Journal of Electronic Imaging

JElectronicImaging.org

LSM: perceptually accurate line segment merging

Naila Hamid
Nazar Khan

SPIE. IS&T imaging.org

Naila Hamid, Nazar Khan, "LSM: perceptually accurate line segment merging," *J. Electron. Imaging* **25**(6), 061620 (2016), doi: 10.1117/1.JEI.25.6.061620.

Downloaded From: https://www.spiedigitallibrary.org/journals/Journal-of-Electronic-Imaging on 12 Aug 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

Journal of Electronic Imaging 25(6), 061620 (Nov/Dec 2016)

LSM: perceptually accurate line segment merging

Naila Hamid* **and Nazar Khan**
Punjab University College of Information Technology, Computer Vision and Machine Learning Group, Old Campus, The Mall Road, Lahore, Pakistan

Abstract. Existing line segment detectors tend to break up perceptually distinct line segments into multiple segments. We propose an algorithm for merging such broken segments to recover the original perceptually accurate line segments. The algorithm proceeds by grouping line segments on the basis of angular and spatial proximity. Then those line segment pairs within each group that satisfy unique, adaptive mergeability criteria are successively merged to form a single line segment. This process is repeated until no more line segments can be merged. We also propose a method for quantitative comparison of line segment detection algorithms. Results on the York Urban dataset show that our merged line segments are closer to human-marked ground-truth line segments compared to state-of-the-art line segment detection algorithms. *© 2016 SPIE and IS&T* [DOI: 10.1117/1.JEI.25.6.061620]

Keywords: line segments; line segment detection; line detection; grouping; merging; spatial proximity; angular proximity; perception; Gestalt; perceptually accurate line segments; quantitative evaluation.

Paper 16379SS received May 3, 2016; accepted for publication Nov. 16, 2016; published online Dec. 22, 2016.

1 Introduction

Line segments play an important role in perception and analysis of images by providing information about object boundaries and scene geometry. High-level vision applications such as stereo matching,[1] three-dimensional (3-D) reconstruction,[2] and vanishing point estimation[3] use line segments as the initial low-level features. However, existing line segment detection techniques have three major weaknesses:

1. For single pixel thick line segments in images, they detect two segments instead of a single segment due to gradient change on both sides of each line segment.
2. They tend to break up a perceptually single line segment into multiple smaller line segments.
3. They break up line segments at intersections.

These weaknesses limit the applicability of line segment detection techniques to high-level applications. If line segment detectors start providing connected, perceptually accurate line segments, then higher-level computer vision techniques such as road detection[3] and single view reconstruction[2] can use the line segments directly without further postprocessing. Demonstration of visual inspection and some of the weaknesses of a line segment detector on a real world fence image are presented in Fig. 1. In this paper, we propose an algorithm that refines the output of an existing line segment detector to obtain perceptually accurate line segments. Numerical results show that the proposed algorithm not only reports a fewer number of line segments but also provides line segments that are closer to ground-truths segments marked according to human perception.

The remaining paper is divided into five sections. A brief overview of relevant line segment detection and grouping techniques is presented in this section. We present our algorithm in Sec. 2 followed by a technique for evaluating the quality of merged line segments using ground-truth markings in Sec. 3. Section 4 presents results on synthetic as well as real images. Finally, we conclude and present future directions in Sec. 5.

1.1 *Line Segment Detection Techniques*

The line segment detector (LSD) by Gioi et al.[4] detects line segments by growing regions on the basis of gradient similarity. Grown regions are passed through a validation step to decide whether they form a line segment or not. Being gradient based, LSD suffers from the usual weaknesses as enumerated earlier. In our work, these weaknesses are overcome by merging detected line segments on the basis of spatial and angular proximity.

The Edge Drawing Lines (EDLines) algorithm[5] is another technique for line segment detection in which line segments are extracted on the basis of edge segments corresponding to the object boundaries. A straightness criterion along with a validation step is applied on the edge segments to get straight line segments. EDLines claims to produce clean and connected chains of edge segments. However, it does not overcome the other two weaknesses of LSD: breaking up line segments at intersections and giving two line segments for a single line due to gradient change on both sides of the line segment.

1.2 *Line Segments Grouping and Merging Techniques*

Grouping and merging techniques can be loosely categorized as top-down and bottom-up. Top-down approaches make global grouping decisions, often based on the Hough transform method, while bottom-up methods make local pairwise decisions.

*Address all correspondence to: Naila Hamid, E-mail: naila.hamid@pucit.edu.pk

1017-9909/2016/$25.00 © 2016 SPIE and IS&T

Downloaded From: https://www.spiedigitallibrary.org/journals/Journal-of-Electronic-Imaging on 12 Aug 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

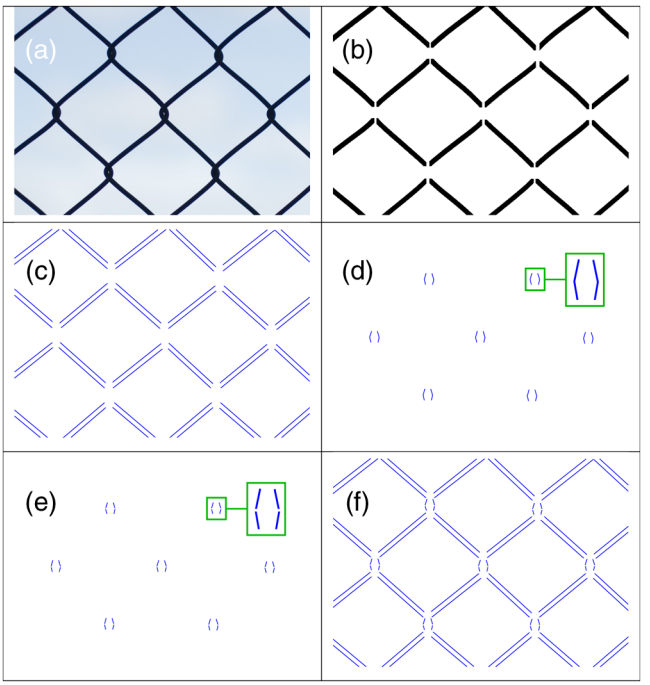

Fig. 1 Visual inspection and some of the weaknesses of an line segment detector: (a) a real world fence image. (b) The fence image contains 24 straight line segments that can be identified via visual inspection. (c) Line segment detectors will detect 48 segments for 24 visual line segments due to gradient change on both sides of each segment. (d) Similarly the seven intersections in the fence can be represented by 14 curved segments (two curved segments per intersection). (e) 14 curved segments yield 28 straight line segments (two segments per curved segment). (f) Therefore, a good line segment detector should detect 76 line segments on the fence image.

A line grouping and merging technique[6] clusters elementary line segments on the basis of angular proximity to obtain base line segments. The base line segments are then divided into individual line segments based on spatial proximity. While acceptable results are shown on synthetic images, they do not produce perceptually coherent line segments on real images. A similar idea is presented by Jang and Hong.[7]

Hough transform-based line segment detection does not consider connectivity of the segments while computing the votes. This problem is addressed by Guerreiro and Aguiar[8] by introducing connectivity in the voting process. This way, edge pixels vote only for the line segments to which they are spatially connected. This leads to detection of longer, connected, and therefore fewer line segments. It overcomes the weakness of gradient-based line segment detectors, such as LSD, that break up connected line segments. However, their method can still detect two line segments instead of a single line due to gradient change along both sides of the line segment.

A method for bottom-up perceptual grouping of line segment pairs is presented by Trucco and Verri.[9] Similar to our criteria, they make a grouping decision based on line segment lengths, interline distances, angles, and overlaps. They also bias the grouping toward longer line segments. However, they do not attempt to compute the merged line segment corresponding to a group. Our work is also different from traditional grouping strategies in that we go one step further than just determining constituents of the group and we actually merge them to obtain a single line segment that satisfies perceptual criteria. Details are presented in the next section.

