

ELSM: Evidence-Based Line Segment Merging

Naila Hamid, Nazar Khan* and Arbish Akram

Computer Vision & Machine Learning Group, Department of Computer Science, University of the Punjab, Allama Iqbal Campus, The Mall Road, 54000, Lahore, Pakistan

*Corresponding author: nazarkhan@pucit.edu.pk

Existing line segment detectors break perceptually contiguous linear structures into multiple line segments. This can be offset by re-merging the segments, but existing merging algorithms over-merge and produce globally incorrect segments. Geometric cues are necessary but not sufficient for deciding whether to merge two segments or not. By restricting the result of any merging decision to have underlying image support, we reduce over-merging and globally incorrect segments. We propose a novel measure for evaluating merged segments based on line segment Hausdorff distance. On images from YorkUrbanDB, we show that our algorithm improves both qualitative and quantitative results obtained from four existing line segment detection methods and is better than two existing line segment merging methods. Our method does not suffer from inconsistent results produced by four recent deep learning-based models. The method is easily customisable to work for line drawings such as hand-drawn maps to obtain vectorized representations.

Keywords: line segment detection; line segment merging; perceptually accurate; evaluation; Hausdorff distance

1. INTRODUCTION

Lines and edges offer crucial perceptual cues for scene understanding. This can be seen from our ability to understand comic strips drawn without any colour information. In the fields of image processing and computer vision, line segments serve as fundamental low level features used to perform many high level tasks such as 3D reconstruction [1], structure from motion [2–4], stereo [5, 6], visual odometry [7] and vanishing point estimation [8, 9]. In such works, line segments are not used directly. Multiple line segments need to be grouped together and refined. Small line segments that do not represent any contiguous underlying image structure need to be discarded. This is because existing line segment detectors

- (i) break line segments at intersections, and
- (ii) detect two line segments for a single line due to large gradient on both sides of the line.

Merging of detected line segment is useful for different applications such as unmanned aerial vehicle landing [10] and chessboard detection [11].

In most works, results are presented by overlaying the detected line segments as one-pixel thick lines on the original image. This can tend to hide the above-mentioned problems. Firstly, a line break event is very often contained in a one-pixel radius. Visual inspection on a coarse scale will hardly identify such cases. The line breaking artefacts are illustrated in Fig. 1 by drawing line segments with emphasized endpoints. Such behaviour is concealed by not highlighting the endpoints while drawing line segments in the same colour. Secondly, since the human perceptual system automatically fills in gaps and ignores redundant information, coarse visual inspection, once again, will underestimate the severity of the problem. Figure 1 visually illustrates both of the aforementioned weaknesses. Any computer algorithm that takes detected line segments as inputs is now left with the task of ascertaining which of these broken

segments corresponded to which perceptually consistent line segment.

An obvious solution to these problems is to merge individual line segments in order to recover longer, contiguous segments [12–14]. However, this raises two fundamental questions:

- (i) **Validity of line pair merging:** when any pair of segments is merged, how can the validity of the merged segment be checked?
- (ii) **Validity of continual mergings:** when already merged segments are iteratively considered for further mergings until convergence, how can the validity of the eventual line segments be checked?

Such questions have not been explicitly targeted by existing merging methods. The weaknesses of the existing methods can be summarised as follows.

W1: Iterative mergings can sometimes eventually produce longer segments that are neither supported geometrically by any of the original smaller segments nor chromatically by underlying image data (Fig. 2).

W2: Methods are biased toward natural imagery making them unsuitable for vectorization of line drawings, which yield two overlapping parallel segments for each visible line (Fig. 3).

In this work, we address both weaknesses as follows.

S1: Supplement geometric information by chromatic information. We use image evidence for validating line segments so that any segments not supported by underlying image intensities are rejected.

S2: Supplement end-points by additional reference points. While end-points are sufficient for defining a segment, they are not sufficient for deciding the proximity between different segments.

S3: For line drawings that yield redundant segment detections, we introduce an overlap tolerance to remove redundancy.

