the best of our knowledge, this is the first attempt to develop a deep learning based line segment detector. However, due to the sophisticated relation between edge map and junctions, it still remains a problem unsolved. Benefiting from our proposed formulation, we can directly learn the line segments from the attraction field maps that can be easily obtained from the line segment annotations without the junction cues.

Our Contributions The proposed method makes the following main contributions to robust line segment detection.
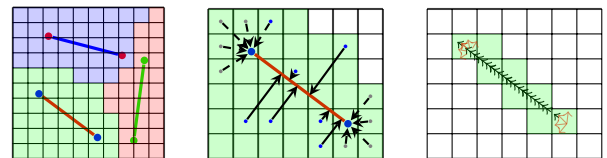
- A novel dual representation is proposed by bridging line segment maps and region-partition-based attraction field maps. To our knowledge, it is the first work that utilizes this simple yet effective representation in LSD.

- With the proposed dual representation, the LSD problem is re-formulated as the region coloring problem, thus opening the door to leveraging state-of-the-art semantic segmentation methods in addressing the challenges of local ambiguity and class imbalance in existing LSD approaches in a principled way.

- The proposed method obtains state-of-the-art performance on two widely used LSD benchmarks, the Wire-Frame dataset (with 4.5% significant improvement) and the YorkUrban dataset.

3. The Attraction Field Representation

In this section, we present details of the proposed region-partition representation for LSD.

3.1. The Region-Partition Map

Let Λ be an image lattice (*e.g.*, 800×600). A line segment is denote by $l_i = (\mathbf{x}_i^s, \mathbf{x}_i^e)$ with the two end-points being $\mathbf{x}_i^s$ and $\mathbf{x}_i^e$ (non-negative real-valued positions due to sub-pixel precision is used in annotating line segments) respectively. The set of line segments in a 2D line segment map is denoted by $L = \{l_1, \cdots, l_n\}$. For simplicity, we also denote the line segment map by L. Figure 2 illustrates a line segment map with 3 line segments in a 10×10 image lattice.

(a) Support regions   (b) Attraction vectors   (c) Squeeze module

Figure 2. A toy example illustrating a line segment map with 3 line segments, its dual region-partition map, selected vectors of the attraction field map and the squeeze module for obtaining line segments from the attraction field map. See text for details.

Computing the region-partition map for L is assigning every pixel in the lattice to one and only one of the n line segments. To that end, we utilize the point-to-line-segment distance function. Consider a pixel $p \in \Lambda$ and a line segment $l_i = (\mathbf{x}_i^s, \mathbf{x}_i^e) \in L$, we first project the pixel p to the straight line going through l_i in the continuous geometry space. If the projection point is not on the line segment, we use the closest end-point of the line segment as the projection point. Then, we compute the Euclidean distance between the pixel and the projection point. Formally, we define the distance between p and l_i by

$$d(p, l_i) = \min_{t \in [0,1]} ||\mathbf{x}_i^s + t \cdot (\mathbf{x}_i^e - \mathbf{x}_i^s) - p||_2^2,$$
$$t_p^* = \arg\min_t d(p, l_i), \tag{1}$$

where the projection point is the original point-to-line projection point if $t_p^* \in (0, 1)$, and the closest end-point if $t_p^* = 0$ or 1.

So, the region in the image lattice for a line segment l_i is defined by

$$R_i = \{p \,|\, p \in \Lambda; d(p, l_i) < d(p, l_j), \forall j \neq i, l_j \in L\}. \tag{2}$$

It is straightforward to see that $R_i \cap R_j = \emptyset$ and $\cup_{i=1}^n R_i = \Lambda$, *i.e.*, all R_i's form a partition of the image lattice. Figure 2(a) illustrates the partition region generation for a line segment in the toy example (Figure 2). Denote by $R = \{R_1, \cdots, R_n\}$ the *region-partition map* for a line segment map L.

3.2. Computing the Attraction Field Map

Consider the partition region R_i associated with a line segment l_i, for each pixel $p \in R_i$, its projection point p' on l_i is defined by

$$p' = \mathbf{x}_i^s + t_p^* \cdot (\mathbf{x}_i^e - \mathbf{x}_i^s), \tag{3}$$

We define the 2D attraction or projection vector for a pixel p as,

$$\mathbf{a}(p) = p' - p, \tag{4}$$

where the attraction vector is perpendicular to the line segment if $t_p^* \in (0, 1)$ (see Figure 2(b)). Figure 1 shows examples of the x- and y-component of an *attraction field map* (AFM). Denote by $A = \{\mathbf{a}(p) \,|\, p \in \Lambda\}$ the attraction field map for a line segment map L.