Table 1 User-defined and adaptive thresholds and parameters.

(a) User-defined threshold and parameter.	
τ_θ	User-defined angular proximity threshold for grouping line segments on the basis of absolute angular difference [Eq. (1)].
ξ_s	User-defined spatial proximity parameter for computing adaptive spatial proximity threshold τ_s [Eq. (4)].
(b) Adaptive thresholds.	
τ_s	Adaptive spatial proximity threshold for grouping line segments on the basis of absolute axis-aligned distance [Eqs. (2) and (3)].
τ_θ^*	Adaptive angular proximity threshold for merging line segment pairs depending upon their angular difference, lengths, and closest distance [Eqs. (7) and (8)].

2 Methodology

Our algorithm takes as input an image and line segments detected by an off-the-shelf line segment detector. The merging pipeline has two main steps. In the first step, we group line segments based on traditional measures of spatial and angular proximity:

1. Spatial proximity: line segments must be spatially close enough to be grouped.
2. Angular proximity: orientation of line segments should not be much different from each other.

In the second step, we consider pairs of line segments within every group and merge them into a single line segment if they satisfy our mergeability criteria. These two steps are repeated until no more line segments can be merged.

In the following, we denote the set of detected line segments by $\mathcal{D}$ and the set of merged line segments by $\mathcal{L}$. For a pair of line segments (L_1, L_2), we denote their lengths and angles by (l_1, l_2) and (θ_1, θ_2), respectively. For any given line segment L_i, we denote its two end-points by (x_{i1}, y_{i1}) and (x_{i2}, y_{i2}). Descriptions of all user-defined as well as adaptive thresholds and parameters are presented in Table 1.

2.1 *Step 1: Grouping Line Segments*

We begin the merging process by sorting the line segments in descending order of length, so smaller segments have less influence on the merging process. This is because longer line segments tend to come from image regions with continuously strong gradients and therefore are more reliable. For the longest line segment L_1, a group $\mathcal{P}_{L_1}$ of segments in close angular proximity is selected as

$$\mathcal{P}_{L_1} = [\forall\, L_2 \in \mathcal{L} : (|\theta_2 - \theta_1| < \tau_\theta)]. \tag{1}$$

This filters out any segments with significantly different orientation from L_1. The set $\mathcal{P}_{L_1}$ is further filtered using spatial

Downloaded From: https://www.spiedigitallibrary.org/journals/Journal-of-Electronic-Imaging on 12 Aug 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

proximity. Specifically, we first find segments in $\mathcal{P}_{L_1}$ with any end-point close to any end-point of L_1 in terms of absolute distance along the horizontal axis as

$$\mathcal{P}_{L_1} = [\forall\, L_2 \in \mathcal{P}_{L_1} : (|x_{11} - x_{21}| < \tau_s \vee |x_{11} - x_{22}| < \tau_s \vee |x_{12} - x_{21}| < \tau_s \vee |x_{12} - x_{22}| < \tau_s)]. \quad (2)$$

This prunes out horizontally distant segments from $\mathcal{P}_{L_1}$. Similarly, vertically distant segments can be pruned out from the remaining segments by

$$\mathcal{P}_{L_1} = [\forall\, L_2 \in P_{L_1} : (|y_{11} - y_{21}| < \tau_s \vee |y_{11} - y_{22}| < \tau_s \vee |y_{12} - y_{21}| < \tau_s \vee |y_{12} - y_{22}| < \tau_s)]. \quad (3)$$

These three filters yield a set $\mathcal{P}_{L_1}$ of segments in close angular and spatial proximity with segment L_1. The sequence of the three filters is important in that the computationally less expensive angular filter is applied first. In addition, axis-aligned absolute distances are also computationally less expensive than Euclidean distances. Finally, each segment L_2 in $\mathcal{P}_{L_1}$ is considered for merging with segment L_1. If merging criteria are met, the merged segment M immediately replaces L_1 and L_2 is removed. Remaining segments in $\mathcal{P}_{L_1}$ are considered for merging with the new merged segment. Pseudocode for the overall procedure is presented in Algorithm 1, and a visualization of the pipeline is shown in Fig. 2.

Algorithm 1 mergeLines.

Data: Set of detected line segments $\mathscr{D}$, spatial distance parameter ξ_s, and angular difference threshold τ_θ

Result: Set of merged line segments $\mathscr{L}$

1 $\mathscr{L} \leftarrow \mathscr{D}$

2 repeat

3 $n \leftarrow |\mathscr{L}|$

4 $\mathscr{L} \leftarrow$ sort lines in $\mathscr{L}$ in descending order of length

5 **for** $L_1 \in \mathscr{L}$ **do**

6 $l_1 \leftarrow \|L_1\|$

7 $\tau_s \leftarrow \xi_s l_1$ [Eq. (4)]

8 $\mathscr{P} \leftarrow$ lines in $\mathscr{L}$ with angular difference from L_1 within $\pm\tau_\theta$ [Eq. (1)]

9 $\mathscr{P} \leftarrow$ lines in $\mathscr{P}$ with spatial distance from L_1 within τ_s [Eqs. (2) and (3)]

10 $\mathscr{R} \leftarrow \varnothing$

11 **for** $L_2 \in \mathscr{P}$ **do**

12 $M \leftarrow$ mergeTwoLines $(L_1, L_2, \xi_s, \tau_\theta)$ (see Algorithm 2)

13 **if** $M \neq \varnothing$ **then**

14 $L_1 \leftarrow M$

15 $\mathscr{L}(L_1) \leftarrow M$ (replace segment L_1 in set $\mathscr{L}$ by the merged segment)

16 $\mathscr{R} \leftarrow \mathscr{R} \cup L_2$(mark segment L_2 for removal from set $\mathscr{L}$)

17 **end**

18 **end**

19 $\mathscr{L} \leftarrow \mathscr{L} \setminus \mathscr{R}$

20 **end**

21 until $|\mathscr{L}| = n$.

2.1.1 *Computational complexity*

Upper bound of the algorithm is of order n^3, where n is the number of detected line segments. This can be understood as follows. The algorithm terminates when the outer loop on line 2 does not merge any pair of line segments; this constitutes the best case behavior. The worst case behavior is when this loop merges only one pair of line segments. This leaves $n-1$ remaining line segments for the next iteration of this loop. Therefore, this loop can run $n-1$ times at most. The internal loops at lines 5 and 11 depend on the sizes of the sets $\mathcal{L}$ and $\mathcal{P}$, both of which are $O(n)$. Therefore, the worst case time complexity is $O(n^3)$. Notice, however, that in practice,

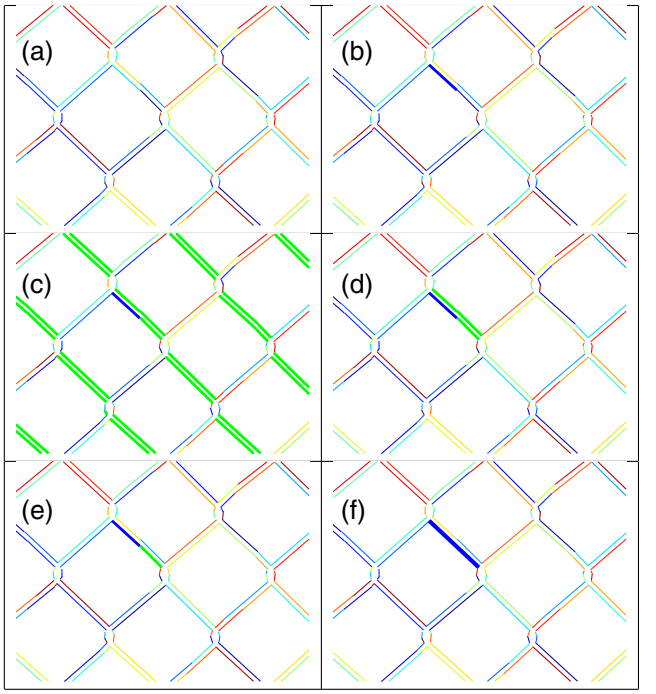

Fig. 2 Visualization of the merging steps of Algorithm 1. (a) Set $\mathscr{D}$ of line segments detected by LSD on the fence image from Fig. 1. (b) A selected line segment L_1 (thick blue). (c) Segments with angular proximity to L_1 (thick green).(d) Set $\mathscr{P}$ of segments with angular as well as spatial proximity to L_1 (thick green). (e) A segment L_2 selected from $\mathscr{P}$ (thick green). (f) Segment M obtained by merging L_1 and L_2 (thick blue). (Best viewed in color.)

