Semantic Line Combination Detector

Jinwon Ko
Korea University
jwko@mcl.korea.ac.kr

Dongkwon Jin
Korea University
dongkwonjin@mcl.korea.ac.kr

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

Abstract

A novel algorithm, called semantic line combination detector (SLCD), to find an optimal combination of semantic lines is proposed in this paper. It processes all lines in each line combination at once to assess the overall harmony of the lines. First, we generate various line combinations from reliable lines. Second, we estimate the score of each line combination and determine the best one. Experimental results demonstrate that the proposed SLCD outperforms existing semantic line detectors on various datasets. Moreover, it is shown that SLCD can be applied effectively to three vision tasks of vanishing point detection, symmetry axis detection, and composition-based image retrieval. Our codes are available at https://github.com/Jinwon-Ko/SLCD.

1. Introduction

A *semantic line* [15] is defined as a meaningful line separating distinct semantic regions in an image. Besides this unary definition, multiple semantic lines in an image are supposed to convey the global scene structure properly [13]. It is challenging to detect such semantic lines because they are often implied by complicated region boundaries. Moreover, they should represent the image composition optimally by dividing it into semantic regions harmoniously.

Semantic lines are essential elements in many vision applications. For example, a horizon line [14, 30, 34, 40], which is a specific type of semantic line, can be exploited to adjust the levelness of an image [15, 29]. A reflection symmetry axis [3, 4, 6, 7, 20], which is another type of semantic line, provides visual cues for object recognition and pattern analysis. Vanishing points, conveying depth impression in images, can be estimated by detecting dominant parallel semantic lines in the 3D world [1, 12, 39]. In autonomous driving systems [13, 26, 36], boundaries of road lanes can be also described by semantic lines.

Recently, several attempts [9, 12, 13, 15, 35] have been made to detect semantic lines. These techniques perform line detection and refinement sequentially. At the detection stage, they extract deep line features to classify and regress each line candidate. At the refinement stage, reliable semantic lines are determined by removing redundant lines. Specifically, to refine line candidates, non-maximum suppression (NMS) is performed in [9, 12, 15], as illustrated in Figure 1(a). Lee *et al*. [15] iteratively select the reliable line near boundary pixels and remove overlapping lines with the selected one. Han *et al*. [9] simplify the NMS process by adopting a Hough line space. Jin *et al*. [12] process each candidate through comparative ranking and matching. These techniques, however, do not consider the overall harmony among detected lines. To cope with this issue, Jin *et al*. [13] estimate the relation score for every pair of detected lines and then decide harmonious semantic lines via graph optimization. But, they may yield sub-optimal results, since only the pairwise relationships between lines are exploited, as in Figure 1(b).

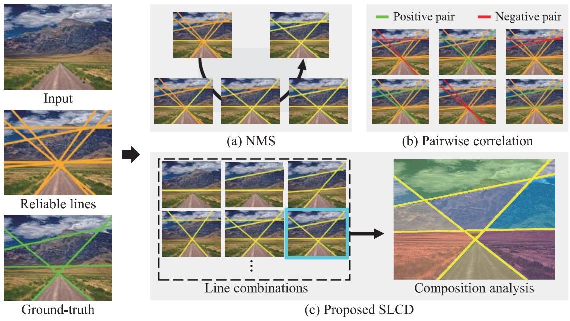
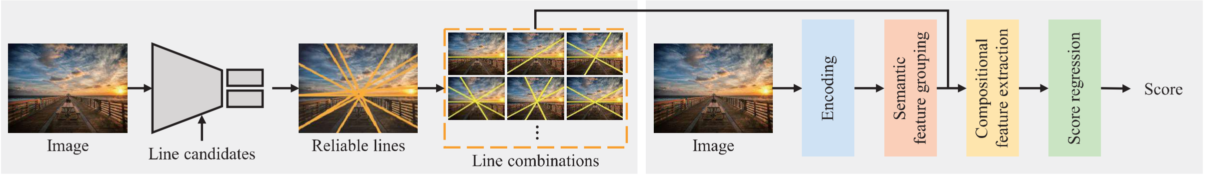


Figure 1. After selecting reliable line candidates, there are two existing approaches to semantic line detection. The first approach in (a) focuses on locating a line near a region boundary and eliminating overlapping lines. However, a redundant line still remains, since this approach does not consider how well a group of detected lines represents the layout of a scene. The second approach in (b) takes into account only the pairwise correlation between two lines, so it may fail to assess the overall harmony of more than two semantic lines. In contrast, in (c), the proposed SLCD generates a number of line combinations, analyzes all lines in each combination at once, and then finds the most harmonious combination that conveys the global scene composition optimally.

[*] Corresponding author.



(a) Line combination generation (b) Line combination assessment

Figure 2. Overview of the proposed SLCD algorithm.

In this paper, we propose a novel algorithm, called semantic line combination detector (SLCD), to find an optimal group of semantic lines. It processes all lines in a line group (or combination) simultaneously, instead of analyzing each pair of lines, to estimate the overall harmony, as in Figure 1(c). Figure 2 shows an overview of the proposed SLCD. First, we select reliable lines from line candidates and then generate a number of line combinations. Second, we score all line combinations and determine the combination with the highest score as the optimal group of semantic lines. To this end, we design two novel modules for semantic feature grouping and compositional feature extraction. We also introduce a novel loss function to guide the feature grouping module. Experimental results demonstrate that SLCD can detect semantic lines reliably on existing datasets and a new dataset, called compositionally diverse lines (CDL). Moreover, it is shown that SLCD can be used effectively in various applications.