3.3. The Squeeze Module

Given an attraction field map A, we first reverse it by computing the real-valued projection point for each pixel p in the lattice,

$$v(p) = p + \mathbf{a}(p), \tag{5}$$

and its corresponding discretized point in the image lattice,

$$v_\Lambda(p) = \lfloor v(p) + 0.5 \rfloor. \tag{6}$$

where $\lfloor \cdot \rfloor$ represents the floor operation, and $v_\Lambda(p) \in \Lambda$.

Then, we compute a line proposal map in which each pixel $q \in \Lambda$ collects the attraction field vectors whose discretized projection points are q. The candidate set of attraction field vectors collected by a pixel q is then defined by

$$\mathcal{C}(q) = \{\mathbf{a}(p) \,|\, p \in \Lambda, v_\Lambda(p) = q\}, \tag{7}$$

where $\mathcal{C}(q)$'s are usually non-empty for a sparse set of pixels q's which correspond to points on the line segments. An example of the line proposal map is shown in Figure 2(c), which project the pixels of the support region for a line segment into pixels near the line segment.

With the line proposal map, our squeeze module utilizes an iterative and greedy grouping algorithm to fit line segments, similar in spirit to the region growing algorithm used in [23].

- Given the current set of active pixels each of which has a non-empty candidate set of attraction field vectors, we randomly select a pixel q and one of its attraction field vector $\mathbf{a}(p) \in \mathcal{C}(q)$. The tangent direction of the selected attraction field vector $\mathbf{a}(p)$ is used as the initial direction of the line segment passing the pixel q.

- Then, we search the local observation window centered at q (*e.g.,* a 3×3 window is used in this paper) to find the attraction field vectors which are aligned with $\mathbf{a}(p)$ with angular distance less than a threshold τ (*e.g.,* $\tau = 10°$ used in this paper).

  - If the search fails, we discard $\mathbf{a}(p)$ from $\mathcal{C}(q)$, and further discard the pixel q if $\mathcal{C}(q)$ becomes empty.

  - Otherwise, we grow q into a set and update its direction by averaging the aligned attraction vectors. The aligned attraction vectors will be marked as used (and thus inactive for the next round search). For the two end-points of the set, we recursively apply the greedy search algorithm to grow the line segment.

- Once terminated, we obtain a candidate line segment $l_q = (\mathbf{x}_q^s, \mathbf{x}_q^e)$ with the support set of real-valued projection points. We fit the minimum outer rectangle using the support set. We verify the candidate line segment by checking the aspect ratio between width and length of the approximated rectangle with respect to a predefined threshold to ensure the approximated rectangle is "thin enough". If the checking fails, we mark the pixel q inactive and release the support set to be active again.
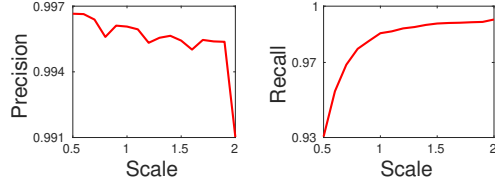
Figure 3. Verification of the duality between line segment maps and attraction field maps, and its scale invariance.

3.4. Verifying the Duality and its Scale Invariance

We test the proposed attraction field representation on the WireFrame dataset [12]. We first compute the attraction field map for each annotated line segment map and then compute the estimated line segment map using the squeeze module. We run the test across multiple scales, ranging from 0.5 to 2.0 with step-size 0.1. We evaluate the estimated line segment maps by measuring the precision and recall following the protocol provided in the dataset. Figure 3 shows the precision-recall curves. The average precision and recall rates are above 0.99 and 0.93 respectively, thus verifying the duality between line segment maps and corresponding region-partition based attractive field maps, as well as the scale invariance of the duality.

So, **the problem of LSD can be posed as the region coloring problem almost without hurting the performance.** In the region coloring formulation, our goal is to learn ConvNets to infer the attraction field maps for input images. The attraction field representation eliminates local ambiguity in traditional gradient magnitude based line heat map, and the predicting attraction field in learning gets rid of the imbalance problem in line *v.s.* non-line classification.