Downloaded From: https://www.spiedigitallibrary.org/journals/Journal-of-Electronic-Imaging on 12 Aug 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

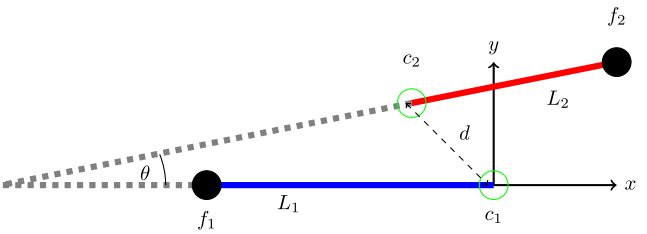

Fig. 3 Merging geometry and notation. The longer line segment L_1 (in blue) is considered as the negative horizontal axis direction of a local two-dimensional coordinate system. The shorter line segment L_2 is shown in red. The closest end-points between the two line segments are marked with empty green circles, and the distance between them is denoted by d. The farthest end-points between the two line segments are marked with filled black circles. Closest point of the longer line segment is considered as the origin of the local coordinate system. (Best viewed in color.)

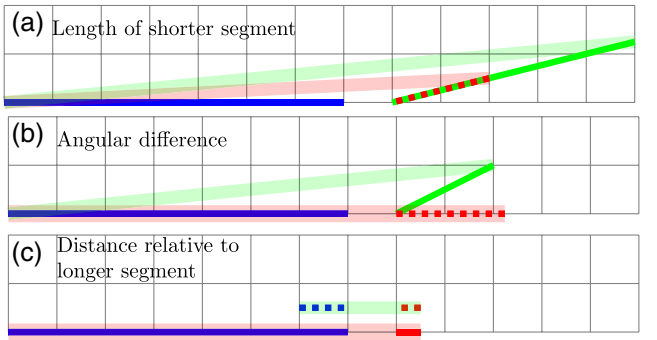

Fig. 4 Perception of mergeability. Each case has two merging scenarios: one is closer to perception and the other is not. The longer line segment is in blue. Potential candidates for merging with blue are the red and green segments (both dashed and solid). Merged segments are transparent: red is closer to perception than green. (a) Despite having the same spatial and angular proximity as the green segment, the red dashed segment is perceptually closer to the blue one. Due to the shorter length of the red segment, the eventual merged segment is closer to the blue one. (b) Despite having the same length and spatial proximity as the green segment, the red dashed segment is perceptually closer to the blue one. Due to greater angular proximity of the red segment, the eventual merged segment is closer to the blue segment. (c) Despite having the same spatial and angular proximity as the dashed pair, the solid pair is perceptually closer. The perceptual difference is due to the relative lengths of the longer segments in the two pairs. Therefore, perceptual mergeability is inversely proportional to length of the shorter segment, angular difference, and relative spatial distance and is proportional to length of the longer segment. (Best viewed in color.)

the number of iterations of all three loops will be far less than n. For example, for the first iteration of the outer loop, if 10 line segments are merged, the repeat loop will have to run $n-10$ times. If the second iterations merge 10 more line segments, the repeat loop will have to run $n-20$ times and so on. Similarly, the first for loop will not run n times after each merging step since the size of set $\mathcal{L}$ keeps decreasing after every merging. The inner-most for loop will iterate much less than n times due to filters applied in lines 8 and 9 of the algorithm. So the actual number of iterations of the three loops is much less than n^3.

2.2 *Step 2: Merging Two Line Segments*

We now describe our perceptually motivated merging criteria for any pair of line segments. To avoid shorter line segments from influencing the merging process, we set the longer line segment as our reference for computing distances and adaptive thresholds. We allow a shorter line segment to be merged into a longer one, but never vice versa. Therefore, without loss of generality, we name the longer line segment L_1 and shorter one L_2. The geometry for two line segments and their merging criterion is shown in Fig. 3. The closest pair of end-points among the four intersegment pairs is denoted by c_1 and c_2. The Euclidean distance between them is denoted by d. If the closest distance d is greater than a spatial proximity threshold τ_s, segments L_1 and L_2 are deemed to be unmergeable. Otherwise, we check subsequent merging criteria that are described next.

Human perception of mergeable line segments does not depend merely on angular and spatial proximity. We contend that our perception also utilizes intersegment distances and the lengths of the segments themselves. It can be seen from Fig. 4 that perception of mergeability is inversely proportional to

1. the length of the shorter line,
2. angular difference, and
3. the relative spatial distance

and is proportional to the length of the longer line segment. Moreover, the amount of angular and spatial disparity that can be tolerated is dependent upon the segments' lengths. We therefore introduce length- and distance-dependent thresholds for making mergeability decisions. This insight leads to adaptive thresholds and, therefore, more flexibility in the proposed technique. This is important because incorrectly set thresholds are the bane of low-level vision tasks such as ours.

Since mergeability is proportional to length l_1 of the longer segment, the threshold for spatial proximity can be set adaptively as

$$\tau_s = \xi_s l_1, \tag{4}$$

where $0 < \xi_s < 1$ is a user-specified fraction. For each segment L_1, the adaptive threshold τ_s determines the maximum allowable value for distance d between the closest points c_1 and c_2. For example $\xi_s = 0.1$ implies that spatial proximity threshold for segment L_1 is 10% of the length of L_1.

To incorporate the inverse relationship between mergeability and length l_2 of the shorter segment, we first normalize l_2 by its maximum allowable value l_1. We thus obtain normalized length

$$\hat{l}_2 = \frac{l_2}{l_1}. \tag{5}$$

Similarly, we normalize the closest distance d by its maximum allowable value τ_s. This gives us normalized distance

$$\hat{d} = \frac{d}{\tau_s}. \tag{6}$$

A combined normalized length-and-distance penalty can be computed as

$$\lambda = \hat{l}_2 + \hat{d}. \tag{7}$$

Downloaded From: https://www.spiedigitallibrary.org/journals/Journal-of-Electronic-Imaging on 12 Aug 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

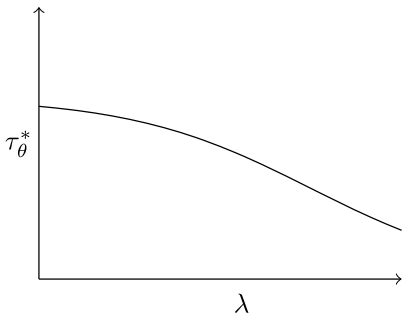

Fig. 5 Adaptive length-and-distance penalized angular threshold for line segment pair merging. When the length l_2 of the shorter line segment increases or the closest distance d between the line segments increases, the angular threshold decreases to make merging harder.