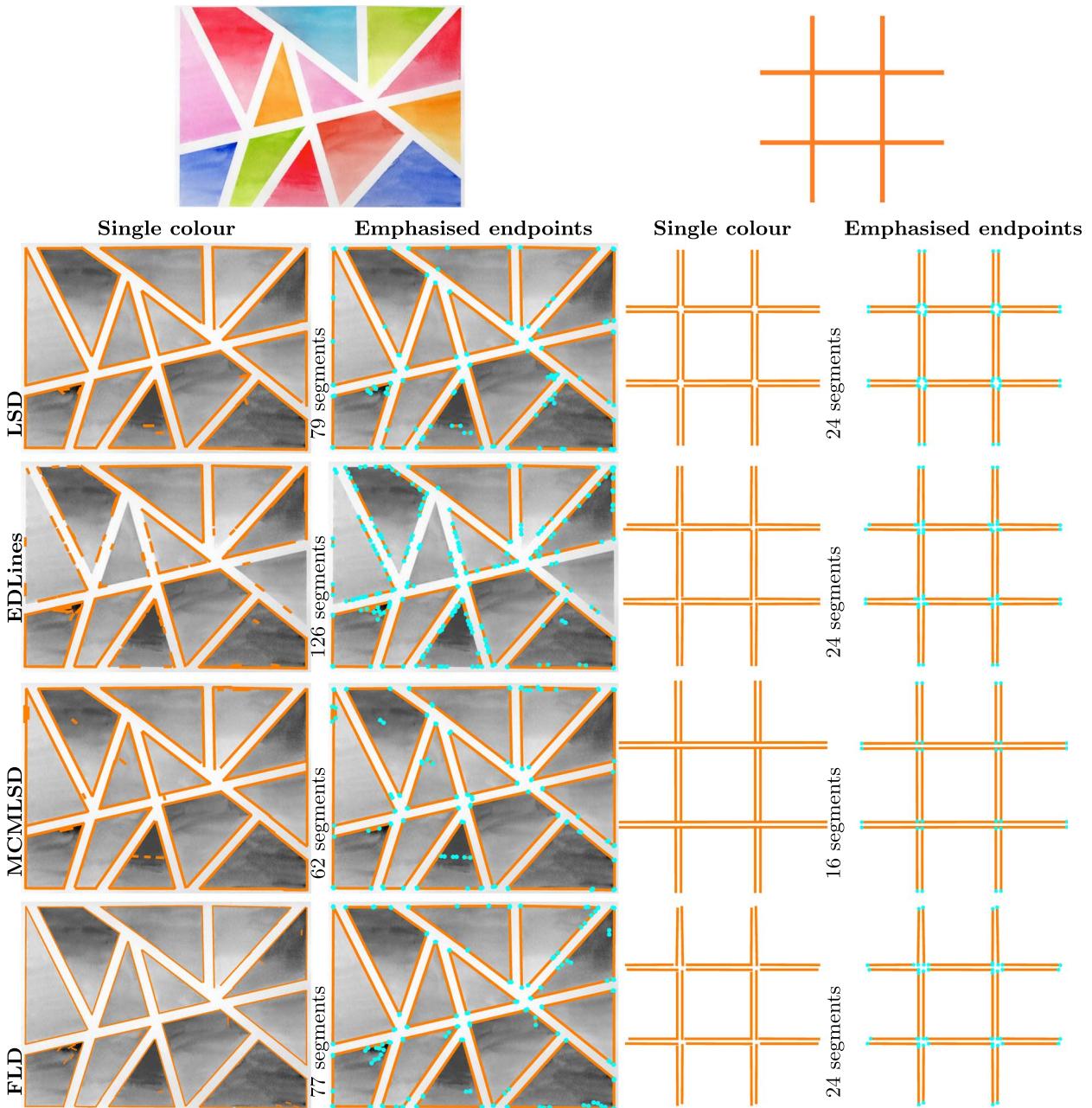


Figure 1. Left: Line segment detectors break perceptually consistent line segments into multiple segments. When overlayed using a single colour, the breaks cannot be easily seen. However, by emphasizing their endpoints, we can observe the breaks. **Right:** For line drawings, Line segment detectors (i) break segments at intersections and (ii) yield two detections per segment due to gradient change on both sides of a single segment. Such double detections are counter-productive for vectorization of line drawings.

By employing line segment evidence and additional reference points for grouping and for accepting or rejecting merging decisions, our proposed evidence-based line segment merging (ELSM) algorithm produces longer, perceptually accurate segments that are still faithful to the underlying image data. Our method is suited for both natural imagery as well as line drawings.

We also address an additional problem related to the evaluation of line segments against ground truth. Existing datasets have an annotation bias so that not all line segments are marked in the ground truth. For instance, YorkUrbanDB [17] predominantly contains ground truth segments at so-called Manhattan orientations. They correspond to visually significant boundaries in building facades. As can be seen from Fig. 4, numerous genuine line segments with strong underlying image gradients

are left unmarked. The ground truth of the Wireframe dataset [18] also marks object frame boundaries. As a consequence, any edge-strength-based line segment detector will be unfairly evaluated using such ground truth. Therefore, we propose a quantitative evaluation scheme that compares merged segments against detected segments based on Hausdorff distance from ground truth. Incompleteness of the ground truth is neutralized when merged segments are compared with detected segments. Numerical results show that the proposed ELSM algorithm avoids both over-segmentation and over-merging and produces line segments that are closer to ground truth segments marked according to human perception.

