Harmonious Semantic Line Detection via Maximal Weight Clique Selection

Dongkwon Jin, Wonhui Park
Korea University
dongkwonjin, whpark@mcl.korea.ac.kr

Seong-Gyun Jeong
42dot.ai
seonggyun.jeong@42dot.ai

Chang-Su Kim
Korea University
changsukim@korea.ac.kr

Abstract

A novel algorithm to detect an optimal set of semantic lines is proposed in this work. We develop two networks: selection network (S-Net) and harmonization network (H-Net). First, S-Net computes the probabilities and offsets of line candidates. Second, we filter out irrelevant lines through a selection-and-removal process. Third, we construct a complete graph, whose edge weights are computed by H-Net. Finally, we determine a maximal weight clique representing an optimal set of semantic lines. Moreover, to assess the overall harmony of detected lines, we propose a novel metric, called HIoU. Experimental results demonstrate that the proposed algorithm can detect harmonious semantic lines effectively and efficiently. Our codes are available at https://github.com/dongkwonjin/Semantic-Line-MWCS.

1. Introduction

A *semantic line* [16, 19] is defined as a meaningful line, separating different semantic regions in a scene, which is approximated by an end-to-end straight line. A group of semantic lines in an image can be regarded as optimal, when they convey the composition of the image harmoniously, as shown in Figure 1(e). Thus, in an optimal set, the lines should harmonize with one another.

Semantic lines provide important visual cues in high-level image understanding [7,9,12,18,20,21,36]. In photography, semantic lines, such as horizontal, vertical, and symmetric ones, are essential composition components. Harmony of such lines are closely related to subjective quality of a photograph [7, 18, 20]. In autonomous driving systems [9, 12, 13], boundaries of road lanes and sidewalks should be detected reliably to control vehicle maneuvers, which can be also described by semantic lines. Moreover, dominant parallel lines intersect at vanishing points [21, 36] under perspective projection, conveying depth impression. They are also semantic lines [16]. However, it is challenging to detect semantic lines, which are often unobvious and implied by complex boundaries of semantic regions.
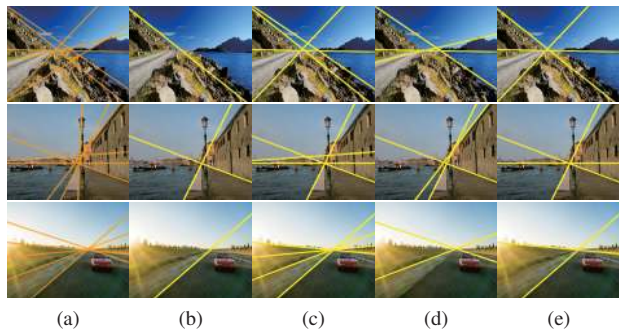

Figure 1: In each scene, straight lines approximating region boundaries are shown in (a). Among them, three subsets of lines are shown in (b), (c), and (d), which are insufficient, over-segmenting, and sub-optimal for describing the composition of the scene, respectively. In contrast, an optimal set of semantic lines in (e) convey the composition of the scene harmoniously.

Many techniques have been developed to detect line segments in a scene by exploiting hand-crafted features [1, 4, 23, 28] or deep features [15, 22, 31, 37]. However, they may extract redundant short line segments or focus on identifying obvious line structures in man-made environments. Recently, several attempts have been made to detect semantic lines [10, 16, 19, 29, 32]. Horizon lines, which are a specific type of semantic lines, have been estimated by CNN-based methods [29, 32]. In [10, 16, 19], semantic line detectors have been proposed. They have two stages: line detection and refinement. In the detection stage, deep line features are extracted to classify line candidates, but implied lines may be undetected or the computational cost for extracting discriminative features can be too high. In the refinement stage, redundant lines are removed through non-maximum suppression (NMS) or pairwise comparison. Although these techniques provide promising results, they may fail to consider the harmony between detected lines and thus may yield sub-optimal results, as shown in Figure 1(d).

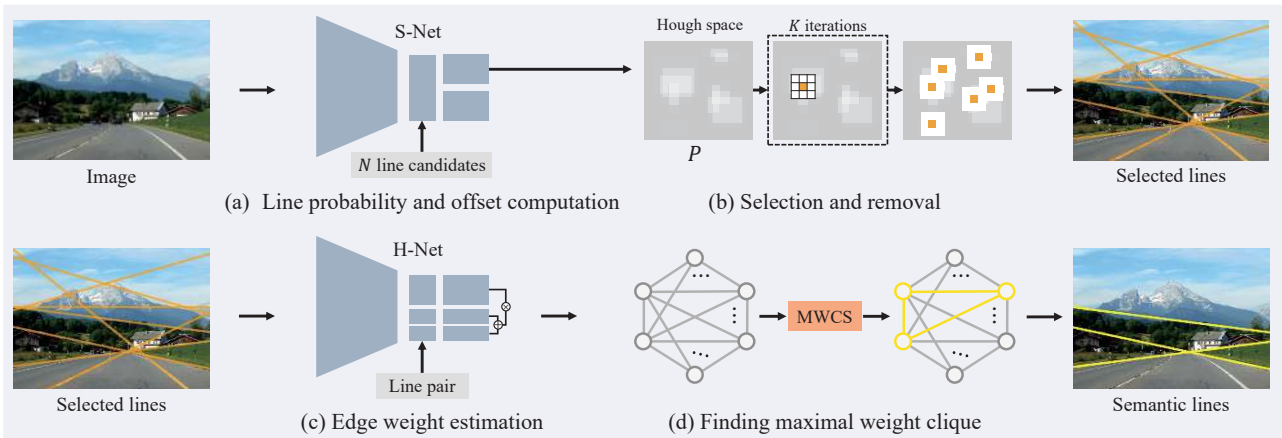In this paper, a novel algorithm to detect an optimal set

Figure 2: Overview of the proposed algorithm.

of harmonious semantic lines is proposed based on maximal weight clique selection (MWCS). We formulate the detection as finding a maximal weight clique in a complete graph [3, 8]. To this end, we design two networks: selection network (S-Net) and harmonization network (H-Net). Given an image and a set of line candidates, S-Net first computes the classification probability and regression offsets of each candidate. Second, we filter out irrelevant lines by performing a selection-and-removal process. Third, we construct a complete graph, in which the node set contains the selected lines. H-Net computes its edge weights. Finally, we determine a maximal weight clique representing harmonious semantic lines. Experimental results demonstrate that the proposed algorithm can detect harmonious semantic lines accurately and efficiently.

