A Novel Linelet-Based Representation for Line Segment Detection

Nam-Gyu Cho, Alan Yuille, *Fellow, IEEE*, and Seong-Whan Lee, *Fellow, IEEE*

Abstract—This paper proposes a method for line segment detection in digital images. We propose a novel linelet-based representation to model intrinsic properties of line segments in rasterized image space. Based on this, line segment detection, validation, and aggregation frameworks are constructed. For a numerical evaluation on real images, we propose a new benchmark dataset of real images with annotated lines called YorkUrban-LineSegment. The results show that the proposed method outperforms state-of-the-art methods numerically and visually. To our best knowledge, this is the first report of numerical evaluation of line segment detection on real images.

Index Terms—Intrinsic properties of digital line, probabilistic line segment representation, line segment validation, image edge detection

1 Introduction

A line segment in a digital image consists of a set of connected pixels which are located on the boundary of geometric, mostly man-made, objects such as buildings, boxes, and roads. It can be detected from both straight lines and straight edges. Line segment detection has been studied for several decades in computer vision and plays an important role for other visual tasks, such as segmentation [1], [2], object detection [3], [4], vanishing point estimation [5], [6], [7], and perceptual organization [8], [9].

A line segment is the result of digitizing the intensity of a line in the real world. At first glance, it is merely a set of connected pixels. However, when we look in more details, there are several intrinsic properties (Fig. 1) that previous methods have not fully exploited, e.g., stair-wise shape.

The Hough transform (HT) [10], one of the most well known edge-based methods, takes an edge map, then searches for possible line configurations by voting on each edge pixel's contribution. As a final step, line segments are produced by cutting a line using thresholds. Generally, it works well on a lattice structure. However, its voting process generates false positives on complicated regions having dense edge responses, such as a tree. Additionally, it fails to detect from regions having small gradient magnitudes, thus no or small edge response, such as a facade. There have been several approaches to improve drawbacks of HT: The progressive probabilistic Hough transform (PPHT) for false positive control [11], orthogonal image scanning [12] and an analysis in Radon space [13] for a better accuracy, and connectivity enforcement to connect separated small line segments [14]. However, performance is still not fully satisfactory (see Fig. 2).

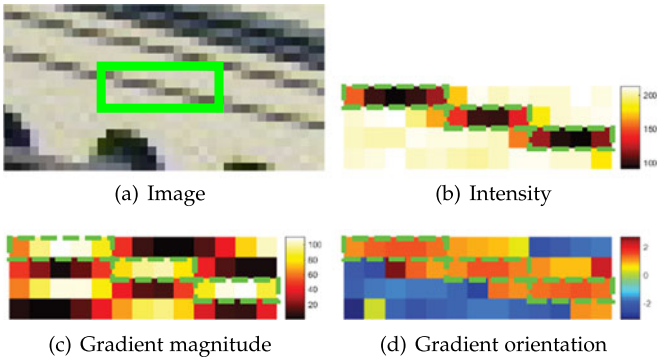
In contrast to edge-based methods, Burn et al. [15] proposed a method which detects line segments locally using gradient orientation values. Later, Grompone von Gioi et al. [16] proposed a fast Line Segment Detector (LSD) which further improved [15]. In LSD, a line segment is detected as an estimate from a line-support region, where pixels within that region have similar gradient orientation value, if it passed a meaningful alignment criterion test, the Helmholtz principle [17]. This principle allows a method to control false positive theoretically without requiring parameter tuning for a group of images. More generally, it can be adopted to a large range of detection applications where the detection can be described in terms of the alignment of primitives, such as pixels in LSD. It was demonstrated that LSD performs better than edge-based methods. However, its estimation process does not guarantee accurate results when line-support regions consist of pixels having similar gradient magnitude, or the Helmholtz principle rejects true positives, target lines in an image, in complex regions (see Fig. 2). Recently, an Edge Drawing Line segments detection (EDLines) method [18] was proposed to speed up LSD by altering the gradient orientation to the edge. However, according to our evaluation on a benchmark dataset, its performance is worse than LSD.

More recently, Liu et al. [19] proposed a Line Segment Perception (LSP) framework motivated by a biological visual cortex. In this framework, given a gray-value input image, orientation information is processed in a feed-forward manner, from simple cells to complex cells, then finally to hypercomplex cells. Since the goal is to perceive line segments its detection results are not precise though it is demonstrated that LSP method is robust to noises and clutters.

All these methods are typically evaluated simply by visualization, which is a non-quantitative evaluation; their

- *N.-G. Cho and S.-W. Lee are with the Department of Brain and Cognitive Engineering, Korea University, Anam-dong, Seongbuk-ku, Seoul 02841, Korea. E-mail: {southq, sw.lee}@korea.ac.kr.*
- *A. Yuille is with the Departments of Cognitive Science and Computer Science, John Hopkins University, Baltimore, MD 21218 and with the Department of Statistics, University of California, Los Angeles, CA 90095. E-mail: yuille@stat.ucla.edu.*

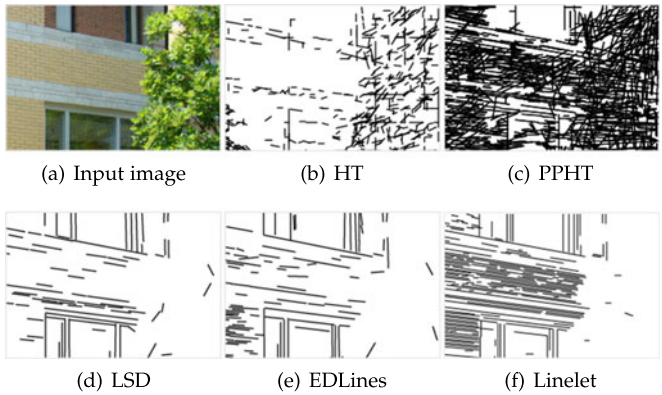
Manuscript received 29 Apr. 2016; revised 22 Feb. 2017; accepted 18 Apr. 2017. Date of publication 10 May 2017; date of current version 10 Apr. 2018. (Corresponding author: Seong-Whan Lee.)
Recommended for acceptance by Y. Liu.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2017.2703841



(a) Image (b) Intensity

(c) Gradient magnitude (d) Gradient orientation

Fig. 1. Illustration of how a line segment is represented in a digital image. At first glance, it is merely a set of connected pixels (a). However, when we look in more details, there are several intrinsic properties such as a stair-like shape of horizontal chunks (b), gradient magnitude pattern (c), and gradient orientation pattern (d) (see Section 2.2 for more details). Note that subfigures (b), (c), and (d) are magnified from the green rectangle of (a) (best seen in color).



(a) Input image (b) HT (c) PPHT

(d) LSD (e) EDLines (f) Linelet

Fig. 2. Line segment detection comparison on a challenging situation where a facade and tree exist together. Edge-based methods (HT and PPHT) generate a number of false positives on the tree which has a complex edge response. They also fail to detect lines on the facade where edge response is not significant. In the meanwhile, local gradient-based methods (LSD and EDLines) successfully reject false positives on the tree by exploiting the Helmholtz principle for a validation strategy, which also leads to rejecting true positives on the facade. The proposed method (Linelet), however, gives a better result compared to others; it detects most of lines on the facade and only a few number of false positives on the tree.

purpose is to demonstrate how methods detect line segments in complex situations [11], [16], [20]. There have been only a few attempts on numerical evaluation, such as evaluation on synthesized line images by varying noise [21]. However, this is not satisfactory since synthesized images cannot cover all the possible challenging real world situations, such as varying illumination, occlusion, etc. One of the most difficult problems for numerical evaluation is annotating line segments in an image accurately. This requires great efforts because sometimes lines are not clearly visible due to occlusions or blur effects, and usually, for an urban image, it is likely that several hundreds of lines exist and need to be labeled.

To summarize, i) the performance of previous methods is not fully satisfactory since they do not consider some intrinsic properties of line segments in digital image (Fig. 1); taking Fig. 2 for example, edge-based methods (HT and PPHT) generate a number of false positives on the tree which has a complex edge response. They also fail to detect lines on some regions such as the facade where edge response is not significant. This is mainly due to a lack of a proper representation of a line, the gradient orientation value similarity of pixels around a line segment as LSD approach tried, which leads to the failure of identifying false positives. In the meanwhile, local gradient-based methods (LSD and EDLines) successfully reject false positives on the tree by exploiting the Helmholtz principle for a validation strategy. However, this principle also leads to rejecting some true positives on the facade where pixels violate the principle. These limitations motivated us to study the intrinsic properties of straight lines (see Section 2.1 analyzing previous method in terms of the intrinsic properties). ii) Also, only using visual evaluation, it is difficult to judge whether a method performs well.

To overcome the first problem, we analyze the intrinsic properties of line segments (Fig. 1) and present a novel linelet-based line segment representation where a linelet is defined as a set of connected pixels either horizontally or vertically, as a result of digitization. This representation models the intrinsic properties by an undirected graph. It also enables us to build not only a probabilistic validation framework to determine how likely a detection is true positive but also an aggregation framework to connect two line segments having similar intrinsic properties. Based on this, a novel line segment detection framework is presented. To conduct a numerical evaluation, as a benchmark, we constructed the YorkUrban-LineSegment dataset by annotating 102 urban images of the YorkUrban dataset [22]. This was a big effect which took several months of work.

The contribution of the proposed method is two-fold: i) the proposed method exploits the intrinsic properties of line segment in image that previous methods have not fully exploited. This enables the proposed method to detect line segments accurately with allowing only a few false positives compared to others. ii) We construct a new line segment dataset and evaluate the proposed method and compare it with other methods on the new dataset. To our best knowledge, this is the first report of numerical evaluation of line segment detection on real images.

1.1 Overview of the Proposed Method

Fig. 3 shows the pipeline of the proposed method. First, the intensity (gray scale), gradient magnitude, and gradient orientation are obtained from an input image (*Preprocessing*). Then, the gradient magnitude is suppressed either horizontally or vertically such that resulting image only contains a set of pixels which of each has the maximum value among its neighbors (*Non-maximal suppression*). Linelets are detected by connecting pixels horizontally ($\text{Linelets}^{\text{Hor}}$) or vertically ($\text{Linelets}^{\text{Ver}}$) from the non-maximal suppressed gradient magnitudes (*Linelet detection*). Next, closely located linelets are grouped using their properties (*Grouping*). Then, the angle of each line segment is estimated using a probabilistic model for lines (*Estimation*). Detected line segments are validated to judge whether they are from true line segments using a model which uses cues from the gradient magnitude and orientation (*Validation*). Finally, an aggregation process is conducted to further improve the results by connecting line segments that are likely to come from the same line in the image (*Aggregation*).