The rest of the paper is organised as follows. Section 2 surveys line segment detection and line segment grouping and merging

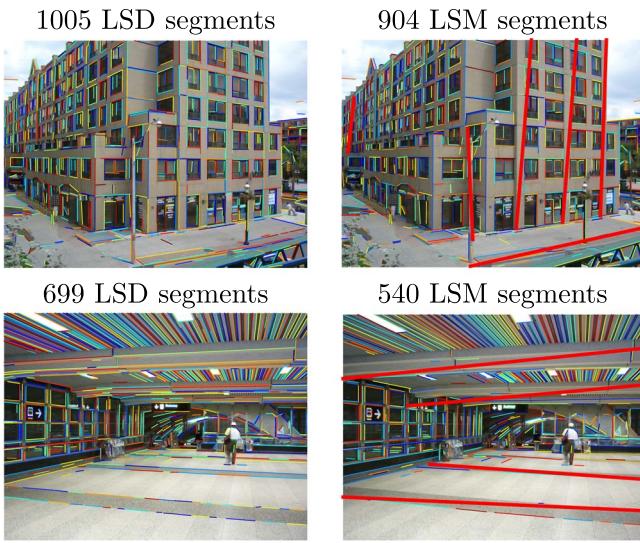


Figure 2. Weakness (**W1**) of current merging approaches. The Line Segment Merging (LSM) algorithm [12] iteratively combines broken segments detected by the Line Segment Detector (LSD) algorithm [15]. However, it fails to control for the global correctness of continual mergings. The highlighted LSM segments (thick red lines) represent merged line segments that have moved away from the original segments and therefore no longer represent any underlying line segments in the input image.

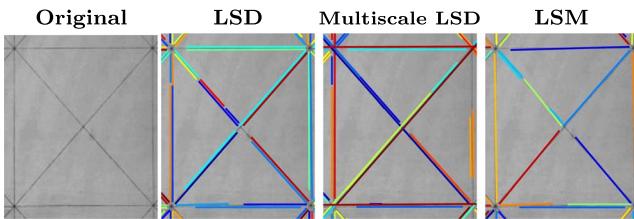


Figure 3. Weakness (**W2**) of current merging approaches on line drawings that consist of roof edges. LSD [15] detects excessively broken segments on both sides of each line. The Multiscale LSD algorithm [16] produces longer segments but still on both sides. The LSM algorithm [12] merges some but not all instances of such perceptually redundant lines. Such instances become computational as well as semantic baggage for downstream tasks such as vectorization and image understanding.

techniques with existing evaluation methods. In Section 3, we present our line segment evidence-based grouping and merging technique. Section 4 describes our proposed evaluation metric. Our experimental setup is described in Section 5 whose results are analysed in Section 6. Concluding thoughts and future directions of research are presented in Section 7.

2. LITERATURE REVIEW

2.1. Line segment detection techniques

Starting from the top-down approach of the classical Hough Transform [19], line detection has evolved from the bottom-up approaches such as Line Segment Detector (LSD) [15], Edge Drawing Lines (EDLines) [20] and Fast Line Segment Detector (FLD) [21] to recent approaches that learn to detect lines given training images with marked ground-truth line segments [22–28]. All existing detection methods produce over-segmented (broken) lines that require post-processing in order to be useful for higher level tasks. While learning-based methods tend to reduce the over-segmentation phenomenon, they can sometimes yield inexplicable false positives as well as false negatives that a pure

gradient-based detector would never suffer from. Furthermore, they are trained on datasets with a Manhattan bias—ground-truth lines are marked only in Manhattan directions while many, otherwise valid, non-Manhattan lines are left unmarked.

Some methods have enforced segment grouping within the line detection process. For example, restricting Hough voting to lines with spatio-angular proximity in connectivity enforcing Hough Transform [29], edge linking between clusters of collinear points in Canny Lines [30] and combining image and Hough space to improve segment localization in the Markov Chain Marginal Line Segment Detector (MCMLSD) [31].

2.2. Use of merged line segments

Line segments play an important role in 3D reconstruction. For single-view reconstruction [32], pairs of intersecting line segments are exploited. However, segments detected via LSD [15] are excessively broken and need to be extended in order to fill small gaps so that they intersect. For estimation of vanishing points [9], line segments are grouped together in order to narrow down the space for computational strategies.

For extraction of line networks in noisy, low-contrast images [33], lines are linked together via directional propagation to reconstruct connectivity of the network. For segmenting land regions in images of historical cadastral maps [34], land boundaries are extracted by connecting the endpoints of broken line segments. Gaps between segments are filled using geometric- and image-based proximity.

For chess board and chess piece recognition [11], line segments detected via CannyLines [30] are excessively broken, and therefore, nearly collinear segments are merged to construct longer segments.

It can be seen that previous efforts treat the merging problem as a means to a specific end. In contrast, we treat it as a single, comprehensive, perceptual problem. As a result, our solution will be applicable to a wider array of computer vision problems compared with previous works that are limited to specific use-cases.

2.3. Line segment grouping and merging techniques

If line detection is performed using the Hough transform, then segments can be grouped together in the Hough voting space to form a base line [13, 14]. Then, segments with proximal base line projections can be merged together. In the Multiscale LSD [16], line segments are detected and iteratively merged across scales to produce longer segments.

Gestalt principles have also been used for deciding mergeability of segments [35]. In the Line Segment Merging (LSM) method from [12], perception-based, adaptive, geometric grouping and merging criteria were introduced for iterative line segment merging. Adaptive, spatio-angular proximity measures were used along with parameters to relax or tighten the merging process. However, use of local geometry alone can lead to globally incorrect mergings. Incremental improvements to the LSM method can be found in human vision inspired multi-scale line segments merging and filtering [36] and a perceptually accurate line segment detection approach (PLSD) [37]. The present work also builds on LSM by incorporating image evidence in addition to geometry in order to produce globally correct mergings.