This work has the following major contributions:

- We formulate the semantic line detection as finding an maximal weight clique in a complete graph.

- We develop two networks, S-Net and H-Net, to construct the complete graph.

- We introduce a novel metric, called HIoU, to assess the overall harmony of semantic lines, which is more reasonable than the existing metrics in [10, 19].

- The proposed algorithm yields competitive semantic line detection performance to the state-of-the-art DRM technique [16], while reducing the computational complexity by a factor of $\frac{1}{20}$.

2. Related Work

2.1. Line segment detection

Line segments give important visual cues for image semantics. In line segment detection [1, 4, 23, 28], many short segments are detected using low-level features, such as image gradients. This approach, however, may not discriminate meaningful lines from noisy ones. To utilize higher-level features, deep learning methods have been proposed [15, 22, 31, 37]. In [15], a line heat map and junctions were predicted by networks. Then, a wireframe was obtained by connecting the junctions based on the heat map. In [37], a line candidate was generated by connecting two junctions and then was classified into either a salient one or not. In [31], attraction field maps were computed by a network to deal with local ambiguity and class imbalance in line segment detection. In [22], a network was trained with a Hough transform block to combine local information with global line priors. These methods [15, 27, 31, 37] focus on detecting obvious lines in man-made environments.

2.2. Semantic line detection

Semantic lines, located near the boundaries of semantic regions, represent the layout and composition of images. Several methods [5, 10, 16, 19, 29, 32] have been developed to detect implied but semantically meaningful lines. In [5, 29, 32], horizon lines were detected by CNNs, which were refined by exploiting vanishing points or using soft labels of line parameters. In [19], Lee et al. proposed the first semantic line detector. They devised a line pooling layer to extract local features along each line candidate. Those features were fed into classification and regression layers to detect semantic lines. Then, an NMS scheme was performed to remove redundant lines, based on the edge detector [30]. In [16], Jin et al. extracted more discriminative line features by designing a region pooling layer and the mirror attention module. Then, they selected the most semantic lines and removed redundant lines alternately through pairwise ranking and matching. In [10], Han et al. transformed line features into a Hough parametric space to facilitate parallel

processing of multiple line candidates. Then, they trained a network to predict a line probability map, which was used to determine semantic lines by computing the centroids of connected components.

2.3. Road lane detection

In autonomous driving systems, it is important to reliably detect the boundaries of road lanes, sidewalks, or crosswalks. Early methods [2, 11, 12, 35] used hand-crafted low-level features to extract lanes. Recently, to cope with complicated road scenes, attempts have been made to detect road lanes using deep semantic segmentation frameworks [13, 14, 24, 25]. In [24], Pan *et al.* proposed a network to learn spatial relationship of lanes through message passing between convolution layers. In [14], a network was designed to generate attention maps at different layers, which were used to refine the output of deeper ones. In [13], the inter-region affinity graph was constructed to transfer structural relationship between lanes from teacher to student networks. In [25], to achieve a high processing speed, a network was developed to identify the location of each lane on a predefined set of rows only.

3. Proposed Algorithm

Figure 2 is an overview of the proposed algorithm, which contains S-Net and H-Net. First, given an image and a set of line candidates, S-Net computes the line probability and the regression offsets of each candidate. Second, irrelevant candidates are filtered out through a selection-and-removal process. Third, a complete graph, whose node set consists of the selected lines, is constructed and its edge weights are computed by H-Net. Finally, a maximal weight clique, representing harmonious semantic lines, is determined.

3.1. Problem formulation

Semantic lines in an image can be regarded as optimal if they convey the composition of the image harmoniously. In other words, in an optimal set, every pair of semantic lines should harmonize with each other. As in Figure 3(b), a pair of semantic lines should direct visual attention to meaningful regions. In contrast, in Figure 3(c), two lines are redundant or inharmonious. Based on this observation, we formulate the semantic line detection as finding a maximal weight clique in a complete graph [3, 8]. In the complete graph, detected lines form the node set, and each edge weight represents how harmonious the associated two lines are. Thus, by finding a maximal weight clique, we find an optimal set of harmonious semantic lines.

3.2. Node selection: filtering line candidates

It is computationally infeasible to construct a complete graph for all line candidates. Therefore, we select reliable nodes only by filtering line candidates.
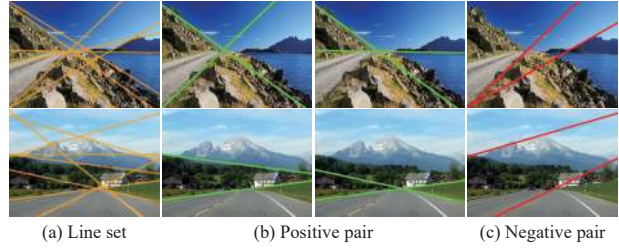

Figure 3: In a line set, a pair of lines can be harmonious and draw visual attention to meaningful regions as in (b). In contrast, they can be redundant or inharmonious as in (c).