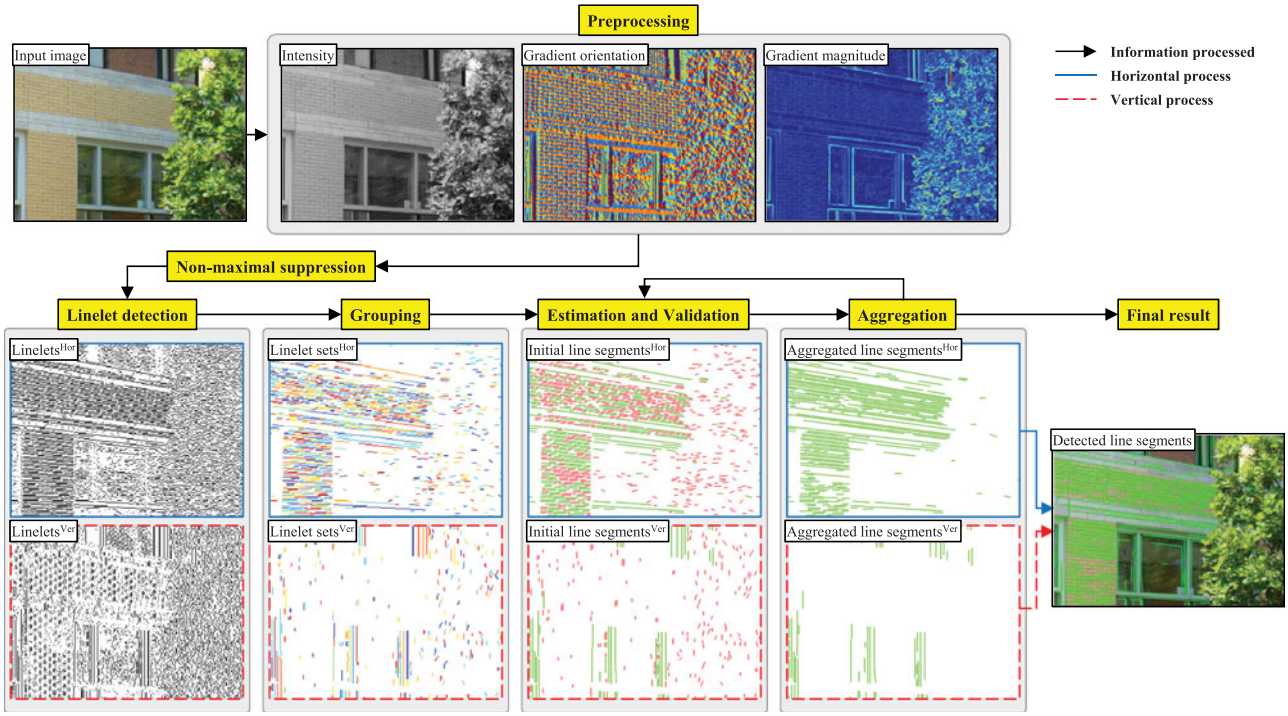


Fig. 3. Illustration of the proposed framework. First, linelets are detected horizontally and vertically (Section 4.1). Next, neighboring linelets are grouped using their properties (Section 2.2). The angle is estimated using a probabilistic model for lines (Section 4.2), then detections are validated to judge whether they are true positives, target lines in the image, using a model which uses cues from the gradient magnitude and orientation (Section 3). Finally, an aggregation process is conducted to further improve the results by connecting line segments that are likely from the same line segment (Section 4.3) (best seen in color).

The rest of the paper is organized as follows. Section 2 establishes the fundamentals of the proposed method by beginning with the analysis of intrinsic properties of line segments in terms of linelet, then describes the proposed linelet-based line segment representation. Based on this, Section 3 presents the validation process. Section 4 describes the detection and aggregation frameworks that integrate all components together. Section 5 introduces our new line segment dataset, Section 6 comments on the experimental results, and Section 7 concludes the paper.

2 Line Segment Representation Using Linelets

As a result of digitization, a line[1] consists of a set of horizontal or vertical chunks of connected pixels, where the size of a chunk is determined by the angle of the line. For example, the slanted line in Fig. 1b consists of 3 chunks and each chunk contains a set of pixels connected horizontally (more example of lines are shown in Fig. 4). Moreover, each chunk has some intrinsic properties, such as length similarity between adjacent chunks (Fig. 1b), similar gradient magnitude values (Fig. 1c) and orientation values (Fig. 1c), that characterize lines in images.

In this paper, to exploit these intrinsic properties, we define a *linelet* as a set of pixels connected in one direction, either horizontally or vertically, and use this as a basic unit to represent lines. Thus, a line can be represented by an undirected graph model where its nodes correspond to linelets and edge indicates the connection between adjacent linelets. The conditional distribution is calculated in terms of potentials where the unary and pairwise potential functions capture intrinsic properties with regard to the angle.

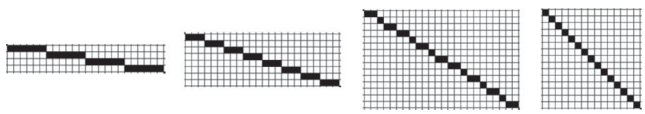
The remainders of this section describe the definition of linelet (Section 2.1), its properties (Section 2.2), and the representation based on it (Section 2.3).

2.1 Line as a Set of Linelets

Motivated by the object boundary quantization representation [23], a line L in an image consists of a set of pixels where their relationship is formulated as,

$$L = \{(x_i, y_i) | y_i = \lfloor \tan(\theta) x_i + \beta \rfloor\}, \quad (1)$$

where θ is the angle between the line and x axis. β is the intercept with the y axis, $\lfloor \cdot \rfloor$ denotes the floor function, which gives rise to the staircase shape (Fig. 1b), and both x_i and y_i are greater than 0. In this paper, we will consider lines with angle $\theta \in [-\frac{\pi}{4}, \frac{3\pi}{4})$ in image plane. Additionally, according to the angle θ, lines are divided into two subsets: *horizontal lines* for $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4})$ and *vertical lines* for $\theta \in [\frac{\pi}{4}, \frac{3\pi}{4})$ (see Fig. 5); as a result, each subset of lines can be detected separately. For example, vertical lines can be detected



(a) 14° (6 pixel) (b) 26° (3 pixel) (c) 43° (2 or 1 pixel) (d) 45° (1 pixel)

Fig. 4. Illustration of four lines with different angle (slope). The lengths, of connected pixels, are of size 6, 3, 2, and 1 respectively, according to the angle (1).

1. In this paper, we use *line segment* and *line* interchangeably.

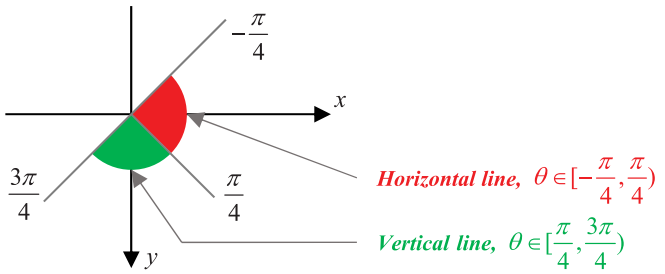


Fig. 5. Illustration of two subsets of lines considered in this paper. For convenience of notation, we assume that the range of angle θ in image plane is from $-\frac{\pi}{4}$ to $\frac{3\pi}{4}$. Additionally, according to the angle θ, lines are divided into two subsets: *horizontal lines* for $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4})$ and *vertical lines* for $\theta \in [\frac{\pi}{4}, \frac{3\pi}{4})$.

exactly the same way that horizontal lines are detected after rotating the image 90 degrees.

Regarding a linelet l, a line L is reformulated as,

$$L = \left\{ l_i^{\xi} | i = 1 : M, \xi \in (hor, ver) \right\}, \tag{2}$$

where M is the number of linelets. Note that a line L has only one type of linelets, either l^{hor} or l^{ver}. Each linelet is defined as,

$$\begin{aligned} l^{hor} &= \{(x_i, y_i) | y_i = y_j, \forall i, j\} \\ l^{ver} &= \{(x_i, y_i) | x_i = x_j, \forall i, j\}, \end{aligned} \tag{3}$$

where pixels of each linelet have the same y_i value for l^{hor} and same x_i value for l^{ver}. Note that x_i and y_i follow (1).

2.2 Linelet Properties

Now we describe a linelet and its properties characterizing lines in digital image. For convenience in explanation, we will consider horizontal lines, i.e., lines with angle $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4})$ (vertical lines case will be discussed together).

During digitization of a line in the real world, a loss of information occurs due to noise or a resolution of vision sensor [23]. As a result, lines in images have some properties, see Fig. 1, that we summarize as follows,

(i) The length of a linelet $len(l)$ is determined by the angle θ. $len(l)$ increases from 1 to the width (horizontal) of an image as $|\theta|$ decreases from $\frac{\pi}{4}$ to 0. However, in a real situation, the length is affected by noise. In this paper, we use $len(l)$ to denote the length of a linelet extracted from an image and $len(\tilde{l}(\theta))$ to denote the length of a linelet constituting a line with the angle θ in an ideal case. Note that $len(\tilde{l}(\theta))$ can vary for some θ due to digitization (see Fig. 4c).

(ii) When l_i and l_j are adjacent, the length difference is small

$$\left| len(l_i) - len(l_j) \right| \leq \tau_{len}, \tag{4}$$

where τ_{len} denotes the linelet length difference threshold to consider variations caused by the angle θ (see Fig. 4) and the noise.

(iii) Adjacent linelets l_i and l_j (assuming that l_i is located at the left side) from the same line satisfy the condition,

$$y(l_j) - y(l_i) = \pm 1, \tag{5}$$

where $y(l_i)$ denotes the y_i value of the linelet l_i. The difference is equal to 1 for $\theta \in (0, \frac{\pi}{4})$ and -1 for $\theta \in [-\frac{\pi}{4}, 0)$.