The value of λ is inversely proportional to the perception of mergeability and can vary from 0 for perceptually mergeable segments to 2 for not so mergeable segments. For example, when both l_2 and d are small, mergeability increases and λ decreases. Finally, this adaptive length-and-distance penalty is used to adaptively adjust the angular difference threshold τ_θ as

$$\tau_\theta^* = \left(1 - \frac{1}{1 + e^{-2(\lambda - 1.5)}}\right)\tau_\theta. \tag{8}$$

The behavior of the adaptive angular threshold τ_θ^* is shown in Fig. 5 as a function of the penalty λ. A simple interpretation of the function is that as the penalty λ increases, the angular threshold decreases to make mergeability more difficult.

Segment pairs that pass the angular difference threshold τ_θ^* are merged according to the merging configurations shown in Fig. 6. All scenarios reduce to using the farthest end-points among the four end-points of segments L_1 and L_2 as the end-points of the merged segment M.

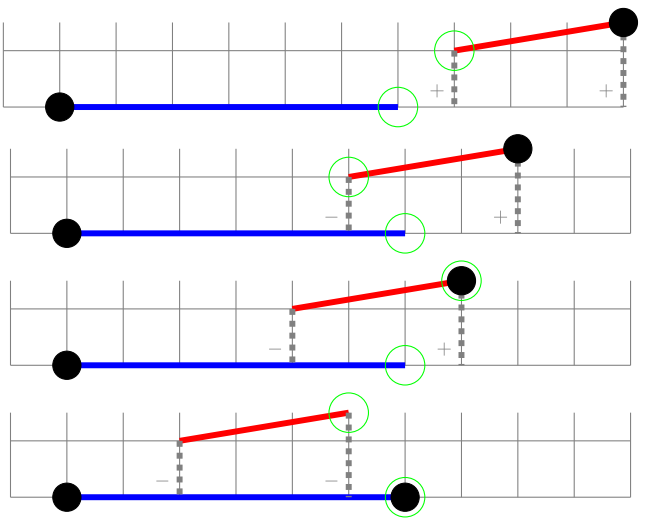

Fig. 6 The four possible merging scenarios for a pair of line segments and their merged results. All line pair configurations can be modeled as one of these four scenarios. The plus and minus signs indicate on which side of the origin a projection falls. They indicate overlap between the two line segments. The end-points of the merged line are marked with black circles. (Best viewed in color.)

Algorithm 2 mergeTwoLines.

Data: Lines L_1 and L_2, spatial distance parameter ξ_s, and angular difference threshold τ_θ

Result: Merged line M

$l_1 \leftarrow \|L_1\|$

$l_2 \leftarrow \|L_2\|$

$\theta_1 \leftarrow \text{angle}(L_1)$

$\theta_2 \leftarrow \text{angle}(L_2)$

if $l_1 < l_2$ **then**

  swap (L_1, L_2)

  swap (l_1, l_2)

  swap (θ_1, θ_2)

end

Compute c_1, c_2 and d according to geometry of Fig. 3

$\tau_s \leftarrow \xi_s l_1$

if $d > \tau_s$ **then**

  $M \leftarrow \varnothing$

  return

end

Compute adaptive angular threshold τ_θ^* via Eqs. (7) and (8) (see Fig. 5)

$\theta \leftarrow |\theta_2 - \theta_1|$

if $(\theta < \tau_\theta^*)$ OR $[\theta > (\pi - \tau_\theta^*)]$ **then**

  $f_1, f_2 \leftarrow$ end-points corresponding to appropriate black circles from Fig. 6

  $M \leftarrow$ segment with end-points f_1 and f_2

  $\theta_M \leftarrow$ angle (M)

  if $|\theta_1 - \theta_M| > \frac{1}{2}\tau_\theta$ **then**

    $M \leftarrow \varnothing$

  end

else

  $M \leftarrow \varnothing$

  return

end

Downloaded From: https://www.spiedigitallibrary.org/journals/Journal-of-Electronic-Imaging on 12 Aug 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

A final check is placed on the absolute angular difference of the longer segment L_1 and the merged segment M. For the experiments in this paper, if the angular difference is greater than $\tau_\theta/2$, we discard the merging. The extra strictness of this threshold is to ensure that merged segments do not stray too far from the longer segment L_1. Pseudocode for the overall algorithm for merging two line segments L_1 and L_2 is presented in Algorithm 2.

MATLAB® implementation of the merging pipeline with a demo file is open to the public and freely available at Ref. 10.

3 Quantitative Evaluation Criterion

Due to the human brain's ability to fill gaps and insert missing information,[11] qualitative evaluation of line detection algorithms is not sufficient. The output of line segment detectors appears better than it actually is because our brain automatically introduces order where there is little or no order. It connects together or fills in the broken line segments. This can be evidenced by looking at line segments detected by the LSD and EDLines algorithms whose weaknesses can often be observed only by zooming into the image (see Figs. 7–11). Due to the absence of a standard benchmark dataset with ground-truth line segments, many line segment extractors[4,5,8] analyze their results on synthetic images while results on real images are open to subjective and visual evaluation. Therefore, we propose a method for quantitative comparison of line segment detection algorithms.

For two line segments L_1 and L_2, let $\mathbf{v}_1 = [x_{11}, y_{11}, x_{12}, y_{12}]^T$ be the four-dimensional vector constructed from the end-points of L_1 and similarly $\mathbf{v}_2 = [x_{21}, y_{21}, x_{22}, y_{22}]^T$ from end-points of L_2. Dissimilarity of line segments L_1 and L_2 can be measured as

$$\delta(L_1, L_2) = \|\mathbf{v}_1 - \mathbf{v}_2\|^2. \tag{9}$$

Let $\mathcal{G}$ be the set of ground-truth line segments in an image, and let $\mathcal{D}$ be any set of detected line segments. For each ground-truth line $G \in \mathcal{G}$, we can compute the dissimilarity δ_G of line G from the most similar line in $\mathcal{D}$ via

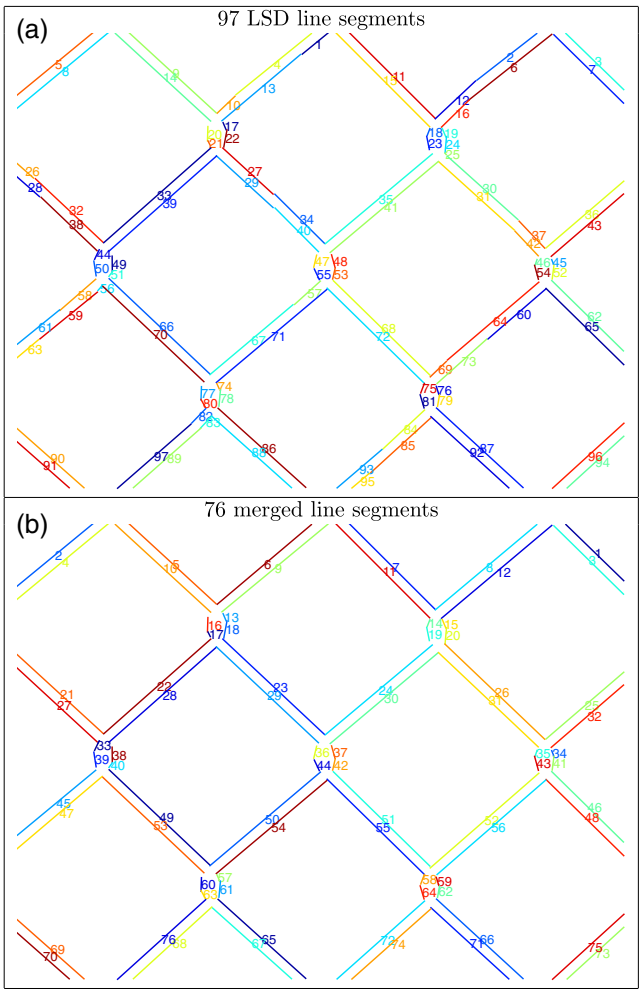

Fig. 7 Merging LSD output. (a) 97 line segments detected by the LSD algorithm on the fence image from Fig. 1. (b) After merging LSD segments by our algorithm using $\xi_s = 0.05$ and $\tau_\theta = 20$ deg, only 76 perceptually accurate segments remain. (Best viewed in color and by zooming on a computer screen.)