Line candidate generation: A line candidate, which is an end-to-end straight line in an image, can be parameterized by polar coordinates in the Hough space [10, 17, 22]. Let $\mathbf{l} = (\rho, \varphi)$ denote a line, where ρ is its distance from the center of the image and φ is its angle from the x-axis. Then, we generate N line candidates, denoted by $\mathbf{l}_n = (\rho_n, \varphi_n)$, $1 \leq n \leq N$, by quantizing ρ and φ uniformly.

S-Net: For each line candidate, we compute its classification probability and regression offsets. To this end, we develop S-Net based on the conventional line detectors [10, 16, 19]. Figure 4(a) shows the architecture of S-Net. From an image, S-Net extracts a convolutional feature map $X = [X^1, X^2, \ldots, X^C] \in \mathbb{R}^{H \times W \times C}$, where H, W, and C denote the feature height, the feature width, and the number of channels. Then, the line feature map $Y = [Y^1, Y^2, \ldots, Y^C] \in \mathbb{R}^{N \times C}$ is obtained by averaging the features of pixels along $\mathbf{l}_n$;

$$Y^c(n) = \frac{1}{|\mathbf{l}_n|} \sum_{\mathbf{p} \in \mathbf{l}_n} X^c(\mathbf{p}) \qquad (1)$$

for $1 \leq n \leq N$ and $1 \leq c \leq C$, where $|\mathbf{l}_n|$ denotes the number of pixels along $\mathbf{l}_n$. We then obtain the probability vector P and the line offset matrix O by

$$P = \sigma(f_1(Y)) \quad \text{and} \quad O = f_2(Y) \qquad (2)$$

where f_1 and f_2 are fully-connected layers of sizes $C \times 1$ and $C \times 2$ for classification and regression, respectively, and $\sigma(\cdot)$ is the sigmoid activation function. For the nth line candidate $\mathbf{l}_n = (\rho_n, \varphi_n)$, P_n indicates the probability that it is semantic, and $O_n = \Delta\mathbf{l}_n = (\Delta\rho_n, \Delta\varphi_n)$ is the offset vector for line refinement in Section 3.4.

The architecture and training process of S-Net are described in detail in the supplemental document.

Selection and removal: In the conventional algorithms [10, 16, 19], to detect semantic lines, only the line candidates with probabilities higher than a threshold are selected and then post-processed (*e.g.* non-maximum suppression). However, this may cause false negatives, which have low probabilities because of being implicit but are semantic nonetheless. To reduce such false negatives, instead
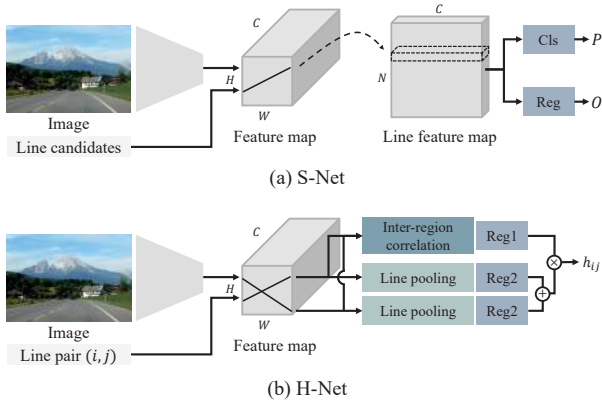
(a) S-Net

(b) H-Net

Figure 4: Architecture of S-Net and H-Net: (a) S-Net takes an image and line candidates and extracts the line feature map. The classification probabilities and regression offsets of the line candidates are then computed by two fully connected layers. (b) H-Net takes an image and a line pair (i, j) to extract local and inter-region features. Two types of regression layers are used to compute the harmony score h_{ij}.

of thresholding, we perform the selection-and-removal process in Figure 2(b). We select the most reliable line $\mathbf{l}_{i^\star}$ by

$$i^\star = \arg\max_i P_i \tag{3}$$

and then remove overlapping lines with the selected one. Specifically, we remove 24 lines within the 5×5 grid centered at $\mathbf{l}_{i^\star}$ in the Hough space [10, 22]. We perform this process K times to compose the node set of K selected lines. Figure 5(b) and (e) show such selected lines on the image and Hough spaces, respectively.

3.3. Edge weighting: harmony score estimation

Inter-region correlation: To tell positive pairs in Figure 3(b) from negative pairs in Figure 3(c), we design the inter-region-correlation (IRC) module that analyzes the regions separated by a pair of lines.

Let $\mathcal{R}_i$, $1 \leq i \leq M$, denote the regions separated by two lines. There can be three or four regions, *i.e.* $M = 3$ or 4. We extract the regional feature vector $\mathbf{r}_i$ of $\mathcal{R}_i$ by

$$\mathbf{r}_i = \tfrac{1}{|\mathcal{R}_i|} \textstyle\sum_{\mathbf{p} \in \mathcal{R}_i} X(\mathbf{p}). \tag{4}$$

We compute the softmax probability a_i of the area $|\mathcal{R}_i|$ to scale the regional feature vectors, and then concatenate the scaled vectors into

$$R = [a_1 \mathbf{r}_1, a_2 \mathbf{r}_2, a_3 \mathbf{r}_3, a_4 \mathbf{r}_4] \tag{5}$$

of size $C \times 4$. If $M = 3$, we fill in the rightmost vector with zeros. Then, R is fed into a fully connected layer to yield

the IRC feature.

H-Net: We develop H-Net using the IRC module. It takes an image and a pair of lines, indexed by i and j, to yield the harmony score h_{ij} ranging from 0 to 1. Figure 4(b) shows the H-Net architecture. The convolution layers of VGG16 [26] are used as the feature extractor, which is followed by three parallel branches of the IRC module and line pooling layers. We employ the line pooling layers to perform the pooling in (1) for lines i and j, respectively. We use two types of regression layers: one for yielding the IRC score of the two lines (Reg1), and the other for computing unary reliability of each line (Reg2). Finally, we compute the harmony score h_{ij} by multiplying the IRC score with the average of the unary reliability levels.