(iv) A gradient magnitude discontinuity occurs across a linelet (Fig. 1c). Thus, similar to edges in an image, pixel intensity changes rapidly across lines, and this is also represented as the local maximum of gradient magnitude, perpendicular to the direction of the edge, of an image.

(v) The gradient magnitude distribution of pixels along a linelet has a characteristic pattern with regard to θ. The middle pixel of a linelet has the highest magnitude where the line in the real world goes through the center of the middle pixel. The magnitude decreases as the location of a pixel gets away from the middle pixel and the decrease in size is determined by the angle θ. Taking Fig. 1c for example, in the linelet at the top (highlighted with dotted rectangle), the middle pixel has the highest magnitude, around 100, and the decrease size is roughly 20 for adjacent pixels. As a ratio, the decrease size is 0.2 ($= \frac{20}{100}$), which equals to the slope of the line ($\tan(11.3^\circ) = 0.2$) roughly. Thus, we denote $\tau_{mag}(\theta)$ as the decreasing magnitude size in a ratio between adjacent pixels of a linelet given θ, and this is approximated as $\tau_{mag}(\theta) \approx |\tan(\theta)|$.

(vi) Similar to (*v*), the gradient orientation distribution of pixels along a linelet has a characteristic pattern with regard to θ. Overall, the direction is orthogonal to the angle of the line with some deviation (see Fig. 1d). Similar to (*v*), $\tau_{ori}(\theta)$ denotes the gradient orientation difference, in a ratio, between adjacent pixels of a linelet given θ, and is approximated as $\tau_{ori}(\theta) \approx |\tan(\theta)|$.

The first four are geometric properties and the last two are gradient properties. Note that they are interrelated in terms of θ (since they are all the results of digitization). In other words, in terms of frequency, the geometric properties are of high frequency. They contain discriminative information since usually lines have strong edge (high frequency) response. Thus, detecting line would become easy once obtaining these properties are guaranteed. However, that is not easy due to noise. In the meanwhile, the gradient properties are of low frequency. They are robust to noise and can be used to construct a strong statistical pixel alignment model, such as the Helmholtz principle [17], but not as discriminative as geometric properties.

The characteristic and limitation of previous approaches can be understood with regard to these properties:

(a) Hough transform methods [11], [24] use the edge map of an image (*iv*), then vote on each edge pixel's contribution to a possible line configuration (i, ii, and iii). When lines are located on a lattice structure where the affect of noise becomes small, results are precise since they directly work with edge pixels where it is highly likely that a line is located. However, their voting scheme sometimes leads to inaccurate results; they generate false positives from complex regions having dense edge response (see Fig. 2) due to the lack of local representation of

gradient properties (v and vi); Also, false negatives are generated when an edge detector, such as Canny [25], fails to capture edges due to its parameter setting for a certain group of image.

(b) Gradient orientation-based methods [15], [16], [18], in particular LSD [16], use line-support regions by clustering pixels sharing similar orientation value (vi). Each region is validated with the Helmholtz principle [17] by checking whether the set of pixels in the region satisfies the meaningful alignment criterion. Then the angle of the line is estimated by calculating the first inertia axis of the region by taking the gradient magnitude as the pixel's mass (iv and v). This method is theoretically because no parameter setting is necessary through all these processes and false positives are well controlled. However, The preciseness of detection, how accurately detected lines are located on the lines in images, is dependent on the distribution of gradient orientation near a line. For example, when a line is located on a region which consists of pixels having similar gradient magnitude, its center and angle estimates tend to have a small error. We observed that this is due to the lack of discriminative component such as edges that HT methods exploited (i, ii, and iii). Additionally, the Helmholtz principle rejects not only false positives but also true positives that are usually located on complex regions such as a facade (see Fig. 2).

2.3 Graphical Model for Lines

We now develop an undirected graphical model for lines. We use this in our pipeline for estimating the angle of a line, and aggregating adjacent line segments of similar properties (see Section 4.3). A line is an undirected graph,

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}), \tag{6}$$

where the nodes $\mathcal{V}$ correspond to linelets and the edge $\mathcal{E}$ indicates the connection between adjacent linelets (see Fig. 6).

The conditional distribution is given by a Gibbs distribution of the angle in terms of potentials where the unary and pairwise potential functions capture intrinsic properties (Section 2.2) with regard to linelets,

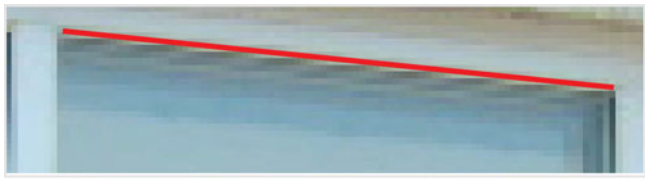
$$p(\theta|L) = \frac{1}{Z(\theta)} \exp\left\{ \sum_{i \in \mathcal{V}} \phi_i(l_i, \theta) + \sum_{ij \in \mathcal{E}} \psi_{ij}(l_i, l_j, \theta) + \phi^g(L, \theta) \right\}, \tag{7}$$

where $L = \{l_i\}_{i \in \mathcal{V}}$ denotes the nodes as a set of linelets and $Z(\theta)$ is a normalization constant. $\phi_i(l_i, \theta)$ is unary potential formulated as,

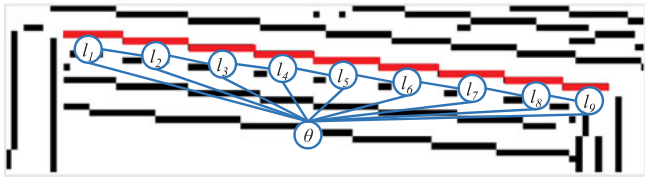
$$\phi_i(l_i, \theta) = \phi_i^{len}(l_i, \theta) + \phi_i^{mag}(l_i, \theta) + \phi_i^{ori}(l_i, \theta). \tag{8}$$

We exploit properties of Section 2.2 to model each unary potential,

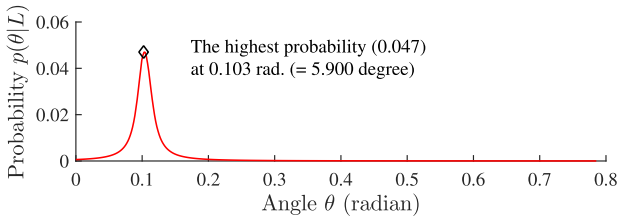
$$\phi_i^{len}(l_i, \theta) = \frac{1}{|len(l_i) - len(\tilde{l}(\theta))|^2 + \epsilon} \tag{9}$$



(a) Input image and a line segment detect by the proposed method (highlighted as red).



(b) Linelets detected from the image (a). Note that a set of linelets corresponding to the detect line segment of (a) is highlighted as red. The graphical model, which consists of nine nodes (linelets), is illustrated on the top of them.



(c) Probability distribution of the line segment of (a) calculated by (7). Note that the line segment has the highest probability when $\theta = 0.103\,\text{rad}\ (= 5.9°)$.

Fig. 6. Illustration of the graphical model (b) and its probability distribution (c) of the detected line segment (a).

$$\phi_i^{mag}(l_i, \theta) = \frac{1}{|\tau_{mag}(\theta) - E[dG_{mag}(l_i)]|^2 + \epsilon} \tag{10}$$

$$\phi_i^{ori}(l_i, \theta) = \frac{1}{|\tau_{ori}(\theta) - E[dG_{ori}(l_i)]|^2 + \epsilon}, \tag{11}$$

where $G_{mag}(l_i)$ and $G_{ori}(l_i)$ are the gradient magnitude and orientation values of pixels of the linelet respectively. $dG_{mag}(l_i)$ denotes the difference values of the gradient magnitude of adjacent pixels of l_i and $dG_{ori}(l_i)$ is the difference values from the gradient orientation. $E[\cdot]$ is the expectation value and ϵ is a small constant to prevent potentials from becoming infinite. $\phi^{len}(l_i, \theta)$ models the property (*i*); it gives higher score having small length difference between $len(l_i)$ and $len(\tilde{l}(\theta))$, the length of a linelet detected from an image and the length of a linelet constituting a line with the angle θ respectively. $\phi^{mag}(l_i, \theta)$ models the property (*v*); it returns higher score when l_i has the gradient magnitude distribution of its pixels similar to that of true line segment of the angle θ. In a similar manner, $\phi^{ori}(l_i, \theta)$ models the property (vi).

$\psi_{ij}(l_i, l_j, \theta)$ is pairwise potential which exploits properties (ii) and (iii),

$$\psi_{ij}(l_i, l_j, \theta) = \begin{cases} 1 & \text{if } |len(l_i) - len(l_j)| \le \tau_{len} \text{ and} \\ & |y(l_i) - y(l_j)| = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

$\phi^g(L,\theta)$ measures the fitness of θ by a distance error,

$$\phi^g(L,\theta) = 1 \Big/ \left(\frac{1}{M}\sum_{i\in V} d_p(L,l_i|\theta)\right), \tag{13}$$

where $d_p(L,l_i|\theta)$ measures the perpendicular distance, the distance between the center of l_i and L, as the average of each linelet l_i's center, in the direction orthogonal to θ.

The angle value $\hat{\theta}$ giving the highest probability (7) is chosen as the angle of the line segment L,

$$\hat{\theta} = \arg\max_\theta p(\theta|L). \tag{14}$$

Fig. 6 shows an illustration of the graphical model of a line segment and its probability distribution calculated by (7).

3 Probabilistic Line Segment Validation

We now develop a probabilistic model which we used to determine whether a detected line is a true positive or not. This model uses cues from the gradient magnitude and orientation (the property v and vi of Section 2.2). This proposed validation step is novel since it gives result as a probability of being true positive. This probabilistic result can be exploited by other applications in a richer way. For example, vanishing point detection [5], [6], [7] or perceptual organization [8], [9] can formulate the goal by taking each line segment's probability, as a weight, into account.