This work has the following major contributions:

- SLCD finds an optimal combination of semantic lines by processing all lines in a line combination at once.
- We construct the CDL dataset containing compositionally diverse images with implied lines. It will be made publicly available.[1]
- SLCD outperforms conventional detectors on most datasets. Also, its effectiveness is demonstrated in three applications: vanishing point detection, symmetry axis detection, and composition-based image retrieval.

2. Related Work

Semantic lines, located near the boundaries of different semantic regions, outline the layout and composition of an image. They play an important role in various vision applications. Horizons [14, 30, 34, 40] are a specific type of semantic lines, which can be applied to adjust the levelness of images and improve their aesthetics [15, 29]. In [1, 12, 39], dominant parallel lines in the 3D world are detected to estimate vanishing points, which convey depth impression on 2D images. Also, in [3, 6, 7, 20], reflection symmetry axes are identified to analyze the shapes of objects or patterns. These types of lines can be regarded as highly implied semantic lines. In [13, 26, 36], straight lanes are detected to aid in vehicle maneuvers in road environments. Furthermore, semantic lines are essential visual cues in photographic composition [16, 18]. They direct viewers' attention and help to compose a visually balanced image.

Several semantic line detectors [9, 12, 13, 15] have been proposed. They perform in two stages: line detection and refinement. At the detection stage, deep line features are extracted to classify and regress each line candidate. At the refinement stage, reliable semantic lines are determined by removing irrelevant candidates, based on NMS [9, 12, 15] or graph optimization [13]. More specifically, Lee *et al.* [15] detect reliable lines with classification probabilities higher than a threshold. They then iteratively select semantic lines and remove overlapping lines with the selected one, by employing the edge detector in [31]. Han *et al.* [9] predict the probability of each candidate in a Hough parametric space. They simplify NMS by computing the centroids of connected components in the Hough space. Jin *et al.* [12] design two comparators to estimate the priority and similarity between two lines. Then, they determine the most reliable lines and eliminate redundant ones alternately through pairwise comparisons. However, these techniques [9, 12, 15] do not consider how well a group of detected lines represents the global scene structure. To address this issue, Jin *et al.* [13] analyze the pairwise harmony of detected lines. They first estimate a harmony score for a pair of detected lines. They then construct a complete graph and determine harmonious semantic lines by finding a maximal weight clique. Their method, however, may yield sub-optimal results because it considers only pairwise relationships between lines. On the contrary, the proposed SLCD finds an optimal combination of semantic lines by analyzing all lines in each combination simultaneously.

3. Proposed Algorithm

We propose a novel algorithm, called SLCD, to detect an optimal combination of semantic lines, an overview of which is in Figure 2. First, we select K reliable lines from line candidates and then generate a number of line combinations. Second, we score all the line combinations and determine the combination with the highest score as the optimal group of semantic lines.

[1] CDL is available at https://github.com/Jinwon-Ko/SLCD.

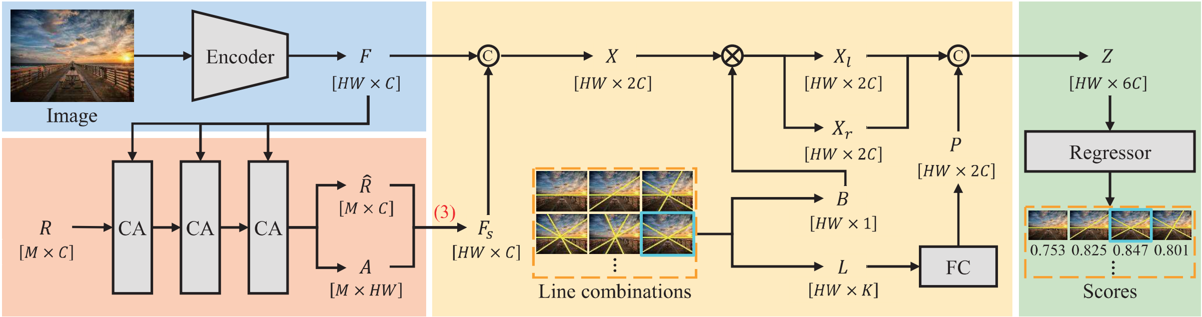


Figure 3. The architecture of SLCD. CA and FC denote cross-attention and fully connected layers, respectively. The blue, red, yellow, and green boxes indicate encoding, semantic feature grouping, compositional feature extraction, and score regression, respectively.

3.1. Generating Line Combinations

Initializing line candidates: We generate line candidates, which are end-to-end straight lines in an image. Each line candidate is parameterized by polar coordinates in the Hough space [9]. By quantizing the coordinates uniformly, we obtain N line candidates. The default N is 1024.

Filtering line candidates: Given the N line candidates, there are 2^N possible line combinations in total. Since this number of combinations is unmanageable, we filter out irrelevant lines to maintain K reliable ones only. To this end, we design a simple line detector by modifying S-Net in [13]. Given an image, the detector obtains a convolutional feature map and extracts line features by aggregating the features of pixels along each line candidate. Then, it computes the classification probability and regression offset of each candidate. We select the most reliable candidate with the highest probability and remove overlapping lines with the selected one. We iterate this process K times to determine K reliable lines. The default K is 8. Figure 4 shows examples of selected reliable lines. Note that, in this filtering, false negatives occur rarely because the K lines are selected via NMS even though their classification probabilities are not high. The architecture of the modified S-Net is described in detail in the supplemental document.

Generating line combinations: From the K reliable lines, we generate all 2^K line combinations.

3.2. Assessing Line Combinations

When a line combination divides an image insufficiently or over-segments it into unnecessary parts, it does not describe the overall structure of the scene properly. On the contrary, an optimal combination of semantic lines should convey the image composition reliably and efficiently (*i.e.* with a small number of lines). To find the best line combination, we develop the semantic line combination detector (SLCD). Figure 3 shows the structure of SLCD, which performs encoding, semantic feature grouping, compositional feature extraction, and score regression. The detailed architecture of SLCD is described in the supplemental document.