We configure the training data for H-Net as follows. It is assumed that every pair of ground-truth semantic lines in an image harmonize with each other. Thus, we declare such pairs as positive, while the others as negative. In other words, a line pair (i, j) is positive only if both lines i and j are semantic. Then, the harmony score $\bar{h}_{ij}$ is annotated as 1 or 0 depending on whether the pair (i, j) is positive or not. However, this strict definition of a positive pair causes a class imbalance: there are too few positive pairs. Thus, we disturb the line locations of each positive pair and annotate the corresponding harmony score $\bar{h}_{ij}$ to be proportional to $e^{-(d_i^2 + d_j^2)}$, where d_i and d_j denote the disturbances of lines i and j. Also, the loss function for training H-Net is defined as $\ell_{\mathrm{H}} = (h_{ij} - \bar{h}_{ij})^2$, where $\bar{h}_{ij}$ is the ground-truth harmony score and h_{ij} is its estimate. The supplemental document describes the training process and architecture of H-Net in more detail.

3.4. Graph optimization: finding harmonious lines

Graph construction: We construct a complete graph $G = (\mathcal{V}, \mathcal{E})$, in which the node set $\mathcal{V} = \{v_1, v_2, \ldots, v_K\}$ represents the K lines selected using S-Net in Section 3.2. Every pair of lines are connected by an edge in the edge set $\mathcal{E} = \{(v_i, v_j) : i \neq j\}$. Each edge is assigned a weight $w(v_i, v_j) = h_{ij}$ by H-Net in Section 3.3. Figure 5(f) visualizes a complete weighted graph.

MWCS: As mentioned earlier, a set of semantic lines is optimal, if any two lines in the set are harmonious with each other. Thus, finding such an optimal set is equivalent to finding a clique of nodes [8], which are mutually connected and have a maximal sum of weights (*i.e.* harmony scores).

Let θ denote a clique, represented by the index set of member nodes. Then, we define the harmonization energy $E_{\mathrm{harmony}}(\theta)$ of clique θ as

$$E_{\mathrm{harmony}}(\theta) = \textstyle\sum_{i \in \theta} \sum_{j \in \theta, j > i} w(v_i, v_j) \tag{6}$$

which is the sum of all edge weights in θ. Finding the clique that maximizes this energy is NP-hard [6]. However, in this

| (a) | (b) | (c) | (d) | (e) | (f) | (g) |

Figure 5: Illustration of the proposed algorithm: (a) input image, (b) selected lines through the selection-and-removal process, (c) semantic lines, (d) probabilities of line candidates in the Hough space, (e) node set corresponding to the selected lines, (f) complete graph, (g) maximal clique. In (f), an edge is depicted in green or red depending on whether its weight is above the threshold κ in (8) or not. As the weight approaches zero, the transparency increases.

work, K is set to be a small number. The default K is 8. There are about 2^K possible cliques, which are also manageable. Thus, exhaustive search is adopted to find a maximal weight clique. First, we generate the set of possible cliques $\Theta = \{\theta_t\}$ in the graph G, where each clique θ_t consists of more than two nodes. Then, we select the maximal weight clique $\theta^\star$ that maximizes the harmonization energy:

$$\theta^\star = \arg\max_{\theta_t \in \Theta} E_{\text{harmony}}(\theta_t) \qquad (7)$$

subject to a constraint

$$\min_{i,j \in \theta} w(v_i, v_j) > \kappa \qquad (8)$$

where κ is a threshold. If there is no clique satisfying the constraint, we select the maximal single-node clique $\theta^\star = \{i^\star\}$ by

$$i^\star = \arg\max_i h_{ii}. \qquad (9)$$

The self-harmony score h_{ii} is obtained by applying the same line as duplicated input to H-Net.

After obtaining the set of harmonious semantic lines, we refine each line by

$$\mathbf{l}_{v_i} + \Delta\mathbf{l}_{v_i} \qquad (10)$$

where $\Delta\mathbf{l}_{v_i}$ denotes the offset vector, generated by the regression layer of S-Net. Figure 5(c) and (g) show the set of harmonious semantic lines on the image and Hough spaces.

4. Experimental Results

4.1. Datasets

SEL [19]: It is the first semantic line dataset, containing 1,750 outdoor images, which are split into 1,575 training and 175 testing images. Each semantic line is annotated by the coordinates of two end points on an image boundary.

SEL_Hard [16]: It is a more challenging dataset for testing semantic line detectors. It contains 300 test images, selected from the ADE20K segmentation dataset [34]. Its semantic lines are less obvious and more severely occluded in more cluttered scenes.

SL5K [33]: It is a rich and diverse dataset in terms of the number of lines and scene categories. It is composed of 4,000 training and 1,000 testing images.

CULane [24]: It is a dataset for road lane detection, containing 88,000 training images. Its 34,680 test images are classified into 9 categories. For each image, the pixel-wise mask for up to 4 road lanes is provided. The proposed algorithm is tested on 3,911 test images in the 'no lane' category, in which each lane is highly implied or even invisible.

4.2. Metrics

Conventional metrics: There are two existing metrics to assess semantic line detection results: mIoU [19] and EA-score [10]. In the mIoU metric, a detected line is regarded as correct if its mIoU score with the ground-truth semantic line is greater than a threshold τ as illustrated in Figure 6(a). In the EA-score, a detected line is regarded as correct if its similarity with the ground-truth is greater than the threshold as shown in Figure 6(b). The similarity is composed of two factors S_d and S_θ, which are based on the Euclidean distance between the midpoints of the lines and the angular distance of the lines, respectively. In both metrics, the precision and the recall are computed by

$$\text{Precision} = \frac{N_l}{N_l + N_e}, \quad \text{Recall} = \frac{N_l}{N_l + N_m} \qquad (11)$$

where N_l is the number of correctly detected semantic lines, N_e is the number of false positives, and N_m is the number of false negatives. Then, the F-measure is computed by

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \qquad (12)$$

The area under curve (AUC) performances of the precision, recall, F-measure curves are measured in the entire range of the threshold τ, which are denoted by AUC_P, AUC_R, and AUC_F, respectively [19].