2.4. Evaluation techniques

Detected vs. ground truth segments. The MCMLSD method [31] produces a probabilistically ranked list of detected segments and



Figure 4. Manhattan bias and incompleteness of ground truth marked in YorkUrbanDB [17]. Many valid segments with strong underlying image gradients are not marked since they do not belong to significant building boundaries. For example, vehicles, pavements, small light poles and small window boundaries are ignored.

employs the Hungarian algorithm to compute one-to-one correspondences with ground truth segments. Such correspondences are then used to compute precision-recall curves. In PLSD [37], the authors also compute precision, recall and F_1 scores but do not use one-to-one correspondences. Instead, they consider a ground truth segment to have been correctly detected if its intersection with detected segments exceeds a threshold.

Deep learning-based detectors [22, 23, 26, 38] use structural average precision (SAP) in which correspondences are established by thresholding squared Euclidean distance between end-points of detected and ground truth segments. Since deep learning models produce probabilistic outputs, the detected segments are also ranked before computing SAP values.

It must be noted that all evaluation metrics based on correspondences require a threshold to determine true and false positives. Metrics that require detection scores for ranking of segments are also only applicable to probability-based detectors. They are not directly applicable to generic detection methods that do not rank the detected segments. Moreover, when detected segments are merged, such scores lose their significance.

A comparison of different evaluation methods is presented in [39] which also introduces a straight line segment distance measure that combines closest distance, modified line segment Hausdorff distance [40, 41] and translation distance between segments to compare two sets of line segments.

Merged vs. detected segments. For quantitative evaluation of the improvement of merged segments over detected segments, the LSM method [12] proposed to compute a Euclidean distance measure between merged and ground truth segments. The same measure was then computed between detected and ground truth segments. The ratio of the two measures was used as the final measure of success of mergeability.

Since precision-recall-based measures involve threshold settings in their evaluation, we propose a parameter-free, novel quantitative evaluation criterion by combining and modifying ideas from [39, 41] and [12].

3. METHODOLOGY

In this section, we present our novel, evidence-based grouping and merging technique. Grouping and merging algorithms use a set of line segments detected by any of the state-of-the-art methods and attempt to produce perceptually more accurate line segments. This means that detected line segments are implicitly considered to be correct. Detected segments might be broken, or some actual lines in the image might be missed. However, a detected line segment is considered to represent a true underlying image structure.

In other words, detected line segments are considered to contain no false positives.

Care must be taken when defining correct and incorrect lines. A line may be detected at the correct image region but with a slightly modified angle due to the numerous numerical approximations such as discretization of image intensities and the size, precision and choice of gradient kernels. While small imperfections in angle of a small line segment might not hold much perceptual weightage, it can significantly affect line segment merging decisions, if not handled properly.

In order to verify merging decisions, we impose the following two conditions:

- (i) **Local:** merging of a pair of line segments should produce a segment that is geometrically close to the longer of the two original line segments, and
- (ii) **Global:** iterative merging of line segments should produce a segment that is supported by the underlying image structure.

We bias decisions toward longer line segments since they hold more perceptual weightage. As highlighted in Fig. 2, existing methods that do not control both local and global behavior of merged line segments can produce globally incorrect line mergings.

In the following two subsections, we discuss grouping and merging along with our proposed line segment evidence approach. Table 1 details the notations used in our explanation. Algorithm 1 describes the whole method in pseudocode. Our algorithm takes two inputs: (i) an image and (ii) line segments extracted from the image by any detector. Our algorithm then applies angular and spatial proximity measures to group line segments. After grouping, pairs in the group are tested and passed through geometry and image evidence-based mergeability criteria to obtain merged line segments.

To start the process, we first sort the detected line segments \mathcal{D} in descending order of length. This is because longer line segments are (i) more meaningful, (ii) more reliable and (iii) come from image regions with continuously strong gradients. The set of sorted segments are denoted by \mathcal{M} and we select the segments in order of decreasing length. Let P denote the selected line segment.

3.1. Grouping line segments

For line segment P , we sequentially construct a spatio-angular proximal group \mathcal{G}_P^M consisting of segments with (i) angles similar to P and (ii) lying in close spatial proximity to P .

Angular proximity. First, an angular proximal group \mathcal{G}_P^M is extracted from \mathcal{M} . The line segments in \mathcal{G}_P^M are the ones that

Table 1. Notation used in the paper.