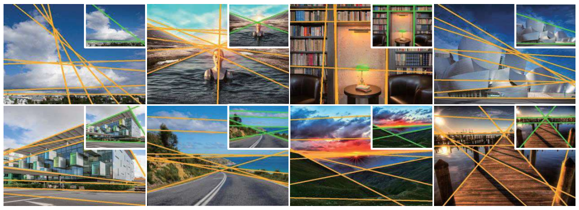


Figure 4. From a set of line candidates, the line detector selects K reliable lines, depicted in orange. A high recall rate is achieved because a sufficient number of reliable lines are selected through NMS. Ground-truth semantic lines in green are in the insets.

Encoding: Given an image, we extract multi-scale feature maps of ResNet50 [11]. Then, we match their spatial resolutions via bilinear interpolation and concatenate them in the channel dimension. We then squeeze the channels using 2D convolutional layers to obtain an aggregated feature map $F \in \mathbb{R}^{HW \times C}$, where H, W, and C are the feature height, the feature width, and the number of channels.

Semantic feature grouping: Semantic lines are located near the boundaries between distinct regions. For their reliable detection, it is desirable to separate different regional parts more clearly. We hence attempt to group pixel features into multiple regions through cross-attention [2, 22, 32, 33]. We employ three cross-attention modules.

Let $R \in \mathbb{R}^{M \times C}$ be a learnable matrix representing M regions, called region query matrix. Then, we convert R into queries and F into keys and values by

$$R_q = RU_q, \quad F_k = (F + S)U_k, \quad F_v = (F + S)U_v, \quad (1)$$

where $S \in \mathbb{R}^{HW \times C}$ denotes the sinusoidal positional encoding [28], and $U_q, U_k, U_v \in \mathbb{R}^{C \times C}$ are projection matrices for queries, keys, and values. We then obtain an updated

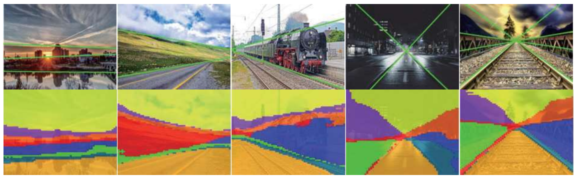


Figure 5. Visualization of semantic feature grouping results. The top row shows input images with ground-truth lines. The bottom one presents the membership maps, representing the semantic region that each pixel belongs to.

region query matrix $\hat{R}$ by

$$A = \underset{M}{\mathrm{softmax}}(R_q F_k^{\mathrm{T}} / \tau), \quad \hat{R} = AF_v + R, \tag{2}$$

where $A \in \mathbb{R}^{M \times HW}$ is the attention matrix with a scaling factor τ, and the letter 'M' means the softmax operation is done in the column direction, as in [22, 32, 33]. Then, in the last cross-attention, we generate a semantic feature map $F_s \in \mathbb{R}^{HW \times C}$ by

$$F_s = A^{\mathrm{T}} \hat{R}. \tag{3}$$

Finally, we channel-wise concatenate F and F_s to obtain a combined feature map $X \in \mathbb{R}^{HW \times 2C}$.

Note that, in (2), the softmax function is applied along the query axis. Thus, in $A = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{HW}]$, each column $\mathbf{p}_i \in \mathbb{R}^M$ is a probability vector. Specifically, the mth element p_i^m in $\mathbf{p}_i$ is the probability that pixel i belongs to the mth region query. Figure 5 visualizes membership maps, computed by

$$\left[\mathrm{argmax}(\mathbf{p}_1), \mathrm{argmax}(\mathbf{p}_2), \ldots, \mathrm{argmax}(\mathbf{p}_{HW}) \right] \tag{4}$$

where each element is the index of the region query that the corresponding pixel most likely belongs to. We see that each image is partitioned into M meaningful regions. Thus, F_s in (3) represents the regional membership of each pixel and is used to make the feature map X more discriminative.

To produce the attention matrix A reliably, we design a novel loss function in Section 3.3. The default M is 8.

Compositional feature extraction: For each line combination, we generate three types of feature maps to extract a compositional feature map. First, we generate a binary mask $B \in \mathbb{R}^{HW}$ for each line combination. Each element in B is 1 if the corresponding pixel belongs to any line in the combination, and 0 otherwise. Then, we decompose the combined feature into a line feature map X_l and a region feature map X_r by

$$X_l = X \otimes B, \quad X_r = X \otimes (1 - B), \tag{5}$$

where $\otimes$ is the element-wise multiplication. In other words, $X_l \in \mathbb{R}^{HW \times 2C}$ contains contextual information for the line pixels only, whereas $X_r \in \mathbb{R}^{HW \times 2C}$ does for the pixels strictly inside the regions divided by the lines.

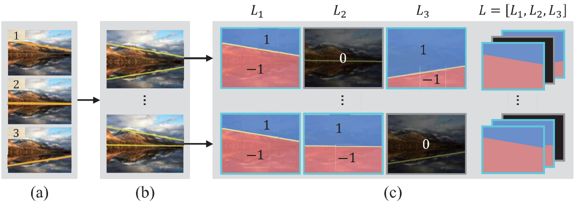


Figure 6. Illustration of the line collection map generation when $K = 3$: (a) reliable lines, (b) line combinations, and (c) line collection maps.