4. Robust Line Segment Detector

In this section, we present details of learning ConvNets for robust LSD. ConvNets are used to predict AFMs from raw input images under the image-to-image transformation framework, and thus we adopt encoder-decoder network architectures.

4.1. Data Processing

Denote by $D = \{(I_i, L_i); i = 1, \cdots, N\}$ the provided training dataset consisting of N pairs of raw images and annotated line segment maps. We first compute the AFMs for each training image. Then, let $D = \{(I_i, \mathbf{a}_i); i = 1, \cdots, N\}$ be the dual training dataset. To make the AFMs insensitive to the sizes of raw images, we adopt a simple normalization scheme. For an AFM $\mathbf{a}$ with the spatial dimensions being $W \times H$, the size-normalization is done by

$$\mathbf{a}_x := \mathbf{a}_x/W, \ \ \mathbf{a}_y := \mathbf{a}_y/H, \quad (8)$$

where $\mathbf{a}_x$ and $\mathbf{a}_y$ are the component of $\mathbf{a}$ along x and y axes respectively. However, the size-normalization will make the

values in $\mathbf{a}$ small and thus numerically unstable in training. We apply a point-wise invertible value stretching transformation for the size-normalized AFM

$$z' := S(z) = -\text{sign}(z) \cdot \log(|z| + \varepsilon), \quad (9)$$

where $\varepsilon = 1\text{e}{-}6$ to avoid $\log(0)$. The inverse function $S^{-1}(\cdot)$ is defined by

$$z := S^{-1}(z') = \text{sign}(z')e^{(-|z'|)}. \quad (10)$$

For notation simplicity, denote by $R(\cdot)$ the composite reverse function, and we still denote by $D = \{(I_i, \mathbf{a}_i); i = 1, \cdots, N\}$ the final training dataset.

4.2. Inference

Denote by $f_\Theta(\cdot)$ a ConvNet with the parameters collected by Θ. As illustrated in Figure 1(b), for an input image I_Λ, our robust LSD is defined by

$$\hat{\mathbf{a}} = f_\Theta(I_\Lambda) \quad (11)$$

$$\hat{L} = Squeeze(R(\hat{\mathbf{a}})) \quad (12)$$

where $\hat{\mathbf{a}}$ is the predicted AFM for the input image (the size-normalized and value-stretched one), $Squeeze(\cdot)$ the squeeze module and $\hat{L}$ the inferred line segment map.

4.3. Network Architectures

We utilize two network architectures to realize $f_\Theta()$: one is U-Net [22], and the other is a modified U-Net, called *a-trous Residual U-Net* which uses the ASSP module proposed in DeepLab v3+ [4] and the skip-connection as done in ResNet [10].

Table 1 shows the configurations of the two architectures. The network consists of 5 encoder and 4 decoder stages indexed by $c1, \ldots, c5$ and $d1, \ldots, d4$ respectively.

- For U-Net, the *double conv* operator that contains two convolution layers is applied and denoted as $\{\cdot\}$. The $\{\cdot\}*$ operator of d_i stage upscales the output feature map of its last stage and then concatenate it with the feature map of c_i stage together before applying the *double conv* operator.

- For the *a-trous Residual U-Net*, we replace the *double conv* operator to the *Residual block*, denoted as $[\cdot]$. Different from the ResNet, we use the plain convolution layer with 3×3 kernel size and stride 1. Similar to $\{\cdot\}*$, the operator $[\cdot]*$ also takes the input from two sources and upscales the feature of first input source. The first layer of $[\cdot]*$ contains two parallel convolution operators to reduce the depth of feature maps and then concatenate them together for the subsequent calculations. In the stage d_4, we apply the 4 ASPP operators

Table 1. Network architectures we investigated for the *attraction field learning*. {} and [] represent the double conv in U-Net and the residual block. Inside the brackets are the shape of convolution kernels. The suffix $*$ represent the bilinear upsampling operator with the scaling factor 2. The number outside the brackets is the number of stacked blocks on a stage.