Input	
\mathcal{D}	Set of detected line segments
w, h	Width and height of input image
User-defined parameters	
$\pi_s \in [0, 1]$	Spatial proximity.
$\pi_t \in \mathbb{Z}^+$	Thickness for line segment evidence
$\pi_r \in \mathbb{Z}^+$	Number of reference points for spatial proximity
User-defined thresholds	
$\tau_\theta > 0^\circ$	Angular proximity
$\tau_o \in [0, 1]$	Overlap tolerance
$\tau_e \in [0, 1]$	Line segment evidence
Processing	
\mathcal{M}	Set of merged line segments
P	Longest line segment
τ_θ^*	Adaptive angular proximity threshold
τ_s^*	Adaptive spatial proximity threshold
\mathcal{G}_P^M	Spatio-angular proximal group with respect to P from \mathcal{M} (used in merging)
\mathcal{G}_P^D	Spatio-angular proximal group with respect to P from \mathcal{D} (used in evidence)
Q	Segment from \mathcal{G}_P^M considered for merging with P .
Line segment pair (P, Q)	
(θ_P, θ_Q)	Angles of pair (P, Q)
(l_P, l_Q)	Lengths of pair (P, Q)
d	Distance between closest endpoints of pair (P, Q)
(m_1, m_2)	Farthest endpoints of pair (P, Q)
M	Merged line segment
p_1	First endpoint (p_{1r}, p_{1c}) of line segment P .
p_2	Second endpoint (p_{2r}, p_{2c}) of line segment P .
p_3	Mid point (p_{3r}, p_{3c}) of line segment P .
q_1	First endpoint (q_{1r}, q_{1c}) of line segment Q .
q_2	Second endpoint (q_{2r}, q_{2c}) of line segment Q .

pass an angular proximity threshold with respect to P as

$$\mathcal{G}_P^M = \{\forall Q \in \mathcal{M} : |\theta_Q - \theta_P| < \tau_\theta\}, \quad (1)$$

where θ_P and θ_Q are the angles of line segment P and Q , respectively. The absolute angular difference for each pair (P, Q) is thresholded so that \mathcal{G}_P^M can have only those line segments which are in close angular proximity with respect to P .

Spatial proximity. For spatial proximity, a user-defined parameter $0 \leq \pi_s \leq 1$ is used to take a fraction of the length of the selected segment as

$$\tau_s^* = \pi_s l_P, \quad (2)$$

where l_P is the length of P and τ_s^* is an adaptively determined spatial proximity threshold calculated separately for every segment P . Compared with shorter lines, longer lines will be allowed to merge with more distant line segments.

Threshold τ_s^* determines a spatial proximity zone around any point p_i belonging to segment P as shown in Fig. 5a. Other segments with an end-point lying within zone Z_i can be considered spatially proximal to segment P . In the following, we describe our approach using three proximal zones along segment P . The method remains unchanged for more proximal zones.

End-points are not enough. The top row of Fig. 5b demonstrates a special case where segments Q_1 and Q_2 both escape spatial proximity checks even though they lie close to segment P . The

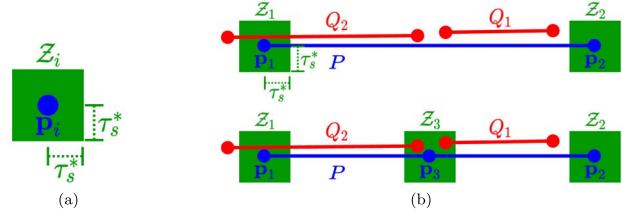


Figure 5. Zone Z_i of spatial proximity around a point p_i on some segment P is defined as the box with absolute distance less than τ_s^* along either axis. Other segments with an end-point lying within zone Z_i can be considered spatially proximal to segment P .

reason is the use of proximal zones only around the end-points of P . Such a situation can be avoided by using one or more additional proximal zones along P as shown in the bottom row using a proximal zone around the mid-point of P .

Each angular proximal segment $Q \in \mathcal{G}_P^M$ can be filtered further in terms of spatial proximity by checking if any end-point of Q lies within any of the proximal zones along P . This corresponds to simply checking pairwise absolute distances between reference points on P and end-points of Q .

Table 1 describes our notation when using the mid-point p_3 as an additional reference point on P . Figure 6 shows the pairwise absolute distances used to decide proximity. Let index i denote points along segment P and let index j denote the two end-points of segment Q . Vertical and horizontal distances can be computed

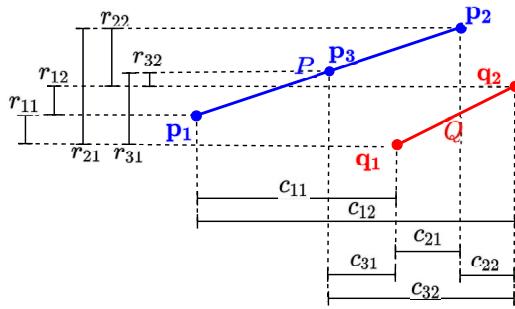


Figure 6. Absolute pairwise distances along rows and columns between three reference points on segment P and two end-points of segment Q.

as

$$r_{ij} = |p_{ir} - q_{jr}| \quad (3)$$

$$c_{ij} = |p_{ic} - q_{jc}|. \quad (4)$$

First, only vertical distances are computed and used to filter line segments in set \mathcal{G}_P^M (obtained via filter (1)) in terms of vertical spatial proximity from P as

$$\mathcal{G}_P^M = \left\{ Q \in \mathcal{G}_P^M : \bigvee_{i,j} (r_{ij} < \tau_s^*) \right\}. \quad (5)$$

The resulting segments in \mathcal{G}_P^M are then filtered again so that segments with endpoints in close horizontal proximity to P are allowed to remain as

$$\mathcal{G}_P^M = \left\{ Q \in \mathcal{G}_P^M : \bigvee_{i,j} (c_{ij} < \tau_s^*) \right\}. \quad (6)$$

The sequence of the three filters (1), (5) and (6) yield the set \mathcal{G}_P^M of line segments that are in close angular and spatial proximity to segment P.