Moreover, we generate a line collection map $L = [L_1, L_2, \ldots, L_K] \in \mathbb{R}^{HW \times K}$, where $L_k \in \mathbb{R}^{HW}$ is a ternary mask for the kth reliable line, as illustrated in Figure 6. If the kth reliable line does not belong to the line combination, all elements in L_k are set to 0. Otherwise, each value in L_k is set to either 1 or -1 to indicate where the corresponding pixel is located between the two parts divided by the kth reliable line. In other words, L_k informs how the kth line splits the image into two regions. We then produce a positional feature map $P \in \mathbb{R}^{HW \times 2C}$ by applying a series of fully connected layers to L. Then, we yield a compositional feature map $Z \in \mathbb{R}^{HW \times 6C}$ by

$$Z = [X_l, X_r, P], \tag{6}$$

which contains information about how the lines in the combination separate the image into multiple parts.

Score regression: Lastly, using a regressor, we estimate a composition score for each line combination. Then, we declare the line combination with the highest score as the optimal group of semantic lines. The regressor takes the compositional feature map Z in (6) as input and predicts a composition score s within $[0, 1]$. It is implemented using a bilinear interpolation layer and a series of 2D convolution layers and fully connected layers with the ReLU activation.

3.3. Loss Functions

To train SLCD, we design two loss functions.

Semantic region separation loss: When two pixels i and j are located in distinct regions, they should be assigned to different region queries. In other words, their probability vectors $\mathbf{p}_i$ and $\mathbf{p}_j$ in the attention matrix A should be far from each other. Let X and Y be the two regions divided by a ground-truth line. Their probability vectors are defined as

$$\mathbf{p}_X = \frac{1}{|X|} \sum_{i \in X} \mathbf{p}_i, \qquad \mathbf{p}_Y = \frac{1}{|Y|} \sum_{j \in Y} \mathbf{p}_j, \tag{7}$$

which should be also far from each other, for the ground-truth line divides the image into two semantic regions.

To measure the distance between probability vectors, we adopt the Kullback-Leibler divergence (KLD) [5]. Then,

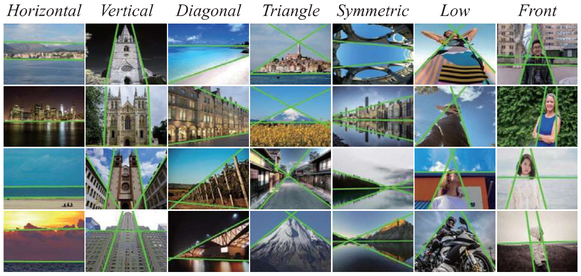


Figure 7. The proposed CDL dataset contains 9,100 images in seven composition classes: *Horizontal*, *Vertical*, *Diagonal*, *Triangle*, *Symmetric*, *Low*, and *Front*. The ground-truth semantic lines are depicted in green.

we define the semantic region segmentation (SRS) loss, by employing all ground-truth lines, as

$$\mathcal{L}_{\text{SRS}} = -\sum_{l=1}^{T} \left[D(\mathbf{p}_{X_l} \| \mathbf{p}_{Y_l}) + D(\mathbf{p}_{Y_l} \| \mathbf{p}_{X_l}) \right] \tag{8}$$

where D denotes the KLD, $\mathbf{p}_{X_l}$ and $\mathbf{p}_{Y_l}$ are the regions divided by the lth ground-truth line, and T is the number of ground-truth lines. Both $D(\mathbf{p}_{X_l} \parallel \mathbf{p}_{Y_l})$ and $D(\mathbf{p}_{Y_l} \parallel \mathbf{p}_{X_l})$ are computed because D is not commutative. With $\mathcal{L}_{\text{SRS}}$, it is possible to roughly segment an image into meaningful parts, as illustrated in Figure 5, even though no ground-truth labels for semantic segmentation are used for training.

Regression loss: We also design a loss for regressing the composition score of each line combination. To this end, we employ the HIoU metric [13] that quantifies the structural layout of a line combination. Let $\mathbf{c}$ and $\mathbf{c}^\star$ denote a line combination and the ground-truth combination, respectively. We compute the HIoU between $\mathbf{c}$ and $\mathbf{c}^\star$ and use it as the ground-truth composition score $\bar{s}$ of $\mathbf{c}$. Then, the regression loss is defined as

$$\mathcal{L}_{\text{reg}} = (s - \bar{s})^2 \tag{9}$$

where s is the predicted score of $\mathbf{c}$. To reduce $\mathcal{L}_{\text{reg}}$ effectively, a ranking loss $\mathcal{L}_{\text{rank}}$ is also employed as in [17].

4. Experimental Results

4.1. Implementation Details

We adopt ResNet50 [11] as the encoder of the proposed SLCD. We use the AdamW optimizer [19] with a learning rate 10^{-4}, a weight decay of 10^{-4}, $\gamma = 0.5$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. We use a batch size of two for 400,000 iterations. Training images are resized to 480 × 480 and augmented by random horizontal flipping. We fix the number of line candidates, reliable lines, and region queries to $N = 1024$, $K = 8$, and $M = 8$, respectively. We set $H = 60$, $W = 60$, and $C = 96$.

Table 1. Comparison of the HIoU scores (%) on the SEL, SEL_Hard, NKL, and CDL datasets.

	SEL	SEL_Hard	NKL	CDL
SLNet [15]	77.87	59.71	65.49	57.78
DHT [9]	79.62	63.39	69.08	63.24
DRM [12]	80.23	**68.83**	67.42	63.96
HSLD [13]	81.03	65.99	74.29	64.98
Proposed	**84.09**	68.15	**76.21**	**68.85**

4.2. Datasets

SEL [15]: It is the first dataset for semantic line detection. It contains 1,750 images, which are split into 1,575 training and 175 test images. Each semantic line is annotated by the coordinates of two endpoints on an image boundary. In SEL, most images are high-quality landscapes, in which semantic lines are often obvious.

SEL_Hard [12]: It contains 300 test images only, selected from the ADE20K segmentation dataset [37]. It is challenging because of cluttered scenes or occluded objects.