stage	U-Net	a-trous Residual U-Net	
c1	$\left\{\begin{array}{l}3\times3,64\\3\times3,64\end{array}\right\}$	$3\times3,64$, stride 1	
	2×2 max pool, stride 2	3×3 max pool, stride 2	
c2	$\left\{\begin{array}{l}3\times3,128\\3\times3,128\end{array}\right\}$	$\left[\begin{array}{l}1\times1,64\\3\times3,64\\1\times1,256\end{array}\right]$	$\times3$
	2×2 max pool, stride 2	$1\times1,128$	
c3	$\left\{\begin{array}{l}3\times3,256\\3\times3,256\end{array}\right\}$	$\left[\begin{array}{l}1\times1,128\\3\times3,128\\1\times1,512\end{array}\right]$	$\times4$
	2×2 max pool, stride 2	$1\times1,256$	
c4	$\left\{\begin{array}{l}3\times3,512\\3\times3,512\end{array}\right\}$	$\left[\begin{array}{l}1\times1,256\\3\times3,256\\1\times1,1024\end{array}\right]$	$\times6$
	2×2 max pool, stride 2	$1\times1,512$	
c5	$\left\{\begin{array}{l}3\times3,512\\3\times3,512\end{array}\right\}$	$\left[\begin{array}{l}1\times1,512\\3\times3,512\\1\times1,2048\end{array}\right]$	$\times3$
		ASPP	
d4	$\left\{\begin{array}{l}3\times3,256\\3\times3,256\end{array}\right\}*$	$\left[\begin{array}{l}1\times1,256;1\times1,256\\3\times3,512\\1\times1,512\end{array}\right]$	$*$
d3	$\left\{\begin{array}{l}3\times3,128\\3\times3,128\end{array}\right\}*$	$\left[\begin{array}{l}1\times1,128;1\times1,128\\3\times3,256\\1\times1,256\end{array}\right]$	$*$
d2	$\left\{\begin{array}{l}3\times3,64\\3\times3,64\end{array}\right\}*$	$\left[\begin{array}{l}1\times1,64;1\times1,64\\3\times3,128\\1\times1,128\end{array}\right]$	$*$
d1	$\left\{\begin{array}{l}3\times3,64\\3\times3,64\end{array}\right\}*$	$\left[\begin{array}{l}1\times1,32;1\times1,32\\3\times3,64\\1\times1,64\end{array}\right]$	$*$
output	1×1, stride 1, w.o. BN and ReLU		

with the output channel size 256 and the dilation rate $1, 6, 12, 18$ and then concatenate their outputs. The output stage use the convolution operator with 1×1 kernel size and stride 1 without batch normalization [13] and ReLU [21] for the *attraction field map* prediction.

4.4. Training

We follow standard deep learning protocol to estimate the parameters Θ.

Loss function. We adopt the l_1 loss function in training.

$$\ell(\hat{\mathbf{a}}, \mathbf{a}) = \sum_{(x,y)\in\Lambda} \|\mathbf{a}(\mathbf{x},\mathbf{y}) - \hat{\mathbf{a}}(\mathbf{x},\mathbf{y})\|_1. \qquad (13)$$

Implementation details. We train the two networks (U-Net and *a-trous* Residual U-Net) from scratch on the training set of Wireframe dataset [12]. Similar to [12], we follow the standard data augmentation strategy to enrich the training samples with image domain operations including mirroring and flipping upside-down. The stochastic gradient descent (SGD) optimizer with momentum 0.9 and initial learning rates 0.01 is applied for network optimization. We

train these networks with 200 epochs and the learning rate is decayed with the factor of 0.1 after every 50 epochs. In training phase, we resize the images to 320×320 and then generate the offset maps from resized line segment annotations to form the mini batches. As discussed in Section 3, the rescaling step with reasonable factor will not affect the results. The mini-batch sizes for the two networks are 16 and 4 respectively due to the GPU memory.

In testing, a test image is also resized to 320×320 as input to the network. Then, we use the squeeze module to convert the *attraction field map* to line segments. Since the line segments are insensitive to scales, we can directly resize them to original image size without loss of accuracy. The squeeze module is implemented with C++ on CPU.

5. Experiments

In this section, we evaluate the proposed line segment detector and make the comparison with existing state-of-the-art line segment detectors [12, 5, 1, 23]. As shown below, our proposed line segment detector outperforms these existing methods on the WireFrame dataset [12] and YorkUrban dataset [6].

5.1. Datasets and Evaluation Metrics

We follow the evaluation protocol from the deep wireframe parser [12] to make a comparison. Since we train on the Wreframe dataset [12], it is necessary to evaluate our proposed method on its testing dataset, which includes 462 images for man-made environments (especially for indoor scenes). To validate the generalization ability, we also evaluate our proposed approach on the YorkUrban Line Segment Dataset [6].