HT [10] generates false positives from non-line objects such as a tree (see Fig. 2). LSD [16] solves this problem by calculating a Number of False Alarm score to measure whether a line segment is a meaningful alignment by taking the Helmholtz principle [17]. However, it sometimes removes not only false positives but also true positives located on complex regions such as a facade (see Fig. 2).

To overcome limitations of previous methods, we propose a probabilistic cue integration method, which exploits intrinsic properties of line segments, to judge whether a line segment is detected from true line segment or not (see Fig. 7). The probability is calculated by a mixture of experts model. Each cue is reformulated by a likelihood density given by a Gaussian, and a prior is considered as the length of line, using Bayes theorem. In principle, we consider it likely that a line segment is detected from true line segment when it holds intrinsic properties. For other case, it measures how it is likely that a line segment is from random noise. As a cue, we consider gradient magnitude (the property v) and orientation (the property vi), since other properties will be considered during linelet detection (Section 4.1) in advance. The remaining part of this section will describe details.

For a line segment L_i, with its observation $\mathcal{O}_i$, e.g., gradient magnitude G_{mag} and orientation G_{ori}, we define a binary random variable f_i,

$$f_i = \begin{cases} f_i^+ & \text{if } L_i \text{ is a true positive} \\ f_i^- & \text{otherwise.} \end{cases} \tag{15}$$

We then estimate the probability $P(f_i^+|\mathcal{O}_i)$ to determine whether L_i is detected from a true line segment. We use a mixture of experts model inspired by [26],

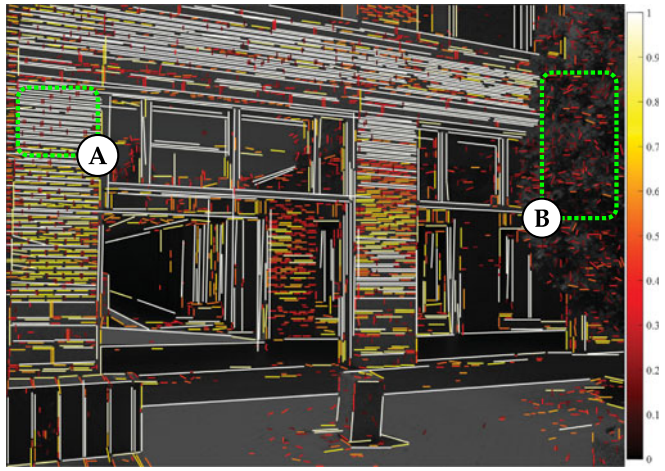


Fig. 7. Illustration of validation results on the image #22. For each detected line segment, the probability of being true positive is colored, from zero (dark red) to one (white). Note that false positives detected from the tree at the right side (B) have low (< 0.5) probability, and then are rejected. In the meanwhile, line segments detected from true lines (A) have high (≥ 0.5) probability (best seen in color).

$$\begin{aligned} P(f_i^+|\mathcal{O}_i) &= \sum_k P(f_i^+, c_k|\mathcal{O}_i) \\ &= \sum_k P(f_i^+|\mathcal{O}_i, c_k)P(c_k|\mathcal{O}_i), \end{aligned} \tag{16}$$

where $c_k \in (mag, ori)$. For each cue, $P(f_i^+|\mathcal{O}_i, c_k)$ is formulated using Bayes theorem,

$$P(f_i^+|\mathcal{O}_i, c_k) = \frac{\lambda_{i,k}^+ P(f_i^+|c_k)}{\lambda_{i,k}^+ P(f_i^+|c_k) + \lambda_{i,k}^- P(f_i^-|c_k)}, \tag{17}$$

where $\lambda_{i,k}^\pm \equiv p(\mathcal{O}_i|f_i^\pm, c_k)$ are the likelihood densities given $f_i^\pm$ and c_k. Additionally, we use a cue-independent prior, $P(f_i|c_k) = P(f_i)$, and determine this prior based on the length of line segment. The remaining parts will describe each component.

3.1 Likelihood Densities

We model the likelihood density of each cue, the gradient magnitude and orientation, as a normal distribution. Both likelihoods are determined from the pixels of linelets of a line segment,

$$\lambda_{i,k}^\pm = p(\mathcal{O}_i|f_i^\pm, c_k) = \mathcal{N}(E[d\mathcal{O}_i]; \tau_k^\pm(\theta), \sigma_k^\pm), \tag{18}$$

where $k \in (mag, ori)$. $d\mathcal{O}_i$ is $\{dG_{mag}(l_i)\}$ for the gradient magnitude and $\{dG_{ori}(l_i)\}$ for the gradient orientation. For true positive, $\tau_k^+(\theta)$ is set as $|\tan(\theta)|$ (Section 2.2). For other case, $\tau_k^-(\theta)$ is learned from a set of images made of random noise. Similar to $\tau_k^\pm(\theta)$, for true positive, σ_k^+ is set as 0.36 ($\approx \frac{\pi}{9}$). For other case, σ_k^- is learned from a set of images made of random noise (details will be discussed in Section 6.1).

3.2 Prior

We model the prior $P(f_i^\pm)$ as a logistic function with regard to the length of a line segment,

$$P(f_i^+) = \frac{1}{1 - e^{-a|L_i|-b}}, \tag{19}$$

Fig. 8. Illustration of the linelet detection process. (From left to right) an input image, the gradient magnitude, non-maximal suppression ($NMS(G_{mag}, \text{“vertical”})$) result, and linelets detected (and color coded). Note that, in the gradient magnitude, dark color means low value and bright color high value (best seen in color).

where $|L_i|$ is the length of line segment L_i, the Euclidean distance between two end points. Parameters a and b are learned from a set of images made of random noise (details will be discussed in Section 6.1).

3.3 Cue Integration

We integrate line segment decisions from different cues using a mixture of experts model, which enables us to control the influence of each cue and adapt it to the information contained in each line (details will be discussed in Section 6.1).

4 Line Segment Detection and Aggregation Framework

This section gives a global picture of the line segment detection framework and aggregation algorithm (Fig. 3).

4.1 Linelet Detection

Linelet detection is done similar to edge [25]. First the gradient magnitude G_{mag} and orientation G_{ori} of an input image I are calculated,

$$G_{mag} = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right], \; G_{ori} = \tan^{-1}\left(\frac{\partial I}{\partial y} \Big/ \frac{\partial I}{\partial x}\right). \tag{20}$$

We exploit the fourth property (iv of Section 2.2); non-maximal suppression is adapted in orthogonal direction, horizontally or vertically. As a result, G^{hor}_{mag} and G^{ver}_{mag} are generated,

$$\begin{aligned} G^{hor}_{mag} &= NMS(G_{mag}, \text{“vertical”}) \\ G^{ver}_{mag} &= NMS(G_{mag}, \text{“horizontal”}), \end{aligned} \tag{21}$$

where $NMS(G_{mag}, \text{“vertical”})$ conducts non-maximal suppression in vertical direction on G_{mag}. For example, this function suppresses a pixel $G_{mag}(x, y)$ unless it has the maximum value among its neighbors $G_{mag}(x \pm 1, y)$. In other words, $NMS(G_{mag}, \text{“vertical”})$ breaks connection between pixels connected in vertical direction, thus results in pixels aligned horizontally. Next, a set of linelets is detected as connecting pixels having non-zero value in one direction (3). For example, a set of l^{hor} is detected by finding horizontally connected pixels from G^{hor}_{mag}, and similarly l^{ver} in vertical direction from G^{ver}_{mag}. It is likely that l^{hor} and l^{ver} of length 1 are detected from the same pixel. For this case, we drop l^{ver} unless it belongs to a bigger cluster (see the second step of Section 4.2). Fig. 8 illustrates the linelet detection process.

4.2 Line Segment Detection

Fig. 3 shows an illustration of the proposed line segment detection framework. Each step is processed as follows,

(a) First, linelets are detected from the non-maximal suppressed gradient magnitude (21), which is calculated from an input image (20).
(b) Linelets are connected to their neighbors satisfying the intrinsic properties (ii) and (iii) of Section 2.2 until no linelet meeting the condition is found.
(c) The angle is estimated for lines by finding θ which gives the highest probability (14).
(d) Each line segment is validated by calculating the probability of current line segment being detected from true line segment (16) (Section 3),

$$P(f_i^+|\mathcal{O}_i) \; > \; P(f_i^-|\mathcal{O}_i). \tag{22}$$

Note that horizontal linelets are only connected to other horizontal linelet, and same for vertical linelets. Also, $x(l_i)$, the x value of l_i is considered in (5) to connect vertical linelets in (b) and (12) to calculate pairwise potential.

4.3 Line Segment Aggregation

Due to noise, sometimes the step (b) of Section 4.2 fails to connect adjacent linelets of the same line segment. To resolve this limitation, we propose an aggregation method to improve detection results by connecting disjoint line segments that probably came from the same line.

In principle, an aggregated one should be more plausible than each of its parts. We assess the plausibility by considering the perpendicular distance between centers in the direction orthogonal to the line, the difference in angles, the distance between endpoints, the response to the true positive test and, finally, the kurtosis of the distribution of angles along the lines.

(a) For each line segment L_i, we find aggregation candidates $\mathcal{L}_{cand} = \{L_j\}$ satisfying conditions,

$$d_p(L_i, L_j|\theta_i) \leq \tau_p, \tag{23}$$

$$d_a(\theta_i, \theta_j) \leq \tau_a, \tag{24}$$

$$d_l(L_i, L_j) \leq \max(\{len(l^i_k)\}) \tag{25}$$

where θ_i is the angle of L_i estimated by (14). $d_p(L_i, L_j|\theta_i)$ returns the perpendicular distance, the distance between the centers of L_i and L_j, in the direction orthogonal to θ_i. $d_a(\theta_i, \theta_j)$ returns the angle difference between two line segments. $d_l(L_i, L_j)$ returns the shortest distance between end points of each line, and $\max(\{len(l^i_k)\})$ denotes the length of the longest linelet of L_i.