NKL [35]: It is a relatively large dataset of 5,200 training and 1,300 testing images. It includes both indoor and outdoor scenes.

CDL: We construct CDL to contain 7,100 scenes with diverse contents and compositions. It is split into 6,390 training and 710 test images. As in Figure 7, the images are categorized into seven composition classes: *Horizontal*, *Vertical*, *Diagonal*, *Triangle*, *Symmetric*, *Low*, and *Front*. In *Low* and *Front*, humans and animals are essential parts of the image composition. In the other classes, most images are outdoor ones, as in the other datasets. To construct CDL, semantic lines were manually annotated in about 200 man-hours. More examples and the annotation process are provided in the supplemental document.

4.3. Metrics

As mentioned earlier, the harmony of detected lines is more important than individual lines in semantic line detection. Therefore, the main paper discusses only HIoU performances. The precision, recall, and f-measure results, which are metrics used in previous work, are compared in the supplemental document.

HIoU: An optimal group of semantic lines conveys a harmonious impression about the composition of an image. To assess the overall harmony of detection results, we use the HIoU metric [13]. It measures the consistency between the division of an image by detected lines and that by the ground truth. Let $S = \{s_1, s_2, \ldots, s_N\}$ and $T = \{t_1, t_2, \ldots, t_M\}$ be the regions divided by the detected lines and the ground-truth lines, respectively. Then, HIoU is computed as

$$\text{HIoU} = \frac{\sum_{i=1}^{N} \max_k \text{IoU}(s_i, t_k) + \sum_{j=1}^{M} \max_k \text{IoU}(t_j, s_k)}{N+M}. \tag{10}$$

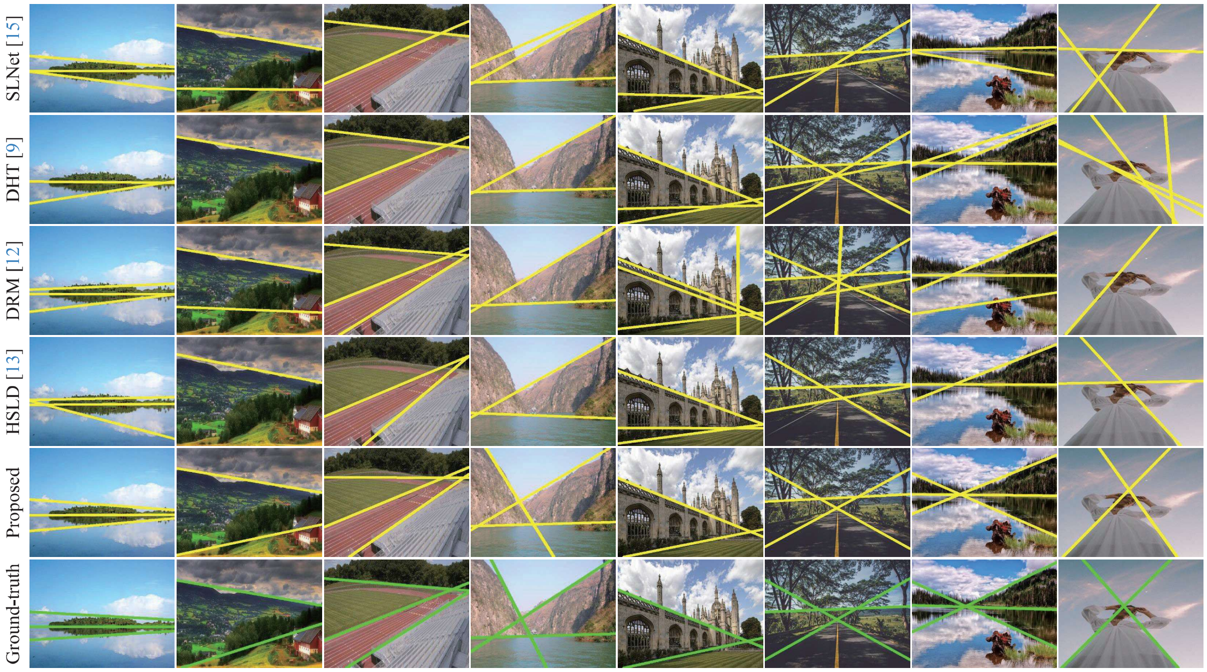


Figure 8. Comparison of detected semantic lines. From the left, two images are selected from each of the SEL, SEL_Hard, NKL, and CDL datasets.

Specifically, for each s_i, the most overlapping region t_k is chosen and the intersection-over-union (IoU) is calculated. This is performed similarly for each t_j. Then, the HIoU score is given by the average of these bi-directionally matching IoUs.

4.4. Comparative Assessment

Table 1 compares the HIoU scores of the proposed SLCD with those of the existing detectors [9, 12, 13, 15] on the SEL, SEL_Hard, NKL, and CDL datasets. Also, Figure 8 compares detection results qualitatively. The existing detectors miss correct lines or fail to remove redundant lines, yielding sub-optimal results. In contrast, SLCD detects semantic lines more precisely and represents the composition more reliably than the existing detectors do. More detection results are available in the supplemental document.

Comparison on SEL: In Table 1, SLCD outperforms the existing detectors on SEL. Compared with the second-best HSLD, SLCD yields a wide HIoU margin of 3.06. This indicates that SLCD finds an optimal group of semantic lines more effectively by processing all lines in each combination simultaneously, instead of performing pairwise comparisons in HSLD.

Comparison on SEL_Hard: As done in [12], we conduct experiments on SEL_Hard using the networks trained on the SEL dataset. In Table 1, SLCD ranks 2nd on SEL_Hard. DRM provides a better result than SLCD, but it demands much higher complexity, as will be discussed in Section 4.5.