However, these metrics measure only the positional accuracy of each detected line. They do not consider how harmonious multiple detected lines are with one another in a scene. Hence, they may yield misleading scores, as exemplified in Figure 7.

HIoU metric: We propose the harmony-based intersection-over-union (HIoU) metric to assess the overall harmony of detected lines. Detected lines tend to convey harmonious impression about the composition of an image, when their

$$\text{mIoU} = \frac{\text{IoU}(R,R') + \text{IoU}(Q,Q')}{2}$$

(a) mIoU metric

$$\text{EA} = (S_\theta(l_1,l_2) \cdot S_d(l_1,l_2))^2$$

(b) EA-score

$$\text{HIoU} = \frac{\text{IoU}(s_1,t_1) + \text{IoU}(s_2,t_1) + \cdots + \text{IoU}(t_3,s_5)}{9}$$
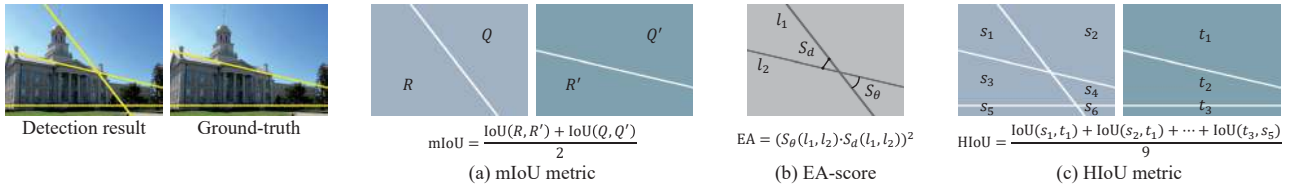
(c) HIoU metric

Figure 6: Illustration of two existing metrics of mIoU [19] and EA-score [10] and the proposed HIoU metric.

Table 1: Comparison of the AUC and HIoU scores (%) on the SEL and SEL_Hard datasets. The processing speeds in frames per second (fps) are also compared. For the AUC scores, the mIoU metric is used.

	SEL				SEL_Hard				fps
	AUC_P	AUC_R	AUC_F	HIoU	AUC_P	AUC_R	AUC_F	HIoU	
SLNet [19]	80.72	<u>84.50</u>	82.57	77.87	74.22	70.68	72.41	59.71	7.35
DHT [10]	<u>89.27</u>	78.53	83.56	79.62	83.55	67.98	75.09	63.39	**30.30**
DRM [16]	85.44	**87.16**	<u>86.29</u>	<u>80.23</u>	87.19	**77.69**	**82.17**	**68.83**	1.05
Proposed	**89.61**	84.21	**86.83**	**81.03**	**87.60**	<u>72.56</u>	<u>79.38</u>	65.99	<u>21.74</u>

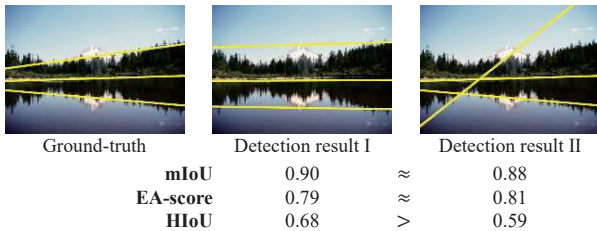|			
Ground-truth	Detection result I		Detection result II
mIoU	0.90	≈	0.88
EA-score	0.79	≈	0.81
HIoU	0.68	>	0.59

Figure 7: Comparison of mIoU [19], EA-score [10], and the proposed HIoU metric: There are two detection results for the same ground-truth. In result I, the position of each detected line is different from the ground-truth, but the detected lines convey the composition of the image relatively well. In result II, two detected lines match the ground-truth exactly, but they are not harmonious with the remaining one. As a group, they are inferior to result I. Since mIoU and EA-score consider only the accuracy of each individual line, they do not tell the difference between these two results and provide only marginally different scores. In contrast, HIoU quantifies the superiority of result I correctly.
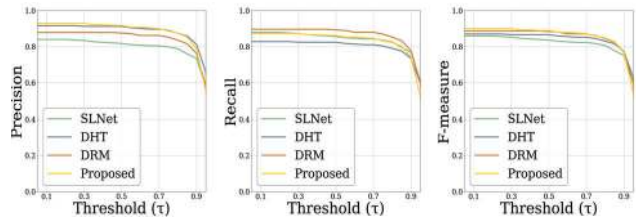

Figure 8: Comparison of the precision, recall, and F-measure curves in terms of the threshold τ on the SEL dataset. The mIoU metric is used.

than the existing metrics do, by considering the harmony among the detected lines.

4.3. Comparative assessment

We compare semantic line detection results of the proposed algorithm with those of the conventional SLNet [19], DHT [10], and DRM [16].

Comparison on SEL: Figure 8 compares the precision, recall, and F-measure curves of the proposed algorithm and the conventional algorithms on the SEL dataset. Table 1 reports the AUC performances of these curves. The proposed algorithm provides a poorer recall but a better precision than the conventional algorithms. F-measure is the harmonic mean of recall and precision. Note that the proposed algorithm outperforms all conventional algorithms in terms of F-measure and HIoU.