All methods are evaluated quantitatively by the *precision* and *recall* as described in [12, 20]. The *precision* rate indicates the proportion of positive detection among all of the detected line segments whereas *recall* reflects the fraction of detected line segments among all in the scene. The detected and ground-truth line segments are digitized to image domain and we define the "positive detection" pixel-wised. The line segment pixels within 0.01 of the image diagonal is regarded as positive. After getting the *precision* (P) and *recall* (R), we compare the performance of algorithms with F-measure $F = 2 \cdot \frac{P \cdot R}{P + R}$.

5.2. Comparisons for Line Segment Detection

We compare our proposed method with Deep Wireframe Parser[3] [12], Linelet[4] [5], the Markov Chain Marginal Line Segment Detector[5](MCMLSD) [1] and the Line Segment
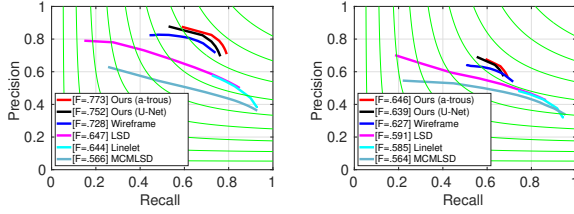
Figure 4. The PR curves of different line segment detection methods on the WireFrame [12] and YorkUrban [6] datasets.

Table 2. F-measure evaluation with state-of-the-art approaches on the WireFrame dataset and York Urban dataset. The last column reports the averaged speed of different methods in *frames per second* (FPS) on the WireFrame dataset.

Methods	Wireframe dataset	York Urban dataset	FPS
LSD [23]	0.647	0.591	19.6
MCMLSD [1]	0.566	0.564	0.2
Linelet [5]	0.644	0.585	0.14
Wireframe parser [12]	0.728	0.627	2.24
Ours (U-Net)	**0.752**	**0.639**	**10.3**
Ours (*a-trous*)	**0.773**	**0.646**	**6.6**

Detector (LSD)[6] [23]. The source codes of compared methods are obtained from the authors provide links. It is noticeable that the authors of Deep Wireframe Parser do not provide the pre-trained model for line segment detection, we reproduced their result by ourselves.

Threshold Configuration In our proposed method, we finally use the *aspect ratio* to filter out false detections. Here, we vary the threshold of the *aspect ratio* in the range $(0, 1]$ with the step size $\Delta\tau = 0.1$. For comparison, the LSD is implemented with the $-\log(\text{NFA})$ in $0.01 \times \{1.75^0, \ldots, 1.75^{19}\}$ where NFA is the number of false alarm. Besides, Linelet [5] use the same thresholds as the LSD to filter out false detection. For the MCMLSD [1], we use the top K detected line segments for comparison. Due to the architecture of Deep Wireframe Parser [12], both the threshold for the junction localization confidence and the orientation confidence of junctions branches are fixed to 0.5. Then, we use the author recommended threshold array $[2, 6, 10, 20, 30, 50, 80, 100, 150, 200, 250, 255]$ to binarize the line heat map and detect line segments.

Precision & Recall To compare our method with state-of-the-arts [12, 5, 1, 23], we evaluate the proposed method on the Wireframe dataset [12] and YorkUrban dataset [6]. The precision-recall curves and F-measure are reported in Figure 4 and Table 2. Without bells and whistles, our proposed method outperforms all of these approaches on Wireframe and YorkUrban datasets by a significant margin even with a 18-layer network. Deeper network architecture with ASPP

---

[6]http://www.ipol.im/pub/art/2012/gjmr-lsd/

module further improves the F-measure performance. Due to the YorkUrban dataset aiming at Manhattan frame estimation, some line segments in the images are not labeled, which causes the F-measure performance of all methods on this dataset decreased.

Speed We evaluate the computational time consuming for the abovementioned approaches on the Wireframe dataset. We run 462 frames with image reading and result writing steps and count the averaged time consuming because the size of testing images are not equal. As reported in Table 2, our proposed method can detect line segments fast (outperforms all methods except for the LSD) while getting the best performances. All experiments perform on a PC workstation, which is equipped with an Intel Xeon E5-2620 2.10 GHz CPU and 4 NVIDIA Titan X GPU devices. Only one GPU is used and the CPU programs are executed in a single thread.