Table 2. Comparison of the HIoU scores (%) on the CDL dataset according to the number M of region queries.

M	4	8	16	32
HIoU	66.43	68.85	68.11	67.07

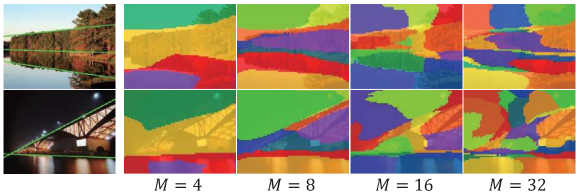


Figure 9. Visualization of the membership maps according to M.

Comparison on NKL: SLCD surpasses all existing detectors. For example, its HIoU score is 1.92 points higher than the second-best HSLD.

Comparison on CDL: Table 1 also lists HIoU scores on the proposed CDL dataset. For a fair comparison, we train the existing detectors on CDL using their publicly available source codes. We see that SLCD outperforms all existing detectors on CDL as well. SLNet, DHT, and DRM yield poor results because they do not consider the overall harmony of detected lines. HSLD is better than these detectors but is inferior to the proposed SLCD. We compare and

Table 3. Comparison of the HIoU scores (%) on the CDL dataset according to the usage of the SRS loss $\mathcal{L}_{\mathrm{SRS}}$.

	w/o $\mathcal{L}_{\mathrm{SRS}}$	Proposed
HIoU	66.71	68.85

Table 4. Ablation studies for the compositional feature extraction on the CDL dataset.

	Line	Region	Positional	HIoU
I	✓			65.12
II		✓		65.74
III	✓	✓		66.37
IV	✓	✓	✓	68.85

discuss the detection results according to the composition classes in the supplemental document.

4.5. Analysis

We conduct several ablation studies to analyze each component of the proposed SLCD.

The number of M: Table 2 lists the HIoU performances on the CDL dataset, according to the number M of region queries. At the default $M = 8$, SLCD achieves the best performance. When M is smaller or bigger than 8, the performance degrades due to under- or over-segmentation of semantic parts, respectively, as shown in Figure 9.

Efficacy of $\mathcal{L}_{\mathrm{SRS}}$: In Table 3, if the proposed SRS loss $\mathcal{L}_{\mathrm{SRS}}$ in (8) is excluded from the training, the performance drops by 2.14 points. It means that the composition analysis based on the SRS loss is essential for finding optimal line combinations.

Efficacy of compositional feature extraction: SLCD extracts the compositional feature map Z for each line combination, by processing the line feature map X_l, the region feature map X_r, and the positional feature map P in (6). In Table 4, Method I uses the line feature map only, while II uses the region feature map only. Method III utilizes both feature maps. Method IV, which is the proposed SLCD, uses all three maps.

Method I yields the worst result, for it uses only the contextual information near line pixels. Method II is slightly better than Method I, by exploiting the regional information. Using both line and region feature maps, III provides a better result. Moreover, IV further improves the performance significantly by utilizing the positional feature map. This is because the overall harmony is estimated more effectively, by combining the line structures in the positional feature map with scene contexts.

Runtime: Table 5 compares the runtimes of SLCD and the existing detectors in seconds per frame (spf). We use a PC with AMD Ryzen 9 3900X CPU and NVIDIA RTX 2080 GPU. The proposed SLCD takes 0.114spf, adding up 0.030spf and 0.074spf for generating and assessing line combinations, respectively. DHT is the fastest detector, but it is inferior to SLCD on all datasets. On the other hand, even though DRM performs better on SEL_Hard, it is about 8.4 times slower than SLCD.

Table 5. Runtime comparison of the proposed SLCD and the existing detectors. The processing times are reported in seconds per frame (spf).

SLNet [15]	DHT [9]	DRM [12]	HSLD [13]	Proposed
0.136	0.033	0.952	0.046	0.114

Table 6. Comparison of the AA scores (%) of VP detection.

	AA1°	AA2°	AA10°
Zhou *et al*. [39]	**18.5**	<u>33.0</u>	60.0
Jin *et al*. [12]	8.6	22.9	<u>68.3</u>
Proposed	<u>16.6</u>	**36.9**	**78.3**

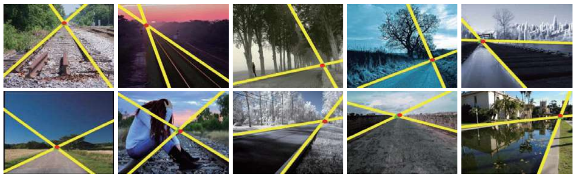


Figure 10. Detected dominant parallel lines and their intersections are depicted by yellow lines and green triangles, respectively. The ground-truth VPs are depicted by red dots.

5. Applications

We apply the proposed SLCD to three vision tasks: dominant vanishing point detection, reflection symmetry axis detection, and composition-based image retrieval. Note that the first two tasks were considered in [12]. Due to the page limit, more details and results are described in the supplemental document.

5.1. Dominant Vanishing Point Detection

A vanishing point (VP) facilitates understanding of the 3D geometric structure. We apply the proposed SLCD to detect dominant VPs, by identifying vanishing lines. We use the AVA landscape dataset [39], in which two dominant parallel lines are annotated for each image. To detect a dominant VP, we generate line combinations containing two lines only. We then find the best combination, whose intersecting point is declared as a VP. Table 6 compares this VP detection scheme with the existing line-based VP detectors in [12, 39]. The angle accuracies (AAs) [38] are compared. We see that SLCD is better than the existing detectors, except that its AA1° score is lower than that of Zhou *et al*. [39]. Figure 10 shows some VP detection results.

Table 7. Comparison of the AUC_A scores (%) of symmetry axis detection.