Comparison on SEL_Hard: Table 1 also compares the results on SEL_Hard. For this comparison as well, we use the same algorithms that are trained using the training images in the SEL dataset. As mentioned previously, SEL_Hard images are much more complicated than SEL images. Also, many of SEL images contain only one semantic line. Thus,

division of the image is consistent with the division by the ground-truth. Suppose that the set of detected lines and the set of ground-truth lines divide the image into regions $S = \{s_1, s_2, \ldots, s_N\}$ and $T = \{t_1, t_2, \ldots, t_M\}$, respectively. Then, we define HIoU as

$$\text{HIoU} = \frac{\sum_{i=1}^{N} \max_k \text{IoU}(s_i, t_k) + \sum_{j=1}^{M} \max_k \text{IoU}(t_j, s_k)}{N+M}. \quad (13)$$

In other words, for each s_i, we find the matching t_k and measure their IoU. Similarly, for each t_j, we find its IoU with the matching s_k. Then, the average of these bi-directional matching IoU's becomes the HIoU score. Figure 6(c) illustrates how to compute an HIoU score. Figure 7 shows that HIoU assesses detected lines more reasonably
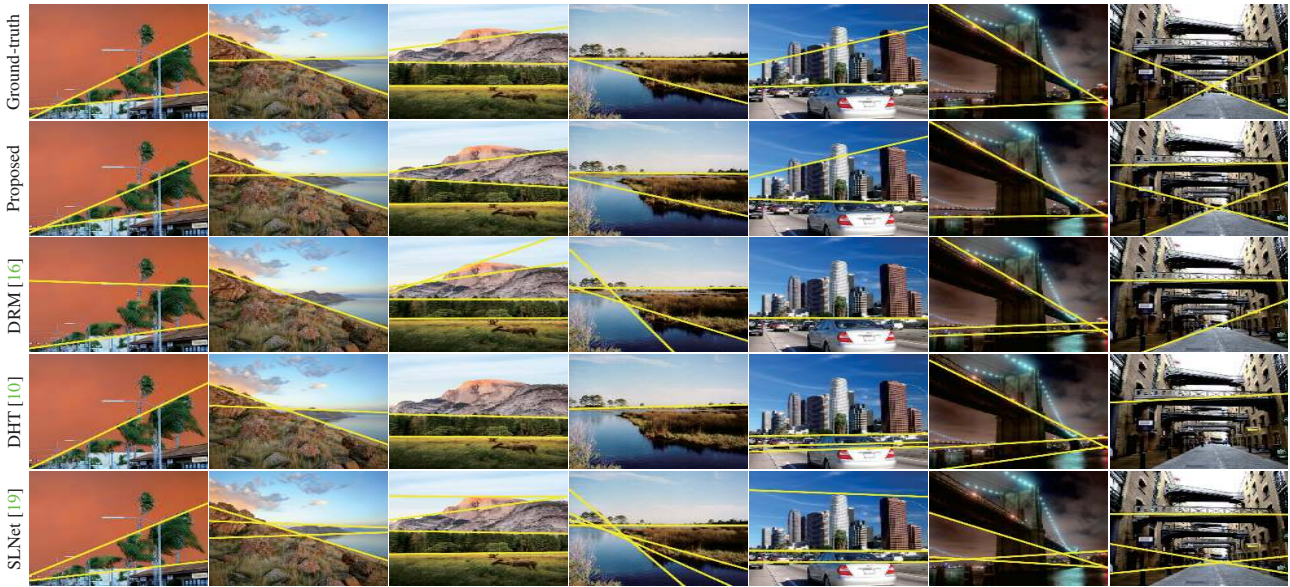
Figure 9: Comparison of semantic line detection results. The left three images are from SEL, and the others from SEL_Hard.

Table 2: Comparison of the EA-scores (Precision, Recall, F-measure) on the SL5K dataset.

	Precision	Recall	F-measure	HIoU
Zhao *et al.* [33]	70.3	74.5	72.3	-
Proposed	**79.4**	**81.4**	**80.3**	74.1

it is challenging to use only SEL images to learn the harmony between lines in more complicated SEL_Hard images. Nevertheless, the proposed algorithm yields competitive results to DRM, which performs the best but demands a too high computational cost. Note that the proposed algorithm is about 20 times faster than DRM. Moreover, the proposed algorithm outperforms DRM in terms of AUC_P.

Figure 9 compares detection results on the SEL and SEL_Hard datasets. The conventional algorithms detect redundant lines near object boundaries or fail to detect implied semantic lines. In contrast, the proposed algorithm detects implied as well as obvious semantic lines more reliably, while ensuring the harmony between detected lines.

Comparison on SL5K: Table 2 compares the performances on the SL5K dataset. Zhao *et al.* [33] report the performances of their algorithm in the EA-score metric only, and their training codes or model parameters are not available. Thus, we compare the results in the EA-score metric only, as done in [33]. We see that the proposed algorithm outperforms Zhao *et al.* by significant margins 9.1, 6.9, and 8.0 in terms of precision, recall, and F-measure, respectively. Also, the proposed algorithm yields the HIoU score of 74.1. Figure 10 shows some detection results.
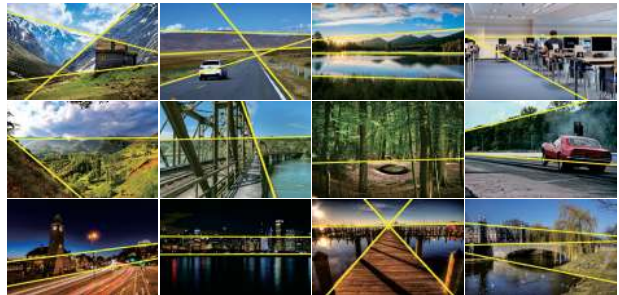

Figure 10: Detection results of the proposed algorithm on the SL5K dataset.