(b) L_j is aggregated with L_i if the aggregated $L_{i'}$ passes the validation test (Section 3) and the distribution of angles along the aggregated is more centered, that is higher kurtosis,

$$P(f_{i'}^+|\mathcal{O}_{i'}) \; > \; P(f_{i'}^-|\mathcal{O}_{i'}), \tag{26}$$

$$Kurt[p(\theta|L_{i'})] \; > \; Kurt[p(\theta|L_i)] \tag{27}$$

where $i' = (i, j)$, and $Kurt[p(\theta|L_{i'})]$ calculates the kurtosis of the distribution of angles corresponding to the linelets in a line. The intuition is that a good line has a more centered distribution of angles (Fig. 9).

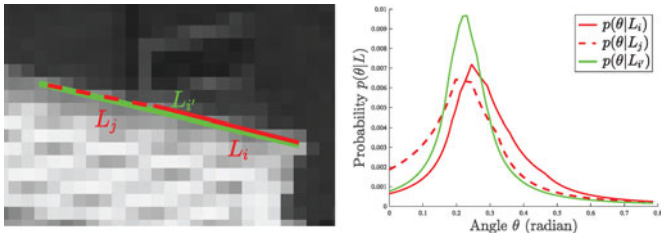


Fig. 9. An example of aggregation of L_i and L_j (left) and the probability distribution (right). The aggregated line segments is more plausible, thus has a peakier probability distribution than before (see text for more details, best seen in color).

4.4 Complexity of the Proposed Framework

The most time-consuming steps of the proposed method are detecting linelets and grouping linelets which have computational costs that are $O(N_{pixel}^2)$ and $O(N_{linelet}^2)$ respectively, where N_{pixel} is the number of pixels and $N_{linelet}$ is the number of linelets. On average, in Matlab implementation, the proposed method takes one minute per image of 640×480 pixel resolution.[2]

5 The YorkUrban-LineSegment Dataset

We now discuss the novel dataset that we construct and use for evaluation. As we mentioned in Section 1, visualization or testing on synthesized images [12], [13], [16] have been mainly used for evaluation due to difficulties of getting accurate ground truth [21]. However, we believe that it is more relevant to evaluate on real images, although there could be small ground truth inaccuracy, because they include more complex and various situations of the real world such as illumination change, texture, noise, or blurry effect compared to synthesized ones.

For this purpose, we propose the YorkUrban-LineSegment, as a new benchmark dataset. We built a web based line segment annotation system (Fig. 10) and annotated line segments on 102 images of 640×480 pixel resolution from the YorkUrban dataset [22], which has been used for evaluating vanishing point estimation performance [5], [6], [7]. Similar to PASCAL Context dataset [27], we wanted to guarantee the highest accuracy of line segments.

Annotation was performed intensely by three in-house annotators. To let them have an idea of how to annotate, two or three examples were shown in advance. Some line segments were too ambiguous to annotate, mainly due to them being too densely located or because blurring effects. In this case, we encouraged annotators to use a zooming function (Fig. 10). Our annotation effort took about two months, around three hours per image. Finally, we obtained around 680 line segment annotations per image—there are around 120 annotations per image in the original dataset. Fig. 11 shows the distribution of line segment annotation with regard to the length and angle. Most of line segments (around 80 percent) have less than 50 pixel length with near horizontal (0 or π) or vertical ($\frac{\pi}{2}$ or $-\frac{\pi}{2}$) angle.

2. Please note that current implementation is written in unoptimized code. When tested on bigger size images, the time cost increases exponentially.

Fig. 10. Screenshots of our annotation system. Annotators can use a zoom in or out function when line segments are densely located (top) or ambiguous from the original resolution due to blur effect or small gradient changes (bottom) (best seen in color).

To our best knowledge, there has been no method of line segment detection that demonstrated its detection accuracy numerically on real images. This is the first dataset providing line segment annotations on real images. One might criticize that the number of the images is relatively small compared to those of existing benchmark dataset such as PASCAL VOC [28] and ImageNet [29]. However, we believe that the size of line segment instances, around 70 k, is enough to evaluate performance.

5.1 Annotation Results Analysis

We found several interesting points, which are closely related to our evaluation criterion, from the annotations: i) annotators tended to connect separated line segments (the top row of Fig. 12). This phenomenon can be well described by the study of human's visual perception such as imaginary triangle from Kanizsa's studies [30]. ii) Annotators approximated curves, such as ellipses and circles, by a series of line segments. They start from the longest line segment, then go to the next longest one, and so on. iii) The distribution of annotation (Fig. 11) represents characteristics of urban images, i.e., most of urban images have Manhattan structure [31].

6 Experimental Evaluation

In this section, we evaluate the performance of the proposed method Linelet and compare it to existing methods: Hough Transform [24], Progressive Probabilistic Hough Transform [11], Line Segment Detector with a false positive control [16],

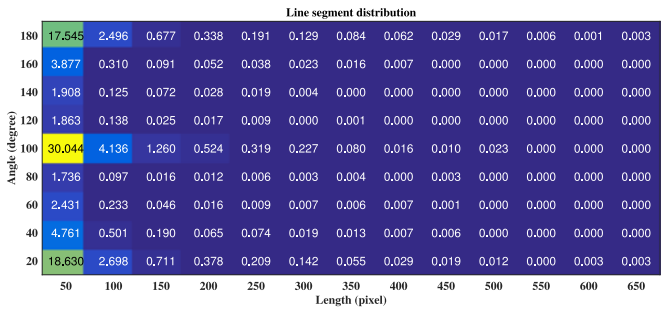


Fig. 11. Distribution of length (column) and angle (row) of the line segment annotation. The number in each cell means the percentage.

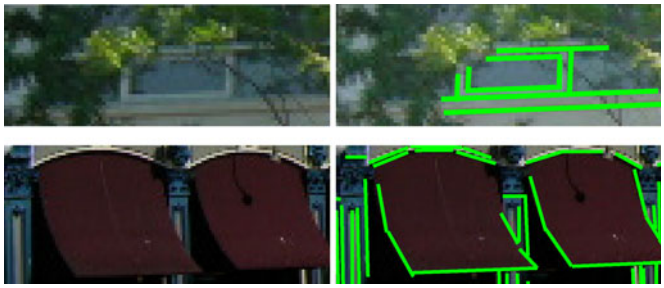


Fig. 12. Annotation examples: Over partial occlusion (the top row) and curved structures (the bottom row). Both examples are of different resolution (best seen in color).

LSD improved (LSDi),[3] and Edge Drawing algorithm (EDLines) [18]. We used the implementation of HT and PPHT provided by the OpenCV [32]. For LSD, LSDi, and EDlines, we used the implementation provided by the authors.

6.1 Implementation Details

We implemented the proposed algorithm in a MATLAB environment. In order to reduce noise effect, after applying the Sobel operator, pixels with gradient magnitude less than 10 are ignored.

Parameters of the proposed method are set as follows: For linelet grouping (Section 4.2), the linelet length difference threshold τ_{len} is 3. In order to learn parameters for validating line segments (Section 3), we generated a set of images of 200×200 pixel size where each pixel is uniformly sampled from 0 to 255 in gray scale; line segments are detected from these images without the proposed validation step, then parameters of models are estimated by taking the mean. As a result, for likelihood densities of being false positive in (18), τ^-_{mag} is 0.94 and σ^-_{mag} is 0.36 for gradient magnitude; τ^-_{ori} is 1.08 and σ^-_{ori} is 0.82 for gradient orientation. For the prior probability of being true positive in (19), a is 20.63 and b is 0.2; thus $P(f_i^-) = 1 - P(f_i^+)$. For cue integration in (16), we set a uniform value, i.e., $P(c_k|\mathcal{O}_i) = \frac{1}{K}$ where K is the number of cues, gradient magnitude and orientation, thus 2 in this paper. For aggregating line segments, the perpendicular distance threshold τ_p in (23) is $\sqrt{2}$ and the angle threshold τ_a in (24) is $\frac{\pi}{4}$.

Parameters for other methods are set as follows: For HT, connecting near line segment threshold τ_{gap} is 5. For PPHT, the threshold minimum gain, number of pixels contribute, τ_{gain} is 30, and τ_{gap} is same as HT. For LSD, LSDi, and EDLines, we used the same setting in the provided code.

6.2 Quantitative Comparison

As we mentioned in Section 5.1, numerical evaluation is complicated since sometimes an annotation contains imaginary lines connecting separated ones (Fig. 12) where it is likely that detection methods predict as separated ones. The opposite situation also occurs—separated annotations on a single line where detection methods predict as a single one. Thus, we use the average of precision (AP), average of recall (AR), intersection over union (IOU) [28], and F-score over all images. The evaluation is processed as follows,

3. The authors of [16] provide the improved version of LSD. This further divides a line-support region on curves, thus results better fitted lines while the original version detects a single line.

TABLE 1
Performance of Methods on the YorkUrban-LineSegment Dataset with Regard to Two Conditions, Con_1 and Con_2

Method	AP		AR		IOU		F-score	
	Con_1	Con_2	Con_1	Con_2	Con_1	Con_2	Con_1	Con_2
HT	0.11	0.17	0.06	0.08	0.06	0.09	0.08	0.11
PPHT	0.10	0.13	0.19	0.27	0.28	0.33	0.13	0.18
LSD	**0.34**	0.38	0.36	0.41	0.32	0.34	0.35	0.39
LSDi	0.32	0.38	0.34	0.40	0.33	0.36	0.33	0.39
EDLines	0.22	0.26	0.23	0.27	0.21	0.24	0.22	0.26
Linelet	**0.34**	**0.39**	**0.45**	**0.52**	**0.41**	**0.43**	**0.39**	**0.45**

1) For each ground truth L_{gt}, find a set of predictions $\{L_p\}$ satisfying following condition,

$$d_p(L_{gt}, L_p|\theta_{gt}) \leq \lambda_{dist} \tag{28}$$

$$d_a(\theta_{gt}, \theta_p) \leq \lambda_{ang} \tag{29}$$

where θ_{gt} and θ_p are the angle of L_{gt} and L_p respectively. $d_p(\cdot,\cdot|\cdot)$ denotes the perpendicular distance same as (23) and $d_a(\cdot,\cdot)$ the angular difference same as (24).