Note on speed-ups Filtering in terms of absolute distance instead of Euclidean distance is faster since it avoids multiplicative operations. Moreover, sequential filtering (first angular (1), then vertical (5), then horizontal (6)) reduces the number of segments to be checked by each subsequent filter. This also results in speeding up the algorithm.

After forming the group \mathcal{G}_P^M , each of the segments can be iteratively considered for merging with segment P. This is described next.

3.2. Merging line segments

After grouping line segments based on angular and spatial proximity, we now introduce merging criteria based on geometry and line segment evidence. In the rest of the paper, whenever two line segments are considered, P will denote the longer segment and Q will denote the shorter segment.

Decision. Each segment Q from the spatio-angular proximal group \mathcal{G}_P^M is considered for merging with segment P. For a pair of line segments (P, Q), overlaps along both coordinate axes are computed as shown in Fig. 7. Normalized overlap is computed as

$$o = \begin{cases} \frac{o_x}{q_x} & o_x > o_y \\ \frac{o_y}{q_y} & \text{otherwise.} \end{cases} \quad (7)$$

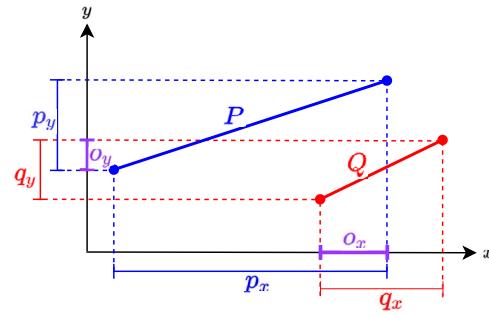


Figure 7. Overlapping geometry of the segments. The overlap o_x along x-axis is the intersection of projection of the segments P and Q along x-axis, i.e. p_x and q_x , respectively; similarly, the overlap o_y along y-axis is the intersection of projection of the segments P and Q along y-axis, i.e. p_y and q_y , respectively; the maximum of both the overlaps is used to decide mergeability of the segments.

To avoid merging of almost parallel, significantly overlapping segments, the normalized overlap is thresholded as

$$0 \leq \tau_0 \quad (8)$$

so that parallel segments (P, Q) are merged only if their overlap is small. For highly overlapping segments, spatial proximity threshold τ_s^* from Equation (2) is scaled down as

$$\tau_s^* = \pi_s l_P (1 - o) \quad (9)$$

in order to merge spatially close segments only.

Customization for line drawings. To allow merging of parallel, highly overlapping segments, such as those detected on both sides of a single segment in line drawings, (i) the overlapping threshold τ_0 can be set to its maximum value, and (ii) spatial proximity threshold τ_s^* can be scaled down so that only nearby segments are merged. Figure 8a demonstrates a typical scenario in line drawings whereby segments P and Q_1 lie on the two sides of a single visible line segment. It is perceptually acceptable to merge them. However, segment Q_2 will have the same normalized overlap as Q_1 and can only be differentiated by considering spatial distance.

After passing overlapping criterion, a spatial proximity filter based on Euclidean distance is applied on the pair (P, Q). Let d_{ij} be the Euclidean distance between reference point p_i on segment P and endpoint q_j on segment Q

$$d_{ij} = \|p_i - q_j\|, \quad (10)$$

where $i \in \{1, \dots, \pi_r\}$ and $j \in \{1, 2\}$. In this paper, the number, π_r , of reference points on P was set equal to either 2 or 3. The reference points on P are defined as follows:

- (i) p_1 and p_2 are the endpoints.
- (ii) p_3 and onwards are successive bisections.

For $\pi_r = 3$, Fig. 8b shows the 6 pairs of distances between reference points on P and endpoints of Q. The closest distance between any reference point on P and endpoint on Q can be calculated as