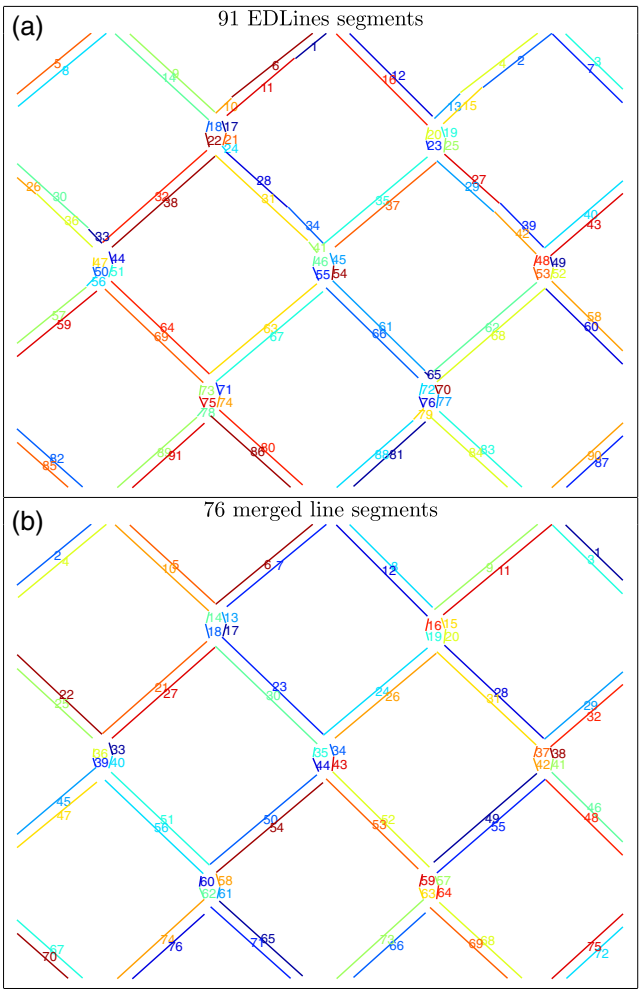

Fig. 8 Merging EDLines output. (a) 91 line segments detected by the EDLines algorithm on the fence image from Fig. 1. (b) After merging EDLines segments by our algorithm using $\xi_s = 0.05$ and $\tau_\theta = 20$ deg, only 76 perceptually accurate line segments remain. (Best viewed in color and by zooming on a computer screen.)

Downloaded From: https://www.spiedigitallibrary.org/journals/Journal-of-Electronic-Imaging on 12 Aug 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

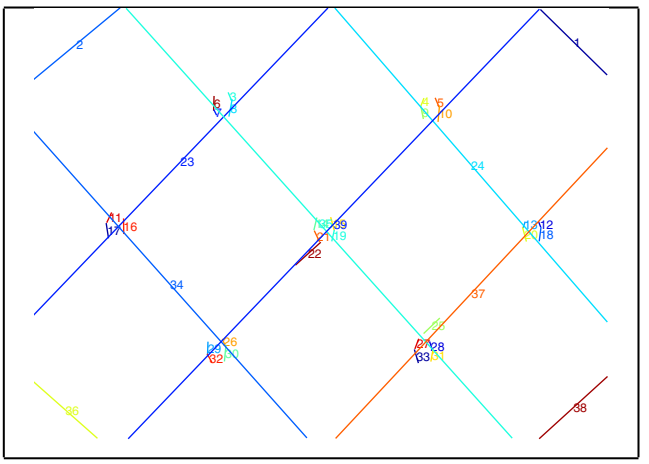

Fig. 9 Greater abstraction of merging the line segments for the fence image from Fig. 1. Compared to Fig. 7(b), this result is obtained by relaxing the spatial proximity parameter ξ_s to 0.255 which allows close parallel lines to become merged. (Best viewed in color.)

$$\delta_G = \frac{\min_{L\in\mathcal{D}}\delta(G,L)}{\max(\|G\|,\|L\|)}. \tag{10}$$

The overall dissimilarity measure of ground-truth line segments $\mathcal{G}$ from detected line segments $\mathcal{D}$ can be computed by averaging the individual dissimilarity measures as

$$\delta(\mathcal{G},\mathcal{D}) = \frac{1}{|\mathcal{G}|}\sum_{G\in\mathcal{G}}\delta_G. \tag{11}$$

Notice that $\delta(\mathcal{G},\mathcal{D})$ represents an asymmetric measure and that lower values of $\delta(\mathcal{G},\mathcal{D})$ imply that line segments in $\mathcal{D}$ are closer to line segments in $\mathcal{G}$. For a collection $\{\mathcal{G}_n,\mathcal{D}_n\}_1^N$ of ground-truth and detected sets of line segments for N images, we can average the individual dissimilarity measures to obtain an overall dissimilarity measure $\delta_{\mathcal{D}}$ for the line segment detection algorithm