Benefiting from the simplicity of original U-Net, our method can detect line segments fast. The deep wireframe parser [12] spends much time for junction and line map fusion. On the other hand, benefiting from our novel formulation, we can resize the input images into 320×320 and then transform the output line segments to the original scales, which can further reduce the computational cost.

Visualization Further, we visualize the detected line segments with different methods on Wireframe and YorkUrban datasets (see Figure 5). The threshold configurations for visualization are as follow:

1. The *a-contrario* validation of LSD and Linelet are set as $-\log \epsilon = 0.01 \cdot 1.75^8$;

2. The top 90 detected line segments for the MCMLSD are visualized;

3. The threshold for line heat map is 10 for the deep wireframe parser;

4. The upper bound of aspect ratio is set as 0.2 for our results.

By observing these figures, it is easy to find that Deep Wireframe Parser [12] can detect more complete line segments compared with the previous methods, however, our proposed approach can get better result in the perspective of completeness. On the other hand, this junction driven approach indeed induces some uncertainty for the detection. The orientation of line segments estimated by junction branches is not accurate, which will affect the orientation of line segments. Meanwhile, some junctions are misconnected to get false detections. In contrast, our proposed method gets rid of junction detection and directly detect the line segments from images.

Comparing with the rest of approaches [23, 1, 5], the deep learning based methods (including ours) can utilize

Figure 5. Some Results of line segment detection on Wireframe [12] and YorkUrban [6] datasets with different approaches LSD [23], MCMLSD [1], Linelet [5], Deep Wireframe Parser [12] and ours with the *a-trous* Residual U-Net are shown from left to right. The ground truths are listed in last column as reference.

the global information to get complete results in the low-contrast regions while suppressing the false detections in the edge-like texture regions. Due to the limitation of local features, the approaches [23, 1, 5] cannot handle the results with global information and still get some false detections even with powerful validation approaches. Although the overall F-measure of LSD is slightly better than Linelet, the visualization results of Linelet are cleaner.

6. Conclusion

In this paper, we proposed a method of building the duality between the region-partition based attraction field representation and the line segment representation. We then pose the problem of line segment detection (LSD) as the region coloring problem which is addressed by learning convolu-

tional neural networks. The proposed attraction field representation rigorously addresses several challenges in LSD such as local ambiguity and class imbalance. The region coloring formulation of LSD harnesses the best practices developed in ConvNets based semantic segmentation methods such as the encoder-decoder architecture and the *a-trous* convolution. In experiments, our method is tested on two widely used LSD benchmarks, the WireFrame dataset [12] and the YorkUrban dataset [6], with state-of-the-art performance obtained and $6.6 \sim 10.4$ FPS speed.

Ackowledgement

References

[1] Emilio J Almazan, Ron Tal, Yiming Qian, and James H Elder. MCMLSD: A Dynamic Programming Approach to Line Segment Detection. In *Proc. of CVPR*, 2017. 2, 3, 6, 7, 8

[2] Dana H Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981. 2, 3

[3] J. Brian Burns, Allen R. Hanson, and Edward M. Riseman. Extracting straight lines. *IEEE TPAMI*, 8(4):425–455, 1986. 3

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *Proc. of ECCV*, 2018. 5

[5] Nam Gyu Cho, Alan Yuille, and Seong Whan Lee. A Novel Linelet-Based Representation for Line Segment Detection. *IEEE TPAMI*, 40(5):1195–1208, 2018. 1, 2, 3, 6, 7, 8

[6] Patrick Denis, James H. Elder, and Francisco J. Estrada. Efficient Edge-Based Methods for Estimating Manhattan Frames in Urban Imagery. In *Proc. of ECCV*, pages 197–210, 2008. 1, 2, 6, 7, 8

[7] Liuyun Duan and Florent Lafarge. Image Partitioning Into Convex Polygons. In *Proc. of CVPR*, pages 3119–3127, 2015. 1

[8] Olivier D Faugeras, Rachid Deriche, Hervé Mathieu, Nicholas Ayache, and Gregory Randall. The Depth and Motion Analysis Machine. *International Journal of Pattern Recognition and Artificial Intelligence*, 6(2&3):353–385, 1992. 1