$$d = \min_{ij} d_{ij}. \quad (11)$$

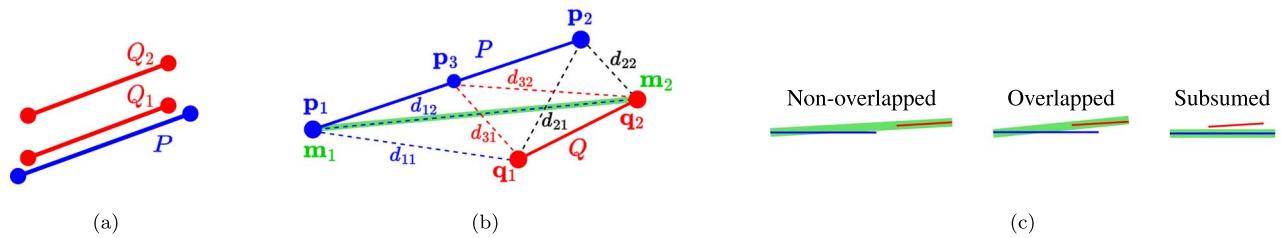


Figure 8. (a) Segments Q_1 and Q_2 have the same normalized overlap with segment P , but only Q_1 can be merged into P in a perceptually satisfactory manner; segment Q_2 is too far, and (b) Six Euclidean distances between reference points of segment P and endpoints of Q represented via dashed lines; closest pair of points corresponds to the minimum of these distances; if P and Q are merged then farthest pair of endpoints, $(\mathbf{m}_1, \mathbf{m}_2)$ will be the endpoints of the merged segment; (c) all possible merging scenarios for two segments P (blue) and Q (red) fall under three cases; for non-overlapped and overlapped mergings, the merged segment (green) consists of the farthest endpoints of P and Q , and for subsumed merging, the shorter segment Q is absorbed into the longer segment P , which becomes the merged segment.

Now d can be used to check whether both line segments are close enough or not using adaptive spatial proximity as

$$d < \tau_s^*. \quad (12)$$

If it passes the criterion, the adaptive angular proximity is computed further.

Mergeability by LSM [12] demonstrated that mere angular and spatial proximity is not enough for making perceptually accurate mergeability decisions. Perception of non-mergeability is proportional to

- (i) relative length of shorter line,
- (ii) relative spatial distance, and
- (iii) angular difference.

Accordingly, perception of non-mergeability (PoN) of (P, Q) can be defined as

$$\text{PoN} = \frac{l_Q}{l_P} + \frac{d}{\tau_s^*}, \quad (13)$$

which is just the sum of the normalized value of the length of the shorter segment and the normalized value of the distance between closest pair of points. Both normalizations are performed by dividing the maximum possible values. The range of values for PoN is from 0 till 2. For mergeable segments, it will be closer to 0, and for non-mergeable segments, it will be closer to 2. An adaptive threshold for angular difference between P and Q can be computed as

$$\tau_\theta^* = \left(1 - \frac{1}{1 + e^{-2(\text{PoN}-1.5)}}\right) \tau_\theta, \quad (14)$$

where τ_θ is a user-defined maximum allowable angle between lines that can be merged. In this way, τ_θ^* becomes a non-linearly controlled, length-and-spatial-distance-adaptive threshold for allowable angle between segments P and Q . For segments considered to be perceptually mergeable (low value of PoN), angular threshold τ_θ^* will be close to its maximum allowable value τ_θ . As a result, merging will be possible even when angular difference is close to its maximum allowance. For perceptually less mergeable segments, angular threshold τ_θ^* will be adaptively restricted to a smaller value so that merging is possible only if the angles match closely. The line segments P and Q will be merged if

$$|\theta_P - \theta_Q| < \tau_\theta^*, \quad (15)$$

Table 2. Differentiating between merging scenarios of Fig. 8c by projecting endpoints of shorter segment Q onto the larger segment P .

Non-overlapped	Both endpoints project outside P
Overlapped	Only one endpoint projects inside P
Subsumed	Both endpoints project inside P

Merged segment Once it is decided that segments P and Q can be merged into a new segment M , the endpoints of M need to be found. Figure 8c demonstrates the three possible merging scenarios relative to the longer segment P . In non-overlapped and overlapped mergings, M is formed by connecting the farthest endpoints of P and Q . In subsumed merging, the shorter segment Q is absorbed into P , and therefore, P becomes the merged segment. The three scenarios can be differentiated by projecting the endpoints of Q onto segment P and selecting the relevant case from Table 2.

Validation After having a merged line segment M with the endpoints as \mathbf{m}_1 and \mathbf{m}_2 , it is further passed through two more validation steps.

Angle: In the first step, the absolute angular difference between the merged line segment M and the original longer line segment P is thresholded as

$$|\theta_P - \theta_M| < \frac{\tau_\theta}{2}, \quad (16)$$

where the angular difference threshold is reduced by half to ensure that the merged line segment M does not move too far from the original segment P .

Line segment evidence: Merged segments that pass the angular threshold are passed through an additional filter that incorporates evidence from existing line segments. The basic idea is to force the merged segment M to lie close to the set of original input segments. This way a newly merged segment cannot be accepted unless it lies over pixels through which a significant number of original line segments pass. As soon as M strays toward regions that have little support from the underlying image, it is rejected.

To gather evidence from the underlying image, we draw a binary evidence image $E_{G_p^D}$ as follows:

- (i) Initialise $E_{G_p^D}$ as an empty array of size $h \times w$ (size of input image).
- (ii) On $E_{G_p^D}$, draw each line segment in G_p^D using line thickness $2\pi_t + 1$ pixels.