2) $\{L_p\}$ is considered as correct when its intersection with the ground truth L_{gt} is equal to or greater than λ_{area},

$$\frac{L_{gt} \cap \{L_p\}}{L_{gt}} \geq \lambda_{area}. \tag{30}$$

We set thresholds to consider a detection as valid. λ_{dist} is 1 (pixel) and λ_{ang} is $\frac{\pi}{36}$ ($= 5^\circ$). In order to consider the percentage of area, we set two values for λ_{area}: 0.75 and 0.5. Putting all together, λ_{area} with 0.75 sets the first condition (Con_1), and λ_{area} with 0.5 sets the second condition (Con_2). Note that these conditions are tight when considering the fact that the goal is to predict line segments with tiny center (1 pixel) distance and angle (5 degree) difference. Therefore, it is likely that many of detection results are counted as false positives (low AP). Nevertheless, we set such a tight conditions, otherwise it is likely that detections predicted for other ground truth are counted as correct.

Performance of methods on the YorkUrban-LineSegment dataset are summarized in Table 1. For both conditions, Linelet outperforms other methods in all categories. Compared to ours, only LSD and LSDi achieve compatible performance in AP and F-score categories. LSD (and LSDi) and Linelet could achieve higher AP compared to others due to exploiting their own validation method to prune false positives. In terms of AR, however, Linelet outperforms LSD by around 0.1 for both conditions. This implies that the proposed validation step prunes false positives successfully while keeping true positives, where some of them are pruned by LSD (see Section 6.4 for a further analysis). PPHT and EDLines get similar scores in AR, but EDLines outperforms in other categories. Since EDLines also exploits the Helmholtz principle for validation, it could achieve better performance, in particular AP. HT obtains the lowest score in all categories due to the lack of proper line representation or its dependency on edge detectors while PPHT copes with these limitations to a certain extent by taking a false positive control step.

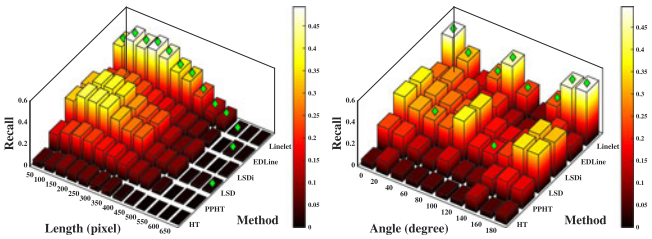


Fig. 13. Average recall with regard to the length (left) and the angle (right) of the ground truth. Each figure shows the performance of each method as a 3D histogram. The *x* axis denotes the range of value, length (left) and angle (right). The *y* axis denotes evaluated methods and the z axis recall score. For each range, e.g., 0 to 50 pixel of the left figure, a method having the highest score is highlighted with the green diamond (best seen in color).

Fig. 13 further analyzes the performance of all methods with regard to the length (left) and the angle (right) of the ground truth under Con_1. According to the distribution of the ground truth (Fig. 11), most of them have line length less than 50 pixel and angles around 90 and 0 degree. Including these groups, i.e., bins of the histogram, the proposed method achieves the highest score for the most of groups. With regard to the length, Linelet performs better than other methods except the last two groups, lines of length 600 or 650 pixel. Note, however, that the proportion of these groups is small (around 0.004 percent) compared to others. With regard to the angle, Linelet outperforms other methods except 40 and 120 degree groups. Our analysis on the performance decrease in these groups is that linelets of 1 pixel in length are vulnerable to noise since they probably alternate between vertical and horizontal. In the meanwhile, LSD and LSDi perform better since the line-support region generation step handles this situation by clustering pixels of similar gradient orientation.

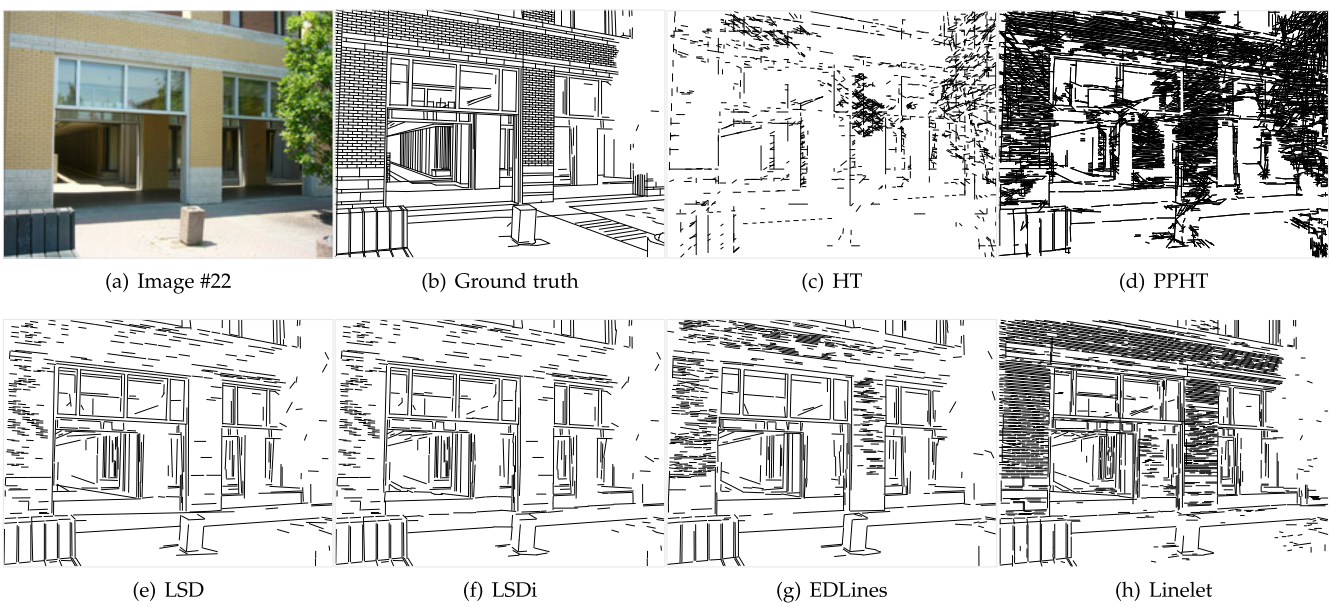
We conducted an additional experiment to see how methods perform on the original YorkUrban dataset. Table 2 summarizes the result. Note that the number of line annotations in the original YorkUrban dataset is relatively small compared to the YorkUrban-LineSegment dataset, 120 versus 680 annotations per image. Overall, the result is similar to the YorkUrban-LineSegment dataset (Table 1). Due to a smaller number of annotations, AP decreases for all methods, e.g., 0.34 to 0.16 for Linelet. One interesting observation is that AR of the proposed method increased significantly while AR of others decreased. This is probably due to the fact that length of lines annotated in the original dataset is longer than the new dataset, thus causing a low overlapping ratio (29) which leads to low AR score. For this reason, the proposed method achieves highest AR because of the aggregation step.

TABLE 2
Performance of Methods on the Original YorkUrban Dataset with Regard to Two Conditions, Con_1 and Con_2

Method	AP		AR		IOU		F-score	
	Con_1	Con_2	Con_1	Con_2	Con_1	Con_2	Con_1	Con_2
HT	0.05	0.08	0.07	0.11	0.10	0.15	0.06	0.09
PPHT	0.03	0.55	0.16	0.24	0.36	0.44	0.06	0.09
LSD	0.11	0.13	0.28	0.33	0.34	0.38	0.16	0.19
LSDi	0.10	0.13	0.25	0.32	0.34	0.39	0.14	0.18
EDLines	0.05	0.07	0.15	0.19	0.16	0.20	0.08	0.11
Linelet	**0.16**	**0.19**	**0.53**	**0.64**	**0.68**	**0.70**	**0.25**	**0.30**

6.3 Qualitative Comparison

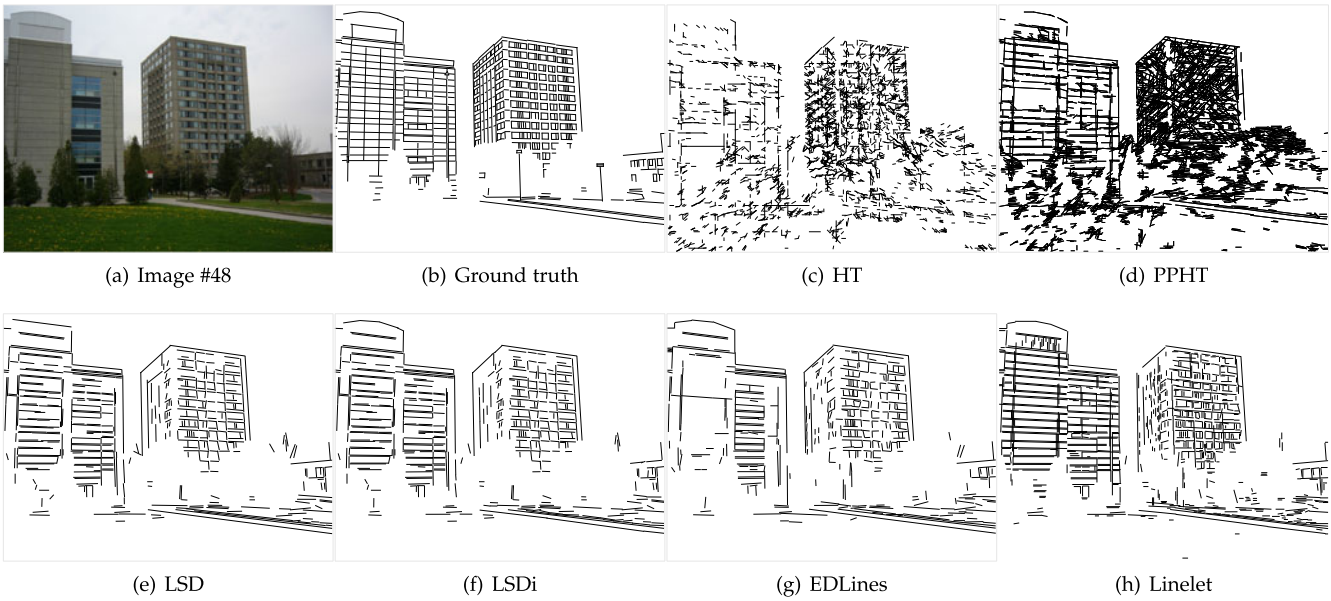
Figs. 14 and 15 show the detection results on the image #22 and #48 of the proposed dataset respectively. Note that methods having their own validation step (LSD, LSDi, EDLines, and Linelet) do not generate false positives from the trees. However, HT and PPHT generate a number of false positives, and this results in low AP scores (Table 1). For complex regions, such as facades and windows, our proposed method detects more ground truth than the other