	ICCV	NYU	SYM_Hard
Cicconet *et al*. [3]	80.80	82.85	68.99
Elawady *et al*. [6]	87.24	83.83	73.90
Cicconet *et al*. [4]	87.38	87.64	81.04
Loy & Eklundh [20]	89.77	90.85	81.99
Jin *et al*. [12]	90.60	92.78	84.73
Proposed	**93.15**	**93.51**	**88.03**

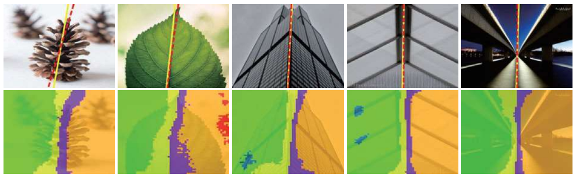


Figure 11. Symmetry axis detection results. In the top row, the ground-truth and predicted axes are depicted by dashed red and solid yellow lines, respectively. The membership maps are also visualized in the bottom row.

5.2. Reflection Symmetry Axis Detection

Reflection symmetry is a visual cue for analyzing object shapes and patterns. Its detection is challenging because the symmetry is only implicit in many cases. We test the proposed SLCD on three datasets: ICCV [7], NYU [4], and SYM_Hard [12]. In these datasets, each image contains a single reflection symmetry axis. Thus, in the proposed SLCD, each of the K reliable lines is regarded as a line combination. Table 7 compares the AUC_A scores [15] of the proposed SLCD and the conventional techniques [3, 4, 6, 12, 20]. SLCD outperforms all these techniques on all datasets. Figure 11 shows some detection results together with the membership maps. We see that regional parts are symmetrically divided along implied axes, enabling SLCD to identify those axes effectively.

5.3. Composition-Based Image Retrieval

Existing image retrieval techniques focus on the visual contents of a query image to find similar images in a database [8, 21, 24, 25]. In this work, however, we attempt to discover images with similar compositions to a query image, *i.e*. composition-based image retrieval.

We test the proposed SLCD on the Oxford 5k and Paris 6k retrieval dataset [23]. We first detect semantic lines in every image, while storing the positional feature map P in (6). We filter out some images whose composition scores are lower than a threshold since a low score indicates a low-quality image with inharmonious composition in general. Then, for a randomly selected query image, we compute the ℓ_2-distances between the positional feature maps P of the query and the remaining images. We determine the images with the smallest distances as retrieval results.

Figure 12 shows the top-4 retrieval results for four query images. We see that SLCD returns structurally similar images to the queries. This means that the positional feature maps P, containing the structural information of semantic lines, can be used to compute the compositional differences. Based on this observation, we perform data clustering by employing k-means [10] on the feature space of P. Figure 13 is t-SNE visualization [27] of the clustering results. The nine images nearest to each centroid are also shown. Note that images with similar composition are grouped into the same cluster, confirming the efficacy of SLCD in the compositional analysis of images.

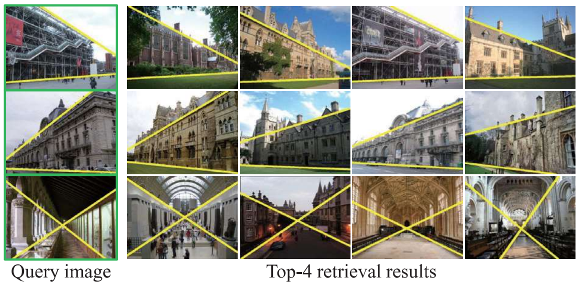


Figure 12. Composition-based retrieval results for query images.

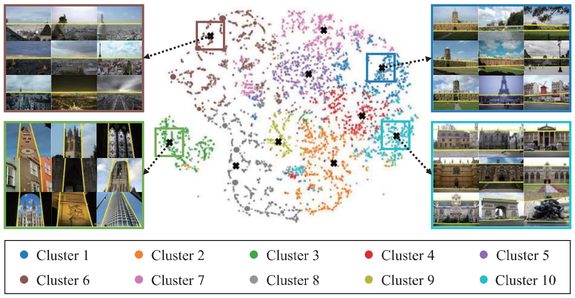


Figure 13. t-SNE visualization [27] of the feature space for the Oxford 5k and Paris 6k dataset [23].

6. Conclusions

We proposed a novel semantic line detector, SLCD, which processes a combination of lines at once to estimate the overall harmony reliably. We first generated all possible line combinations from reliable lines. Then, we estimated the score of each line combination and determined the best combination. Experimental results demonstrated that the proposed SLCD can detect semantic lines reliably on existing datasets, as well as on the new dataset CDL. Furthermore, SLCD can be successfully used in vanishing point detection, symmetry axis detection, and image retrieval.

Acknowledgements

This work was conducted by Center for Applied Research in Artificial Intelligence (CARAI) grant funded by DAPA and ADD (UD230017TD) and was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (No. NRF-2021R1A4A1031864 and No. NRF-2022R1A2B5B03002310).

References

[1] Jean-Charles Bazin, Yongduek Seo, Cédric Demonceaux, Pascal Vasseur, Katsushi Ikeuchi, Inso Kweon, and Marc Pollefeys. Globally optimal line clustering and vanishing point estimation in manhattan world. In *Proc. IEEE CVPR*, 2012. 1, 2

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. ECCV*, 2020. 3

[3] Marcelo Cicconet, Vighnesh Birodkar, Mads Lund, Michael Werman, and Davi Geiger. A convolutional approach to reflection symmetry. *Pattern Recog. Lett.*, 95:44–50, 2017. 1, 2, 8

[4] Marcelo Cicconet, David GC Hildebrand, and Hunter Elliott. Finding mirror symmetry via registration and optimal symmetric pairwise assignment of curves: Algorithm and results. In *Proc. IEEE ICCV Workshops*, 2017. 1, 8