[9] Yasutaka Furukawa and Yoshihisa Shinagawa. Accurate and robust line segment extraction by analyzing distribution around peaks in hough space. *CVIU*, 92(1):1–25, 2003. 3

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Sun Jian. Deep Residual Learning for Image Recognition. In *Proc. of CVPR*, 2016. 5

[11] Qibin Hou, Jiangjiang Liu, Ming-Ming Cheng, Ali Borji, and Philip H S Torr. Three Birds One Stone: A Unified Framework for Salient Object Segmentation, Edge Detection and Skeleton Extraction. In *Proc. of ECCV*, 2018. 2, 3

[12] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. Learning to Parse Wireframes in Images of Man-Made Environments. In *Proc. of CVPR*, 2018. 1, 2, 3, 5, 6, 7, 8

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 6

[14] Varsha Kamat-Sadekar and Subramaniam Ganesan. Complete description of multiple line segments using the hough transform. *Image Vision Comput.*, 16(9-10):597–613, 1998. 3

[15] Iasonas Kokkinos. Pushing the Boundaries of Boundary Detection Using Deep Learning. In *Proc. of ICLR*, 2016. 3

[16] Iasonas Kokkinos. UberNet: Training a 'Universal' Convolutional Neural Network for Low-, Mid-, and High-Level Vision using Diverse Datasets and Limited Memory. In *Proc. of CVPR*, 2017. 3

[17] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. Richer Convolutional Features for Edge Detection. In *Proc. of CVPR*, 2017. 3

[18] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. Convolutional Oriented Boundaries. In *Proc. of ECCV*, 2016. 1, 3

[19] Kevis Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. Convolutional Oriented Boundaries: From Image Segmentation to High-Level Tasks. *IEEE TPAMI*, 40(4):819–833, 2018. 3

[20] David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE TPAMI*, 26(5):530–549, 2004. 3, 6

[21] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 6

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proc. of MICCA*, 2015. 2, 5

[23] R G von Gioi, J Jakubowicz, J M Morel, and G Randall. LSD: A Fast Line Segment Detector with a False Detection Control. *IEEE TPAMI*, 32(4):722–732, 2010. 1, 2, 3, 4, 6, 7, 8

[24] Gui-Song Xia, Julie Delon, and Yann Gousseau. Accurate Junction Detection and Characterization in Natural Images. *IJCV*, 106(1):31–56, 2014. 1

[25] Tianzhu Xiang, Gui-Song Xia, Xiang Bai, and Liangpei Zhang. Image stitching by line-guided local warping with global similarity constraint. *Pattern Recognition*, 83:481–497, 2018. 1

[26] Saining Xie and Zhuowen Tu. Holistically-Nested Edge Detection. In *Proc. of ICCV*, 2015. 1, 3

[27] Chi Xu, Lilian Zhang, Li Cheng, and Reinhard Koch. Pose Estimation from Line Correspondences: A Complete Analysis and a Series of Solutions. *IEEE TPAMI*, 39(6):1209–1222, 2017. 1

[28] Zezhong Xu, Bok-Suk Shin, and Reinhard Klette. Accurate and robust line segment extraction using minimum entropy with hough transform. *IEEE TIP*, 24(3):813–822, 2015. 3

[29] Zezhong Xu, Bok-Suk Shin, and Reinhard Klette. Closed form line-segment extraction using the hough transform. *Pattern Recognition*, 48(12):4012–4023, 2015. 3

[30] Zezhong Xu, Bok-Suk Shin, and Reinhard Klette. A statistical method for line segment detection. *CVIU*, 138:61–73, 2015. 3

[31] Nan Xue, Gui-Song Xia, Xiang Bai, Liangpei Zhang, and Weiming Shen. Anisotropic-Scale Junction Detection and Matching for Indoor Images. *IEEE TIP*, 27(1):78 – 91, 2018. 1

[32] Zhan Yu, Xinqing Gao, Haibing Lin, Andrew Lumsdaine, and Jingyi Yu. Line Assisted Light Field Triangulation and Stereo Matching. In *Proc. of ICCV*, 2013. 1

[33] Yibiao Zhao and Song-Chun Zhu. Scene Parsing by Integrating Function, Geometry and Appearance Models. In *Proc. of CVPR*, pages 3119–3126, 2013. 1

[34] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. LayoutNet: Reconstructing the 3D Room Layout from a Single RGB Image. *CoRR*, abs/1803.08999, 2018. 1