(a) Image #22 (b) Ground truth (c) HT (d) PPHT

(e) LSD (f) LSDi (g) EDLines (h) Linelet

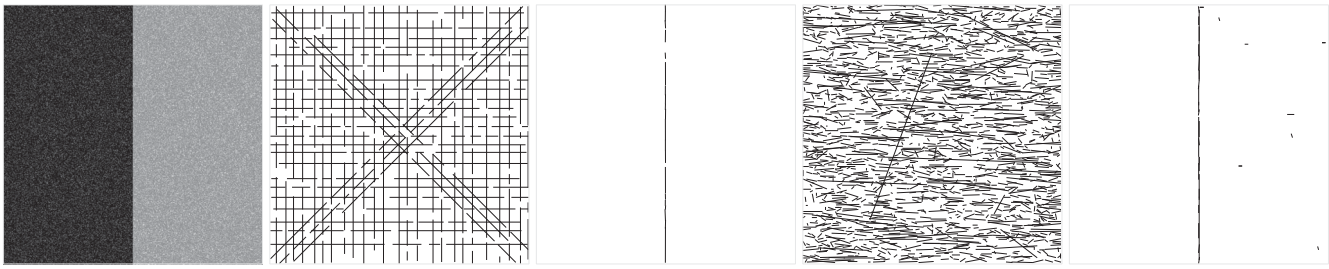
Fig. 14. Line segment detection results on the image #22 of the dataset. On the tree, HT and PPHT, predict a number of false positives while methods having a validation step, LSD, LSDi, EDLines, and Linelet, successfully reject most of them. On the facade, HT fails to detect most of line segments (note that the result will vary for different parameter settings) while other methods do not. LSD, LSDi, and EDLines detect few of line segments from the facade because most of true positives are rejected by the Helmholtz principle. PPHT detects a number of false positives with some of true positives due to a connection step while the proposed method detects most of true positives.



(a) Image #48 (b) Ground truth (c) HT (d) PPHT

(e) LSD (f) LSDi (g) EDLines (h) Linelet

Fig. 15. Line segment detection results on the image #48 of the dataset. Similar to Fig. 14, HT and PPHT predict a number of false positives from the trees and grass region. LSD, LSDi, and EDLines detect most of line segments from the facades. However, compared to the proposed method, they miss some details, such as vertical and horizontal line segments on the left building.



(a) Image, 512×512 (b) HT (c) LSD (d) EDLines (e) Linelet

Fig. 16. Line segment detection results on a noisy edge image (Gaussian noise with $\sigma = 50$). Note that under the presence of strong noise, only Linelet and LSD detect the true edge as a set of fractions while HT and EDLines generate many false positives.

methods. As described in Section 2.2, the Helmholtz principle [17] adopted by LSD, LSDi, and EDLines, rejects not only false positives but also true positives in complex regions. In particular, for the image #22 (Fig. 14), most of line segments on the facade were rejected by the Helmholtz principle while our proposed method relatively detects well. This implies that our proposed validation method exploiting the intrinsic properties of line segments (Section 2.2) is a better one. Fig. 16 shows the detection results on a pure noisy image. Under the presence of strong noise, only Linelet and LSD detect the true edge as a set of fractions; Linelet makes a few false positives. In the meanwhile, HT and EDLines generate many false positives.

Conclusively, the proposed method predicts more ground truth accurately while allowing small false positives. But, we argue that other applications such as vanishing point estimation can prune out short line segments, which are probably false positives. Thus, they still can be benefited. By analogy, for edge detection, it is better to detect too many edges than too few.

6.4 Analysis of the Proposed Validation and Aggregation Methods

We conducted four additional experiments to analyze the influence of the proposed validation and aggregation methods, along with the Helmholtz principle. Results are summarized in Table 3. Note that Linelet$^{\mathrm{nVnA}}$ means the proposed method without both validation and aggregation step, Linelet$^{\mathrm{VnA}}$ without aggregation step, Linelet$^{\mathrm{nVA}}$ without validation step, and Linelet$^{\mathrm{VA}}$ with both validation and aggregation step. Also note that results reported in Section 6.3 are of Linelet$^{\mathrm{VA}}$. Linelet$^{\mathrm{HA}}$ denotes our method modified to use the Helmholtz principle for validation by applying it to the results obtained by Linelet$^{\mathrm{VA}}$. Note that the Helmholtz

TABLE 3
Performance of the Proposed Method on the YorkUrban-LineSegment Dataset with Regard to the Influence of the Validation and Aggregation Methods

Method	AP		AR		IOU		F-score	
	Con_1	Con_2	Con_1	Con_2	Con_1	Con_2	Con_1	Con_2
Linelet$^{\mathrm{nVnA}}$	0.22	0.27	0.44	0.54	**0.46**	**0.48**	0.29	0.36
Linelet$^{\mathrm{VnA}}$	0.30	0.36	0.40	0.49	0.42	0.45	0.34	0.42
Linelet$^{\mathrm{nVA}}$	0.26	0.30	**0.48**	**0.56**	0.44	0.45	0.34	0.39
Linelet$^{\mathrm{VA}}$	0.34	0.39	0.45	0.52	0.41	0.43	**0.39**	**0.45**
Linelet$^{\mathrm{HA}}$	**0.45**	**0.53**	0.33	0.39	0.32	0.35	0.38	**0.45**

Note that "V" and "A" denote the proposed validation and aggregation step respectively. "n" denotes "not exploited" and "H" the Helmholtz principle used in LSD.

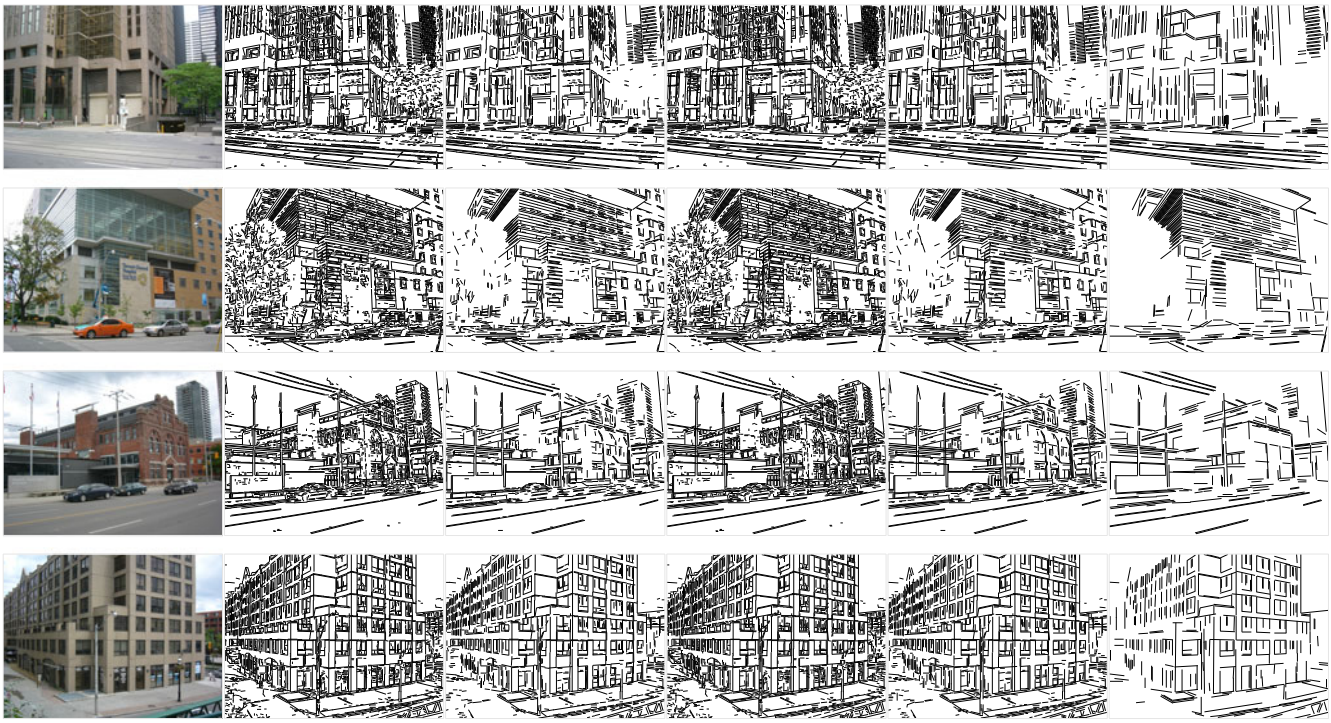


Fig. 17. Illustration of the effect of the validation and aggregation methods on (from top to bottom row) image #70, #80, #84, and #90. "V" denotes validation and "A" aggregation. "nV" denotes that validation method is not used and "nA" means that aggregation method is not used. "HA" denotes the Helmholtz principle is used in our method for validation. Note that the validation process rejects most of false positives, in particular those on trees. Compared to others, Linelet^{HA} seems to favor lines located on clear boundaries.

principle applied to our method is different from LSD. In LSD, it is applied to line-support regions. In the meanwhile, in our experiment, it is applied to line segments, i.e., pixels constituting their linelets. We will first analyze the influence of the proposed validation and aggregation step, then compare with the Helmholtz principle.

The proposed validation step increases the AP by 0.08 in Con_1 and 0.09 in Con_2, Linelet^{nVnA} to Linelet^{VnA} and Linelet^{nVA} to Linelet^{VA}. However, in average, it also decreases the AR by 0.035 in Con_1 and 0.045 in Con_2. The decrease in IOU is around 0.02 for both conditions. Note that the decrease in AR is reduced when the aggregation method is adopted: In Linelet^{nVnA} to Linelet^{VnA} case, the decrease is 0.04 while 0.03 in Linelet^{nVA} to Linelet^{VA} case. The aggregation method increases AR by 0.045 in average.