[5] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2nd edition, 2006. 4

[6] Mohamed Elawady, Christophe Ducottet, Olivier Alata, Cécile Barat, and Philippe Colantoni. Wavelet-based reflection symmetry detection via textural and color histograms. In *Proc. IEEE ICCV Workshops*, 2017. 1, 2, 8

[7] Christopher Funk, Seungkyu Lee, Martin R Oswald, Stavros Tsogkas, Wei Shen, Andrea Cohen, Sven Dickinson, and Yanxi Liu. 2017 ICCV Challenge: Detecting symmetry in the wild. In *Proc. IEEE ICCV*, 2017. 1, 2, 8

[8] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *Int. J. Comput. Vis.*, 124:237–254, 2017. 8

[9] Qi Han, Kai Zhao, Jun Xu, and Ming-Ming Cheng. Deep hough transform for semantic line detection. In *Proc. ECCV*, 2020. 1, 2, 3, 5, 6, 7

[10] J. A. Hartigan and M. A. Wong. A K-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979. 8

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE CVPR*, 2016. 3, 5

[12] Dongkwon Jin, Jun-Tae Lee, and Chang-Su Kim. Semantic line detection using mirror attention and comparative ranking and matching. In *Proc. ECCV*, 2020. 1, 2, 5, 6, 7, 8

[13] Dongkwon Jin, Wonhui Park, Seong-Gyun Jeong, and Chang-Su Kim. Harmonious semantic line detection via maximal weight clique selection. In *Proc. IEEE CVPR*, 2021. 1, 2, 3, 5, 6, 7

[14] Florian Kluger, Hanno Ackermann, Michael Ying Yang, and Bodo Rosenhahn. Temporally consistent horizon lines. In *Proc. IEEE ICRA*, 2020. 1, 2

[15] Jun-Tae Lee, Han-Ul Kim, Chul Lee, and Chang-Su Kim. Semantic line detection and its applications. In *Proc. IEEE ICCV*, 2017. 1, 2, 5, 6, 7, 8

[16] 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. 2

[17] Debang Li, Junge Zhang, Kaiqi Huang, and Ming-Hsuan Yang. Composing good shots by exploiting mutual relations. In *Proc. IEEE CVPR*, 2020. 5

[18] Ligang Liu, Renjie Chen, Lior Wolf, and Daniel Cohen-Or. Optimizing photo composition. *Computer Graphics Forum*, 29:469–478, 2010. 2

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[20] Gareth Loy and Jan-Olof Eklundh. Detecting symmetry and symmetric constellations of features. In *Proc. ECCV*, 2006. 1, 2, 8

[21] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proc. IEEE ICCV*, 2017. 8

[22] Luigi Piccinelli, Christos Sakaridis, and Fisher Yu. iDisc: Internal discretization for monocular depth estimation. In *Proc. IEEE CVPR*, 2023. 3, 4

[23] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting Oxford and Paris: Large-scale image retrieval benchmarking. In *Proc. IEEE CVPR*, 2018. 8

[24] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning CNN image retrieval with no human annotation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41:1655–1668, 2018. 8

[25] Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22:1349–1380, 2000. 8

[26] Lucas Tabelini, Rodrigo Berriel, Thiago M. Paixao, Claudine Badue, Alberto F. De Souza, and Thiago Oliveira-Santos. Keep your eyes on the lane: Real-time attention-guided lane detection. In *Proc. IEEE CVPR*, 2021. 1, 2

[27] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9 (11), 2008. 8

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. NeurIPS*, 2017. 3

[29] Yaoting Wang, Yongzhen Ke, Kai Wang, Jing Guo, and Shuai Yang. Spatial-invariant convolutional neural network for photographic composition prediction and automatic correction. *J. Vis. Commun. Image Represent.*, 90:103751, 2023. 1, 2

[30] Scott Workman, Menghua Zhai, and Nathan Jacobs. Horizon lines in the wild. In *Proc. BMVC*, 2016. 1, 2

[31] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proc. IEEE ICCV*, 2015. 2

[32] Qihang Yu, Huiyu Wang, Dahun Kim, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. CMT-DeepLab: Clustering mask transformers for panoptic segmentation. In *Proc. IEEE CVPR*, 2022. 3, 4

[33] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. *k*-means mask transformer. In *Proc. ECCV*, 2022. 3, 4

[34] Menghua Zhai, Scott Workman, and Nathan Jacobs. Detecting vanishing points using global image context in a non-manhattan world. In *Proc. IEEE CVPR*, 2016. 1, 2

[35] Kai Zhao, Qi Han, Chang-Bin Zhang, Jun Xu, and Ming-Ming Cheng. Deep hough transform for semantic line detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44:4793–4806, 2021. 1, 5

[36] Tu Zheng, Yifei Huang, Yang Liu, Wenjian Tang, Zheng Yang, Deng Cai, and Xiaofei He. CLRNet: Cross layer refinement network for lane detection. In *Proc. IEEE CVPR*, 2022. 1, 2

[37] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *Proc. IEEE CVPR*, 2017. 5

[38] Yichao Zhou, Haozhi Qi, Jingwei Huang, and Yi Ma. NeurVPS: Neural vanishing point scanning via conic convolution. In *Proc. NeurIPS*, 2019. 7

[39] Zihan Zhou, Farshid Farhat, and James Z Wang. Detecting dominant vanishing points in natural scenes with application to composition-sensitive image retrieval. *IEEE Trans. Multimedia*, 19:2651–2665, 2017. 1, 2, 7

[40] Rui Zhu, Xingyi Yang, Yannick Hold-Geoffroy, Federico Perazzi, Jonathan Eisenmann, Kalyan Sunkavalli, and Manmohan Chandraker. Single view metrology in the wild. In *Proc. ECCV*, 2020. 1, 2