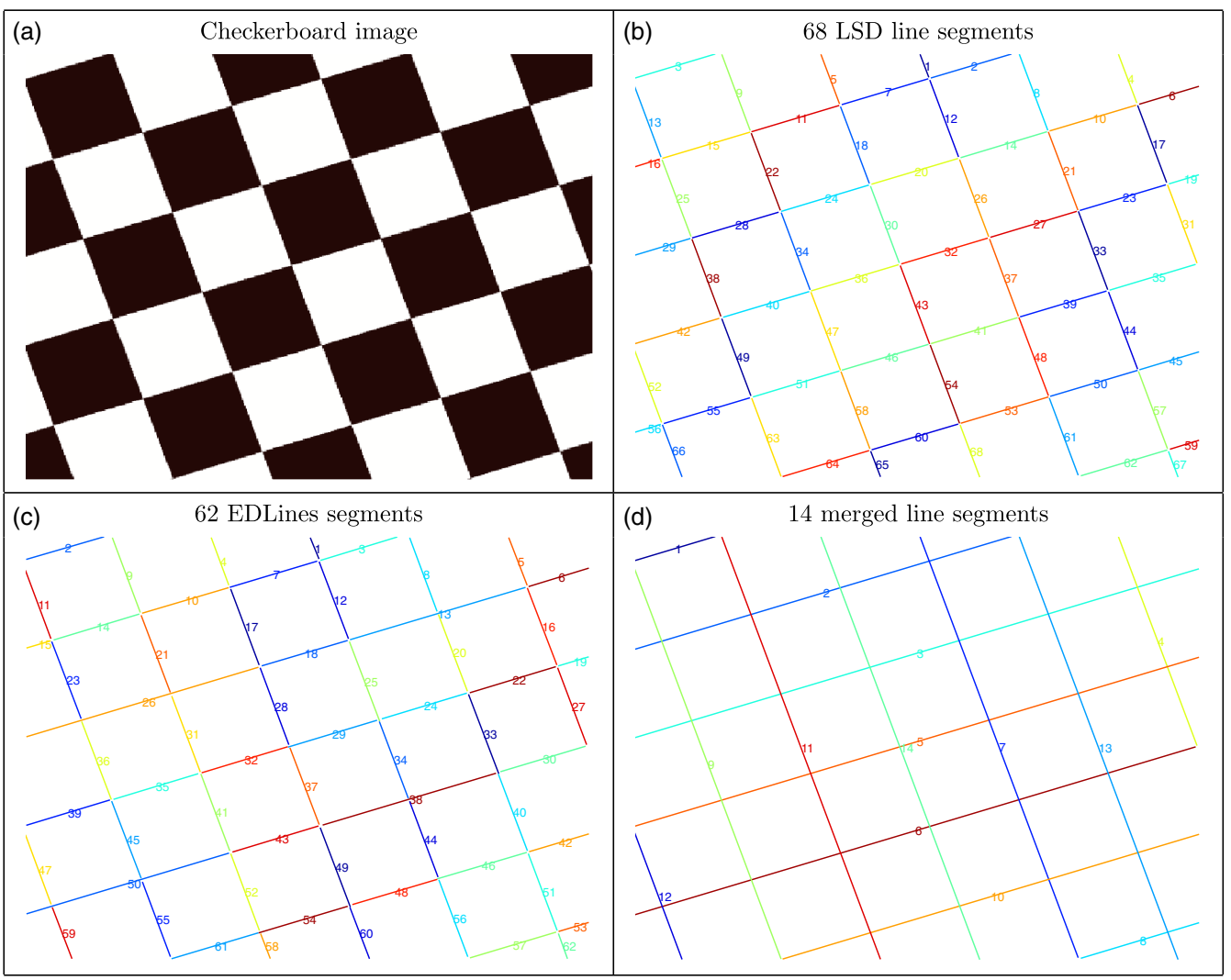

Fig. 10 Merging LSD and EDLines output. (a) A synthetic image of a checkerboard containing 14 line segments that can be identified via visual inspection. Line segment detectors that break segments at intersections should detect 68 segments that can also be identified via visual inspection. (b) LSD detects 68 and (c) EDLines detects 62 line segments. (d) After merging LSD result by our algorithm, the original 14 line segments are recovered. Merging EDLines result also yields the same 14 line segments. Both results were obtained using $\xi_s = 0.05$ and $\tau_\theta = 20$ deg. (Best viewed in color and by zooming on a computer screen.)

Downloaded From: https://www.spiedigitallibrary.org/journals/Journal-of-Electronic-Imaging on 12 Aug 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

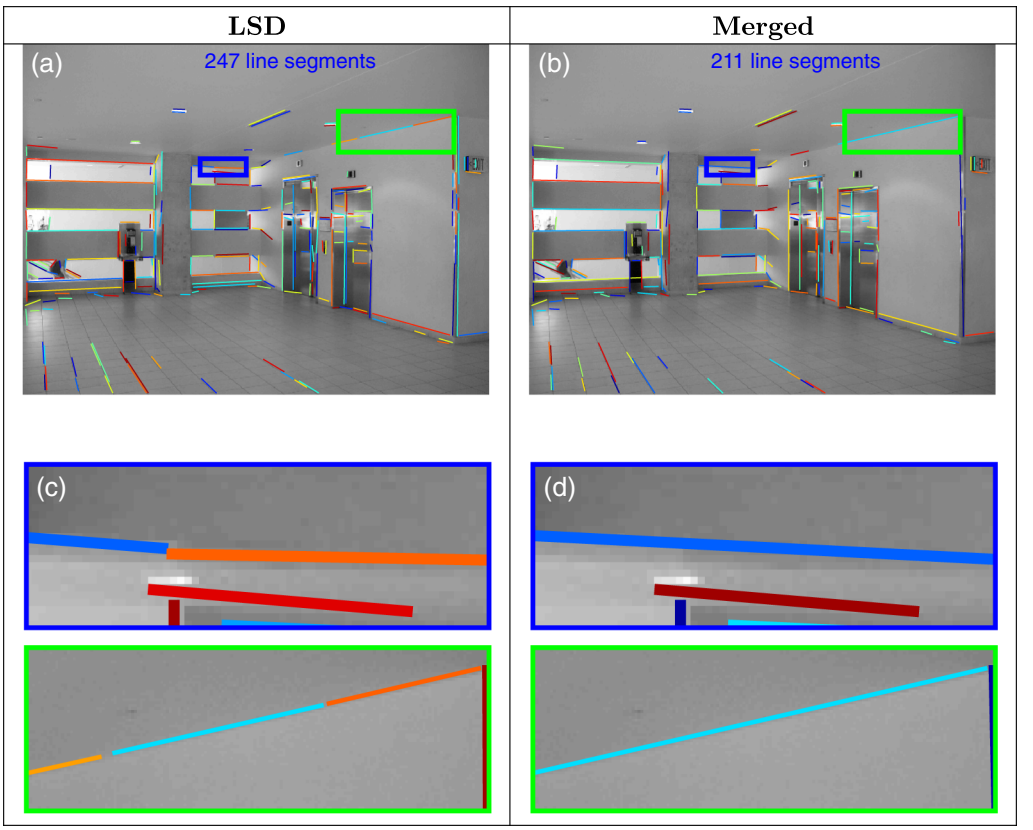

Fig. 11 Line segments obtained by LSD and merged using our method on an image from York Urban dataset. (a) LSD gives broken line segments (viewable by zooming into the image). (b) Merged LSD line segments using our method. (c) LSD has given multiple broken line segments for a perceptually single line segment. (d) Our merging algorithm has merged the broken line segments. (Best viewed in color.)

$$\delta_{\mathcal{D}} = \frac{1}{N}\sum_{n=1}^{N} \delta(\mathcal{G}_n, \mathcal{D}_n). \tag{12}$$

Our merging algorithm can be similarly evaluated by computing $\delta_{\mathcal{L}}$ for the collection of merged line segments $\{\mathcal{L}_n\}_1^N$. The success of any merging algorithm can be measured by the ratio

$$r = \frac{\delta_{\mathcal{D}}}{\delta_{\mathcal{L}}}, \tag{13}$$

where $\mathcal{D}$ represents input line segments. A merging algorithm is useful only when $r > 1$.

4 Results and Analysis

We have already shown in Fig. 1 that a good line segment detector should detect 76 line segments on the fence image. The state-of-the-art LSD algorithm[12] detects 97 line segments instead of 76 as shown in Fig. 7. It detects more line segments because it tends to break up perceptually coherent segments. Our merging process applied on the LSD output merges these segments to obtain the original 76 perceptually accurate line segments.

A similar analysis for the EDLines algorithm[13] is shown in Fig. 8. It should be observed that we obtain identical merging results in Figs. 7 and 8 corresponding to two different initial line segment detectors. This demonstrates the ability of our merging algorithm to produce consistent, perceptually accurate results in a manner that is not overly sensitive to the input segments.

Figures 7 and 8 show merging results for one setting of the user-defined spatial proximity parameter ξ_s. By relaxing this parameter, we can obtain greater abstraction of merging the line segments, as shown in Fig. 9. It can be observed that now close-lying parallel line segments and thick intersections have also been merged. Such an abstraction makes it easier to answer questions such as how many intersections can be seen in the fence.

Another comparison is shown in Fig. 10, which is a synthetic image of a checkerboard. LSD and EDLines break line segments at each intersection, and our algorithm merges all the broken intersections, thus yielding 14 perceptually accurate line segments.

4.1 *Effect of Scaling and Noise*

The effect of scaling on the fence image from Fig. 1 can be seen in Tables 2 and 3. The adverse effect of scale on our merging process is, in part, due to the fact that scale affects the output of the segment detectors that we have used. However, it should be noted that scaling does not affect our algorithm as much as it affects the detectors. For example, without scaling, LSD detects 97 segments. When the image is scaled up, LSD starts detecting more segments. However, our algorithm merges many of them to yield a number of segments that is closer to the perceptually accurate number of 76. When the image is scaled down, LSD

Downloaded From: https://www.spiedigitallibrary.org/journals/Journal-of-Electronic-Imaging on 12 Aug 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

Table 2 Effect of scaling the fence image form Fig. 1. On the original image size, LSD detected 97 line segments and we merged them into 76 perceptually accurate line segments (Fig. 7). When scaled up, LSD detects more segments. Our merging algorithm using $\xi_s = 0.05$ and $\tau_\theta = 20$ deg produces a result closer to the perceptually accurate value of 76. When scaled down, LSD detects less than 76 segments and our algorithm stops the merging process quickly since further merging would lead to deviation from perceptual accuracy.