Taking the Helmholtz principle as the validation step of the proposed method (Linelet^{HA}) increases AP by 0.11 in Con_1 and 0.14 in Con_2, but also decreases AR by 0.12 in Con_1 and 0.13 in Con_2. It looks like the proposed validation step and the Helmholtz made a fair trade-off between AP and AR, thus same F-score. Nevertheless, the proposed method achieves higher IOU. Fig. 17 compares detection results of five methods on some images of the new dataset. Note that the proposed validation step successfully removes false positives while keeping true positives. The aggregation step connects line segments close to each other while keeping them plausible. As summarized in Table 3, Linelet^{VA} and Linetlet^{HA} make a trade-off between precision and recall: The Helmholtz seems to favor a line to be located on a clear boundary in images, thus achieving high AP, while the proposed validation step tries to consider both clear and ambiguous ones.

6.5 Discussion

We conclude that exploiting intrinsic properties of line segments is useful. Experimental results demonstrated that our proposed method can detect more line segments accurately, with a small portion of false positives and false negatives, compared to other methods.

The results of HT and PPHT look bad compared to other methods.[4] However they work well when the length of a target line is long enough with a strong edge response, e.g., outer shapes of buildings (Fig. 14). In the meanwhile, the results of LSD, LSDi, and EDLines, exploiting the Helmholtz principle as a validation step, look similar to those of the proposed method except for some failure on details. This demonstrates that the principle rejects some true positives while the proposed validation step does not. The Helmholtz principle is certainly attractive theoretically. It can be adopted to a large range of detection applications where the detection can be described in terms of the alignment of primitives, such as pixels in LSD. On the other hand, the proposed validation step is only designed for line segment. Compared to theoretically constructed LSD, ours, like most computer methods, has no bounds and instead is evaluated on benchmarked datasets.

Our proposed validation step has several advantages: i) the probability of being a true positive can be exploited by other applications such as vanishing point detection [5], [6], [7] or perceptual organization [8], [9], in a way that the goal of each method can be formulated by taking each line segment's probability, as a weight, into account. ii) This measure can be used to analyze a group of images so that one can

4. Note that the results will vary for different implementation.

judge whether they were taken from urban cities. Additionally, the aggregation step of the proposed method resembles human visual process to a certain degree, e.g., perceiving of an imaginary triangle from Kanizsa's drawing [30], in the sense that separated line segments having similar properties are likely to be connected (i.e., a grouping process).

7 Conclusion

In this paper, linelets were proposed to represent intrinsic properties of line segments in digital images using an undirected graph. Based on these properties, a probabilistic validation measure was presented. A line segment detection framework with an aggregation method was described and its performance was evaluated numerically and visually. For numerical evaluation, we constructed the YorkUrbanLineSegment dataset, which took several months to label, as a benchmark. To the best of our best knowledge, this is the first benchmark dataset for line segment annotations on real images. We demonstrated that the proposed detection method is more accurate than other state-of-the-art methods. As a future work, we plan to extend linelet to represent more geometric shapes such as ellipses, and other applications such as vanishing point estimation.

Acknowledgments

The authors would like to thank J. Flynn, B. Bonev, and S. Qiao who helped for improving early draft, in particular J. Flynn for many helpful discussions, and V. Nguyen for setting up the annotation system. The authors also thank to reviewers for their critical reviews. This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. 2016-0-00152, Development of Smart Car Vision Techniques based on Deep Learning for Pedestrian Safety). Partial support was also provided by US National Science Foundation Expedition in Computing "Visual Cortex on Silicon" with award CCF-1317376. All correspondence should be directed to S.-W. Lee.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[2] J. De Bock and W. Philips, "Line segment based watershed segmentation," in *Comput. Vis./Comput. Graph. Collaboration Tech.*, vol. 4418, pp. 579–586, 2007.

[3] J. Gall, A. Yao, N. Razavi, L. V. Gool, and V. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2188–2202, Nov. 2011.

[4] A. Y. Chia, D. Rajan, M. K. Leung, and S. Rahardja, "Object recognition by discriminative combinations of line segments, ellipses, and appearance features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1758–1772, Sep. 2012.

[5] S. Ramalingam and M. Brand, "Lifting 3d manhattan lines from a single image," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec 2013, pp. 497–504.

[6] Y. Xu, S. Oh, and A. Hoogs, "A minimum error vanishing point detection approach for uncalibrated monocular images of manmade environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 1376–1383.

[7] J. Lezama, R. Grompone Von Gioi, G. Randall, and J.-M. Morel, "Finding vanishing points via point alignments in image primal and dual domains," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 509–515.

[8] M. Park, K. Brocklehurst, R. T. Collins, and Y. Liu, "Translation-symmetry-based perceptual grouping with applications to urban scenes," *Lecture Notes Comput. Sci. Including Subseries Lecture Notes Artif. Intell. Lecture Notes Bioinformat.*, vol. 6494, no. 3, pp. 329–342, 2011.

[9] J. Liu and Y. Liu, "Local regularity-driven city-scale facade detection from aerial images," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 3778–3785.

[10] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recog.*, vol. 13, no. 2, pp. 111–122, 1981.

[11] J. Matas, C. Galambos, and J. Kittler, "Robust detection of lines using the progressive probabilistic Hough transform," *Comput. Vis. Image Understanding*, vol. 78, no. 1, pp. 119–137, 2000.

[12] K. Yang, S. S. Ge, and H. He, "Robust line detection using two-orthogonal direction image scanning," *Comput. Vis. Image Understanding*, vol. 115, no. 8, pp. 1207–1222, 2011.

[13] D. Shi, J. Gao, P. S. Rahmdel, M. Antolovich, and T. Clark, "UND: Unite-and-divide method in Fourier and radon domains for line segment detection," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2501–2506, Jun. 2013.

[14] R. F. C. Guerreiro and P. M. Q. Aguiar, "Connectivity-enforcing hough transform for the robust extraction of line segments," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4819–4829, Dec. 2012.

[15] J. Burns, A. Hanson, and E. Riseman, "Extracting straight lines," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 4, pp. 425–455, Jul. 1986.

[16] R. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, Apr. 2010.

[17] A. Desolneux, L. Moisan, and J.-M. Morel, *From Gestalt Theory to Image Analysis: A Probabilistic Approach*. Berlin, Germany: Springer, 2007.

[18] C. Akinlar and C. Topal, "EDLines: A real-time line segment detector with a false detection control," *Pattern Recog. Lett.*, vol. 32, no. 13, pp. 1633–1642, 2011.

[19] X. Liu, Z. Cao, N. Gu, S. Nahavandi, C. Zhou, and M. Tan, "Intelligent line segment perception with cortex-like mechanisms," *IEEE Trans. Syst. Man Cybern.: Syst.*, vol. 45, no. 12, pp. 1522–1534, Dec. 2015.

[20] N. Merlet and J. Zerubia, "New prospects in line detection by dynamic programming," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 4, pp. 426–431, Apr. 1996.

[21] M. Dubska, A. Herout, and J. Havel, "Pclines – line detection using parallel coordinates," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2011, pp. 1489–1494.

[22] P. Denis, J. H. Elder, and F. J. Estrada, "Efficient edge-based methods for estimating manhattan frames in urban imagery," in *Proc. 10th Eur. Conf. Comput. Vis.*, 2008, pp. 197–210.

[23] L. Dorst and A. W. Smeulders, "Discrete representation of straight lines," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 4, pp. 450–463, Apr. 1984.

[24] P. V. C. Hough, "Machine analysis of bubble chamber pictures," in *Proc. Int. Conf. High Energy Accelerators Instrum.*, 1959, pp. 554–558.

[25] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Nov. 1986.

[26] S. Alpert, M. Galun, A. Brandt, and R. Basri, "Image segmentation by probabilistic bottom-up aggregation and cue integration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 315–26, Feb. 2012.

[27] R. Mottaghi, et al., "The role of context for object detection and semantic segmentation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 891–898.

[28] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, pp. 248–255.

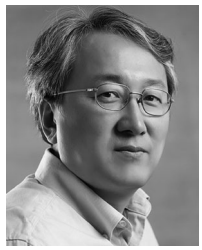
[30] G. Kanizsa, *Organization in Vision: Essays on Gestalt Perception*. Westport, Connecticut, USA: Praeger Publishers, 1979.

[31] J. M. Coughlan and A. L. Yuille, "Manhattan world: Orientation and outlier detection by Bayesian inference," *Neural Comput.*, vol. 15, no. 5, pp. 1063–1088, 2003.

[32] G. Bradski, et al., "The opencv library," *Dr. Dobb's J.*, vol. 25, no. 11, pp. 120–126, 2000.

Nam-Gyu Cho received the BS degree in information and communication engineering from University of Incheon, Incheon, Korea, in 2009, the MS degree in computer science and engineering from Korea University, Seoul, Korea, in 2011, and the PhD degree in brain and cognitive engineering from Korea University, Seoul, Korea, in 2017. His research interests include computer vision, machine learning, and computational models of vision.

Alan Yuille received the BA degree in mathematics from the University of Cambridge in 1976 and received the PhD degree on theoretical physics. His thesis, supervised by Prof. S.W. Hawking, was approved in 1981. He was a research scientist in the Artificial Intelligence Laboratory at MIT and the Division of Applied Sciences at Harvard University from 1982 to 1988. He served as an assistant and associate professor at Harvard until 1996. He was a senior research scientist at the Smith-Kettlewell Eye Research Institute from 1996 to 2002. He was a full professor of Statistics at the University of California, Los Angeles, as a full professor with joint appointments in computer science, psychiatry, and psychology. He moved to Johns Hopkins University in January 2016. His research interests include computational models of vision, mathematical models of cognition, medical image analysis, and artificial intelligence and neural networks. He is a fellow of the IEEE.



Seong-Whan Lee received the BS degree in computer science and statistics from Seoul National University, Seoul, in 1984, and the MS and PhD degrees in computer science from the Korea Advanced Institute of Science and Technology, Seoul, Korea, in 1986 and 1989, respectively. Currently, he is the Hyundai-Kia Motor chair professor and the head of the Department of Brain and Cognitive Engineering, Korea University. He is a fellow of the IAPR, and the Korea Academy of Science and Technology. His research interests include pattern recognition, artificial intelligence and brain engineering. He is a fellow of the IEEE.