Comparison on CULane: We compare the proposed algorithm with the conventional road lane detectors [14, 25] on the 'no lane' category in CULane, in which lanes are implicit or invisible. Conventional techniques are based on the segmentation framework and the ground-truth is also given as a binary mask for each lane. Thus, for comparison, we declare the most overlapping line with the segmentation mask of each lane as a semantic line. The experimental settings are described in detail in the supplemental document. Figure 11 shows some ground-truth semantic lines and compares their detection results. Although the lines are extremely unobvious, the proposed algorithm detects them more reliably than the conventional detectors. Table 3 compares the AUC and HIoU scores. Note that, unlike the conventional detectors, the proposed algorithm does not use the information of the maximum number of lanes in a scene. The conventional algorithms poorly recall implied or invisible lanes. The proposed algorithm is slightly less pre-
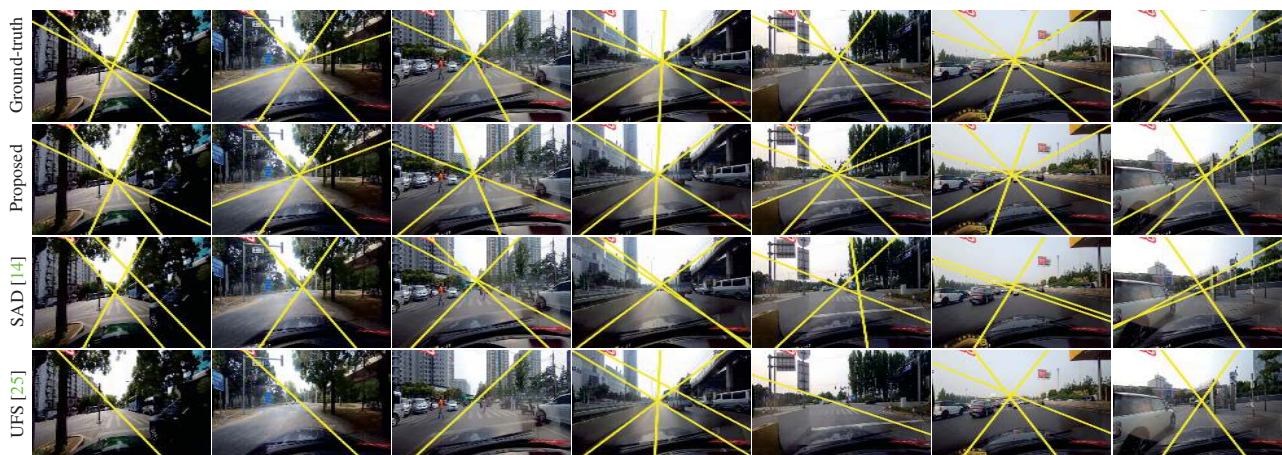
Figure 11: Comparison of semantic line detection results on the CULane dataset ('no lane' category).

Table 3: Comparison of the AUC and HIoU scores (%) on the 'no lane' category in the CULane dataset.

	AUC_P	AUC_R	AUC_F	HIoU
UFS [25]	93.00	83.47	87.98	72.68
SAD [14]	**93.64**	84.20	88.67	74.77
Proposed	92.43	**91.66**	**92.04**	**76.46**

Table 4: Ablation studies for the S-Net, H-Net, and MWCS process on the SEL dataset.

		AUC_F	HIoU
I.	S-Net	77.75	69.03
II.	S-Net+H-Net+MWCS(w/o IRC)	84.66	79.14
III.	S-Net+H-Net+MWCS(w/o offset)	86.60	80.33
IV.	S-Net+H-Net+MWCS	86.83	81.03

cise, but provides significantly higher recall and F-measure scores than the conventional detectors. Also, the proposed algorithm yields a better HIoU score than the conventional detectors, by exploiting the harmonious property of road lanes, such as parallelness and equal width between adjacent lanes.

Running time analysis: Table 1 also compares the running times. We use a PC with Intel Core i5-8500 CPU and NVIDIA RTX 2080 ti GPU. Note that SLNet and DRM require a lot of time to extract discriminative line features. Especially, DRM is the slowest method at 1.05 fps, because its mirror attention module and iterative ranking-and-matching process are too demanding. The proposed algorithm and DHT are much faster. Although DHT is the fastest, its recall performance is not competitive.

4.4. Ablation studies

We conduct ablation studies to analyze the efficacy of the proposed S-Net, H-Net, and MWCS process on the SEL

dataset. Table 4 compares several ablated methods. Method I uses S-Net only to detect semantic lines, in which the selection-and-removal process is performed iteratively until the maximum probability becomes lower than 0.5. Method II uses H-Net and the MWCS process as well, but H-Net is trained without employing the IRC module. In Method III, line offsets are not used to refine detection results. Method I is significantly inferior to the other methods, indicating that both H-Net and MWCS are essential for detecting harmonious semantic lines. Also, by comparing II with IV, we see that the inter-region correlation feature is effective for estimating the harmony between two lines. Also, from III with IV, note that the performance is improved by refining detected lines using regression offsets.

5. Conclusions

We proposed a novel semantic line detector. First, we developed S-Net to compute the line probabilities and offsets of line candidates. Second, we filtered out irrelevant lines through a selection-and-removal process. Third, we constructed a complete graph, whose edge weights were computed by H-Net. Finally, we determined a maximal weight clique representing a group of harmonious semantic lines. Also, to assess the overall harmony of detected lines, we proposed a novel metric called HIoU. It was experimentally demonstrated that the proposed algorithm can detect harmonious semantic lines effectively and efficiently.

Acknowledgements

References

[1] Cuneyt Akinlar and Cihan Topal. EDLines: A real-time line segment detector with a false detection control. *Pattern Recog. Lett.*, 32(13):1633–1642, 2011.

[2] Mohamed Aly. Real time detection of lane markers in urban streets. In *IEEE Intelligent Vehicles Symposium*, pages 7–12. IEEE, 2008.