Scale factor	LSD segments	After merging
2	164	92
1.5	127	82
1	97	76
0.5	70	69

Table 3 Similar to Table 2, merging EDLines output either moves toward or remains close to perceptually consistent segments.

Scale factor	EDLines segments	After merging
2	151	90
1.5	130	89
1	91	76
0.5	66	64

detects fewer than 76 segments, and our algorithm stops the merging process quickly since further merging would lead to deviation from perceptual accuracy. Similar analysis holds true for EDLines.

The effect of noise on the fence image from Fig. 1 can be seen in Tables 4 and 5. Noise also affects our algorithm adversely. But once again, its effect on the detectors is more pronounced than on the merging algorithm. For the particular case of LSD output (which happens to be highly noise sensitive), noise causes angular and spatial differences to be amplified for short line segments. Since such differences are included in key merging criteria for our algorithm, our performance suffers too. The EDLines algorithm is more robust to noise. Therefore, its output requires considerably less merging. Our algorithm automatically adjusts and performs the appropriate level of merging.

Table 4 Effect of noise on the fence image from Fig. 1. As noise increases, LSD detects many more broken segments which our algorithm using $\xi_s = 0.05$ and $\tau_\theta = 20$ deg tries to merge as much as possible. The failure to obtain 76 perceptually accurate segments is due to the amplification of angular disparity for shorter segments in the presence of noise.

Variance of Gaussian noise	LSD segments	After merging
0.01	472	110
0.02	540	167
0.03	558	213
0.04	538	276

Table 5 Similar to Table 4. For low noise levels, EDLines output is not disturbed and our merging using $\xi_s = 0.05$ and $\tau_\theta = 20$ deg yields 70 perceptually accurate segments. For increased noise, EDLines starts detecting less than perceived segments and our algorithm stops merging.

Variance of Gaussian noise	EDLines segments	After merging
0.01	93	76
0.10	98	72
0.50	30	30
1.00	3	3

4.2 *Evaluation Using Ground-Truth Data*

For numerical evaluation with human-marked ground-truths, we have used the York Urban dataset,[14] which is a collection of 102 (45 indoor and 57 outdoor) images of urban environments with ground-truth line segment markings. The size of each image is 640×480. Ground-truth line segments on each image were manually marked with subpixel precision using an interactive tool. Since the original work accompanying the dataset focused on man-made structures such as buildings, the ground-truth consists of line segments representing man-made structures only. Other valid line segments, such as those representing shadow boundaries, are not marked in the ground-truth. As a result, the marked ground-truth underestimates the actual number of perceptually accurate line segments. This does not, however, affect our evaluation criterion since our eventual comparison is between detected and merged line segments [Eq. (13)].

In Table 6, we evaluate results from LSD and our merging algorithm applied to LSD results against ground-truth line

Table 6 Merged line segments compared to LSD line segments in terms of (a) merging success r [see Eq. (13)] and (b) average number of line segments merged per image. The best case is highlighted in bold. In this best case, LSD detects on average 606 line segments per image and we merge 113 of them in 2.95 s per image on average.

	τ_θ			
ξ_s	1 deg	5 deg	10 deg	15 deg
	(a) Merging success r			
0.01	1.0584	1.0901	1.0908	1.0905
0.05	1.1306	**1.1838**	1.1809	1.1762
0.10	1.1691	1.1011	0.9367	0.8848
	(b) Average number of segments merged per image			
0.01	5	12	13	13
0.05	31	**90**	107	113
0.10	58	165	218	237

Downloaded From: https://www.spiedigitallibrary.org/journals/Journal-of-Electronic-Imaging on 12 Aug 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

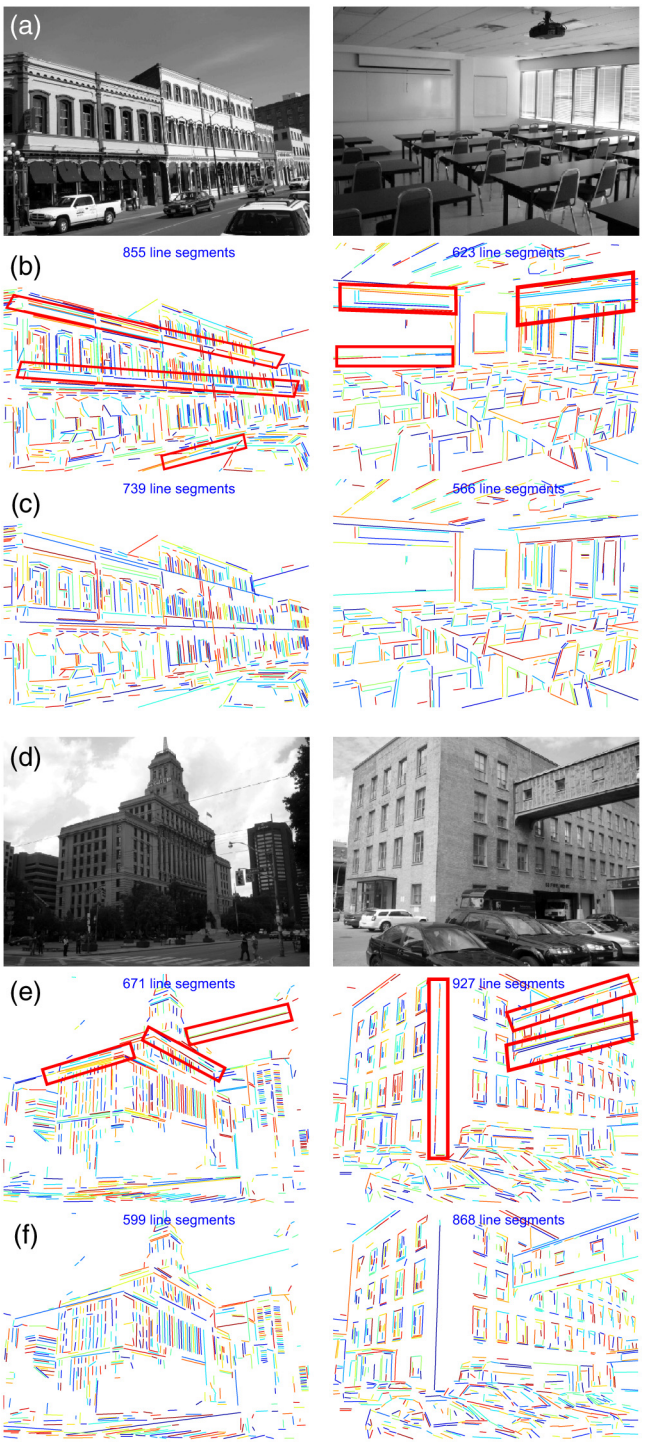

Fig. 12 LSD and merged LSD results. (a) and (d) York Urban dataset images. (b) and (e) LSD results. The red rectangles highlight the weakness of LSD that it breaks perceptually contiguous linear structures. (c) and (f) Merging applied on LSD results. Areas corresponding to the red rectangles demonstrate merged segments that are more consistent with human perception. (Best viewed in color and by zooming on a computer screen.)

segments. The table shows the merging success r computed by our quantitative evaluation criterion. Results are shown for 12 configurations of our user-defined spatial proximity parameter ξ_s and user-defined angular threshold τ_θ. Results for the best configuration are highlighted. Successful configurations correspond to the ones with $r > 1$. For those,

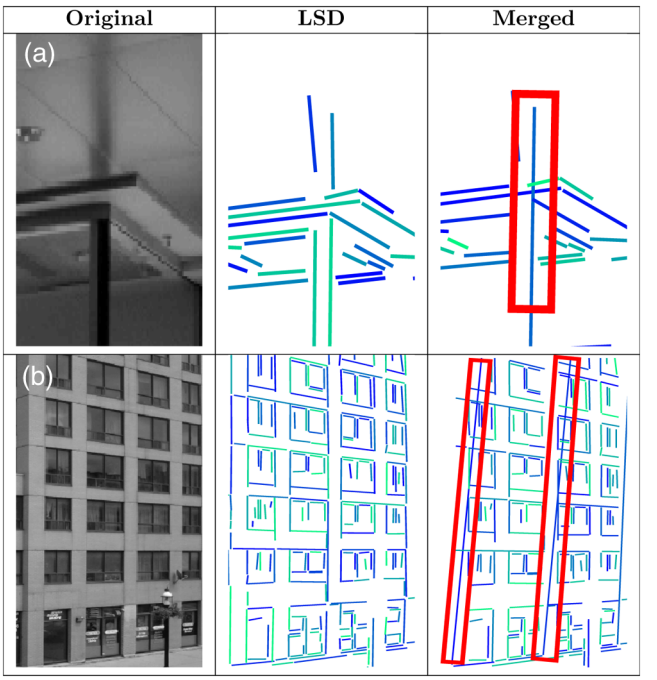

Fig. 13 Weaknesses of the merging algorithm can be seen in the areas marked by red rectangles. (a) Nearby parallel segments that are perceptually distinct can become merged if adaptive thresholds are not estimated correctly. (b) Sequence of local mergings can eventually lead to a globally incorrect line segment.

it can be concluded that our method reduces the number of segments in a way that brings the segments closer to ground-truth that was marked according to human perception. A similar analysis on the output of EDLines on the York Urban dataset revealed that optimal ranges for ξ_s and τ_θ were from 0.05 to 0.1 and 1 deg to 5 deg, respectively.

Figures 11 and 12 show the weaknesses of the LSD algorithm on some examples from the York Urban dataset. It can be observed that LSD consistently breaks lines at intersections and does not yield long, contiguous line segments. Our merging algorithm, on the other hand, merges parallel and broken line segments to yield contiguous, perceptually more consistent line segments.

Figure 13 shows some failure cases of our merging algorithm. It can be observed that sequences of local mergings can eventually lead to a globally incorrect line segment. This is a trade-off between ease of local decision making and global optimality. It can also be observed that incorrect estimation of adaptive thresholds can cause close but perceptually distinct parallel segments to become merged.

5 Conclusions and Future Directions

In this paper, we have proposed a new method to refine the output of an off-the-shelf line segment detector. Our method extracts perceptually more meaningful line segments by merging individual line segments. We introduce unique, adaptive mergeabilty criterion. We have also presented a method for evaluating line segment extractors in the presence of ground-truth markings. Experimental results on synthetic as well as real imagery from the York Urban dataset show that our method yields line segments that lie much closer to human-marked ground-truths compared to results from two other state-of-the-art line segment detectors.

Downloaded From: https://www.spiedigitallibrary.org/journals/Journal-of-Electronic-Imaging on 12 Aug 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

The main strengths of the proposed algorithm are as follows:

1. Thin line segments are recovered as single line segments instead of two separate ones on each side.
2. Segments continue across intersections.
3. Long perceptually coherent segments are recovered as such.
4. Abstraction of linear image content can be obtained by varying user-specified parameters.

The main weaknesses are as follows:

1. Sequences of local mergings can eventually lead to a globally incorrect merging.
2. Due to incorrect estimation of adaptive thresholds, close but perceptually distinct parallel line segments can become merged. Moreover, in some cases, the direction of this merged segment can be slightly different from the original parallel segments.

While we demonstrate on different images that the proposed algorithm overcomes weaknesses of previous line detectors, it is quite evident from our results that the merging algorithm is missing a key proximity criterion—appearance. The algorithm can be extended by comparing the image data underlying the original and merged line segments. This should improve the "blind" merging algorithm by giving it "sight." The weaknesses mentioned above can also be addressed by considering appearance information.

Moreover, since our goal was to get perceptually accurate line segments, we have compared our results with human-marked ground-truths. An alternative method of measuring the benefits of merging is to see how much it improves other computer vision algorithms such as stereomatching,[1] 3-D reconstruction,[2] and vanishing point estimation,[3] among others.

Acknowledgments

Helpful insights on merging two line segments were provided by Malik Masood Ahmed. Hints on computing the computational complexity of the algorithm were provided by Mehwish Irfan Poshni.

References

1. H. Bay, V. Ferraris, and L. Van Gool, "Wide-baseline stereo matching with line segments," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. **1**, pp. 329–336, IEEE (2005).
2. A. Zaheer, M. Rashid, and S. Khan, "Shape from angle regularity," in *European Conf. on Computer Vision*, pp. 1–14, Springer (2012).
3. H. Kong, J.-Y. Audibert, and J. Ponce, "General road detection from a single image," *IEEE Trans. Image Process.* **19**(8), 2211–2220 (2010).
4. R. G. Von Gioi et al., "LSD: a line segment detector," *Image Process. Line* **2**, 35–55 (2012).
5. C. Akinlar and C. Topal, "EDlines: a real-time line segment detector with a false detection control," *Pattern Recognit. Lett.* **32**(13), 1633–1642 (2011).
6. A. Bandera et al., "Mean shift based clustering of Hough domain for fast line segment detection," *Pattern Recognit. Lett.* **27**(6), 578–586 (2006).
7. J.-H. Jang and K.-S. Hong, "Fast line segment grouping method for finding globally more favorable line segments," *Pattern Recognit.* **35**(10), 2235–2247 (2002).
8. R. F. Guerreiro and P. M. Aguiar, "Connectivity-enforcing hough transform for the robust extraction of line segments," *IEEE Trans. Image Process.* **21**(12), 4819–4829 (2012).
9. E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, Vol. **201**, Prentice Hall, Englewood Cliffs (1998).
10. N. Hamid and N. Khan, "LSM: perceptually accurate line segment merging (code)," http://faculty.pucit.edu.pk/nailah/research/LSM (December 2016).
11. L. Albertazzi, Ed., *Shapes of Forms: From Gestalt Psychology and Phenomenology to Ontology and Mathematics*, Kluwer, Dordrecht (1999).
12. R. G. Von Gioi et al., "LSD: a line segment detector (Code)," http://www.ipol.im/pub/art/2012/gjmr-lsd/lsd_1.5.zip (November 2015).
13. C. Akinlar and C. Topal, "EDlines: a real-time line segment detector with a false detection control (Code)," http://ceng.anadolu.edu.tr/CV/downloads/downloads.aspx (November 2015).
14. P. Denis, J. H. Elder, and F. J. Estrada, "Efficient edge-based methods for estimating manhattan frames in urban imagery," in *European Conf. on Computer Vision*, pp. 197–210, Springer (2008).

Naila Hamid is a PhD candidate at Punjab University, College of Information Technology, Pakistan, where she received a BS degree in software engineering in 2012 and an MPhil degree in computer science in 2014. Her research focuses on the fusion of geometric and chromatic information for improving low-level computer vision tasks.

Nazar Khan is an assistant professor at Punjab University, College of Information Technology, Pakistan. He received his BSc degree in computer science from Lahore University of Management Sciences, Pakistan, in 2003, an MSc degree in computer science from the University of Saarland, Germany, in 2007, and a PhD in computer science from the University of Central Florida, USA, in 2013. His research interests are in computer vision and machine learning.

Downloaded From: https://www.spiedigitallibrary.org/journals/Journal-of-Electronic-Imaging on 12 Aug 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use