[3] Gary Chartrand and Ping Zhang. *Chromatic Graph Theory*. CRC press, 2019.

[4] Agnès Desolneux, Lionel Moisan, and Jean-Michel Morel. Meaningful alignments. *Int. J. Comput. Vis.*, 40(1):7–23, 2000.

[5] Raul Diaz and Amit Marathe. Soft labels for ordinal regression. In *Proc. IEEE CVPR*, 2019.

[6] Corinne Feremans, Martine Labbé, and Gilbert Laporte. Generalized network design problems. *European Journal of Operational Research*, 148(1):1–13, 2003.

[7] Michael Freeman. *The Photographer's Eye: Composition and Design for Better Digital Photos*. Focal Press, 2007.

[8] Jonathan Gross and Jay Yellen. *Graph Theory and Its Applications*. CRC Press, 2007.

[9] Chunzhao Guo, Takayuki Yamabe, and Seiichi Mita. Robust road boundary estimation for intelligent vehicles in challenging scenarios based on a semantic graph. In *Intelligent Vehicles Symposium*. IEEE, 2012.

[10] Qi Han, Kai Zhao, Jun Xu, and Ming-Ming Cheng. Deep Hough transform for semantic line detection. In *Proc. ECCV*, 2020.

[11] Yinghua He, Hong Wang, and Bo Zhang. Color-based road detection in urban traffic scenes. *IEEE Transactions on Intelligent Transportation Systems*, 5(4):309–318, 2004.

[12] Aharon Bar Hillel, Ronen Lerner, Dan Levi, and Guy Raz. Recent progresss in road and lane detection: A survey. *Mach. Vis. Appl.*, 25(3):727–745, 2014.

[13] Yuenan Hou, Zheng Ma, Chunxiao Liu, Tak-Wai Hui, and Chen Change Loy. Inter-region affinity distillation for road marking segmentation. In *Proc. IEEE CVPR*, 2020.

[14] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning lightweight lane detection CNNs by self attention distillation. In *Proc. IEEE ICCV*, 2019.

[15] 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. IEEE CVPR*, 2018.

[16] Dongkwon Jin, Jun-Tae Lee, and Chang-Su Kim. Semantic line detection using mirror attention and comparative ranking and matching. In *Proc. ECCV*, 2020.

[17] Nahum Kiryati, Yuval Eldar, and Alfred M Bruckstein. A probabilistic Hough transform. *Pattern Recog.*, 24(4):303–316, 1991.

[18] Bert Krages. *Photography: The Art of Composition*. Simon and Schuster, 2012.

[19] Jun-Tae Lee, Han-Ul Kim, Chul Lee, and Chang-Su Kim. Semantic line detection and its applications. In *Proc. IEEE ICCV*, 2017.

[20] Jun-Tae Lee, Han-Ul Kim, Chul Lee, and Chang-Su Kim. Photographic composition classification and dominant geometric element detection for outdoor scenes. *J. Vis. Commun. Image Represent.*, 55:91–105, 2018.

[21] Seokju Lee, Junsik Kim, Jae Shin Yoon, Seunghak Shin, Oleksandr Bailo, Namil Kim, Tae-Hee Lee, Hyun Seok Hong, Seung-Hoon Han, and In So Kweon. VPGNet: Vanishing point guided network for lane and road marking detection and recognition. In *Proc. IEEE ICCV*, 2017.

[22] Yancong Lin, Silvia L Pintea, and Jan C van Gemert. Deep Hough-transform line priors. In *Proc. ECCV*, 2020.

[23] Jiri Matas, Charles Galambos, and Josef Kittler. Robust detection of lines using the progressive probabilistic Hough transform. *Comput. Vis. Image Understand.*, 78(1):119–137, 2000.

[24] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial As Deep: Spatial CNN for traffic scene understanding. In *Proc. AAAI*, 2018.

[25] Zequn Qin, Huanyu Wang, and Xi Li. Ultra fast structure-aware deep lane detection. In *Proc. ECCV*, 2020.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015.

[27] Yi Sun, Xushen Han, and Kai Sun. Sem-LSD: A learning-based semantic line segment detector. *arXiv preprint arXiv:1909.06591*, 2019.

[28] Rafael Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. LSD: A fast line segment detector with a false detection control. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(4):722–732, 2008.

[29] Scott Workman, Menghua Zhai, and Nathan Jacobs. Horizon lines in the wild. In *Proc. BMVC*, 2016.

[30] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proc. IEEE ICCV*, 2015.

[31] Nan Xue, Song Bai, Fudong Wang, Gui-Song Xia, Tianfu Wu, and Liangpei Zhang. Learning attraction field representation for robust line segment detection. In *Proc. IEEE CVPR*, 2019.

[32] Menghua Zhai, Scott Workman, and Nathan Jacobs. Detecting vanishing points using global image context in a non-manhattan world. In *Proc. IEEE CVPR*, 2016.

[33] Kai Zhao, Qi Han, Chang-Bin Zhao Zhang, Jun Xu, and Ming-Ming Cheng. Deep Hough transform for semantic line detection. *arXiv preprint arXiv:2003.04676*, 2020.

[34] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE2020K dataset. In *Proc. IEEE CVPR*, 2017.

[35] Shengyan Zhou, Yanhua Jiang, Junqiang Xi, Jianwei Gong, Guangming Xiong, and Huiyan Chen. A novel lane detection based on geometrical model and gabor filter. In *IEEE Intelligent Vehicles Symposium*, pages 59–64. IEEE, 2010.

[36] Yichao Zhou, Haozhi Qi, Jingwei Huang, and Yi Ma. NeurVPS: Neural vanishing point scanning via conic convolution. In *Proc. NIPS*, 2019.

[37] Yichao Zhou, Haozhi Qi, and Yi Ma. End-to-end wireframe parsing. In *Proc. IEEE ICCV*, 2019.