Algorithm 1 mergeLines

Data: Set of detected line segments \mathcal{D}
 h, w height and width of the image
User-defined parameters: π_s, π_t, π_r
User-defined thresholds: $\tau_\theta, \tau_o, \tau_e$
Result: Set of merged line segments \mathcal{M}

```

 $\mathcal{M} \leftarrow \mathcal{D}$ 
repeat
   $n \leftarrow |\mathcal{M}|$ 
   $\mathcal{M} \leftarrow$  sort lines in  $\mathcal{M}$  in descending order of length
  for  $P \in \mathcal{M}$  do
     $l_p \leftarrow \|P\|$ 
     $\tau_s^* \leftarrow \pi_s l_p$ 
     $\mathcal{G}_p^M \leftarrow$  spatio-angular proximal group from  $\mathcal{M}$  for segment  $P$  using filters (1), (5) and (6) (used in merging)
     $\mathcal{G}_p^D \leftarrow$  spatio-angular proximal group from  $\mathcal{D}$  for segment  $P$  (used in evidence)
     $E_{\mathcal{G}_p^D} \leftarrow$  evidence image of size  $h \times w$ 
     $\mathcal{R} \leftarrow \emptyset$ 
    for  $Q \in \mathcal{G}_p^M$  do
       $M \leftarrow \emptyset$ 
      if  $l_p < l_Q$  then
        swap( $P, Q$ )
      end
       $d \leftarrow$  closest distance between endpoints of  $P$  and  $Q$  (Equation (11))
       $o \leftarrow$  normalised overlap between  $P$  and  $Q$  (Equation (7))
      if  $o \leq \tau_o$  then
         $\tau_s^* \leftarrow \pi_s l_p$ 
      else
         $\tau_s^* \leftarrow \pi_s l_p (1 - o)$ 
      end
      if  $d < \tau_s^*$  then
         $\tau_\theta^* \leftarrow$  compute adaptive angular threshold (Equation (14))
        if  $|\theta_p - \theta_Q| < \tau_\theta^*$  then
           $M \leftarrow (\mathbf{m}_1, \mathbf{m}_2)$  (Figures 8(b), (c))
          if  $|\theta_p - \theta_M| < \frac{\tau_\theta}{2}$  then
             $e_M \leftarrow$  evidence value of  $M$  from  $E_{\mathcal{G}_p^D}$ 
            if  $e_M > \tau_e$  then
               $P \leftarrow M$ 
               $\mathcal{R} \leftarrow \mathcal{R} \cup Q$ 
            end
          end
        end
      end
    end
  end
   $\mathcal{M} \leftarrow \mathcal{M} \setminus \mathcal{R}$ 
end
until  $|\mathcal{M}| = n$ 

```

The set \mathcal{G}_p^D is used so that unrelated but intersecting segments from the original set \mathcal{D} do not interfere with evidence. By unrelated, we mean segments that are far from P in terms of angular and/or spatial proximity. After computing the binary evidence image $E_{\mathcal{G}_p^D}$, we can compute the evidence e_M supporting the merged segment M as follows:

- (i) Along segment M , count number of on-pixels in $E_{\mathcal{G}_p^D}$.
- (ii) Divide by length of M .

Computed this way, e_M lies between 0 and 1 and can now be thresholded to determine whether M is a valid merged line or not. Accordingly, we employ a user-defined threshold τ_e to filter merged line segments that have strayed away from the underlying original segments. If M passes the criterion

$$e_M > \tau_e, \quad (17)$$

it is considered as a valid merged segment and replaces segment P in set \mathcal{M} so that any further processing takes place on this merged segment. When all the line segments are processed in \mathcal{G}_p^M , a final merged line segment will be reported as M . Now the process can be repeated for the next longest line in set \mathcal{M} until no line segment remains to be merged. Algorithm 1 outlines the pseudocode for the whole method and visualization of the pipeline is shown in Figs 9 and 10.

3.3. Computational complexity

Time complexity. The upper bound of Algorithm 1 is explained here in terms of the number of detected line segments n . The algorithm converges if no merging is performed in the repeat-until-loop. If even a single merging is performed, the loop runs again for $n - 1$ segments. Therefore, in the worst case, this loop executes $n - 1$ times resulting in complexity of order $O(n)$.

Sorting of segments in \mathcal{M} takes $O(n \log n)$ time. The iteration count for the first for-loop depends upon how many segments were merged in the previous iteration. If we consider the worst case, only two segments would have been merged and there will be $n - 1$ segments left including the merged segment. Therefore, the complexity will be of order $O(n)$.

The second for-loop iterates according to the number of line segments in spatial and angular proximal group \mathcal{G}_p^M . In the worst case, it can have $n - 1$ segments, and thus, the complexity will be of order $O(n)$ again.

Therefore, worst case complexity of Algorithm 1 is of order $O(n(\log n + n^2)) = O(n^3)$ in the worst case of all three loops. In practice, however, each of the three loops reduces the number of segments by more than 1. For instance, $|\mathcal{G}_p^M| \ll |\mathcal{M}|$ in the second for-loop. Therefore, the algorithm takes much less than n^3 steps to converge, in practice.

Space complexity. Algorithm 1 has $O(n + hw)$ space complexity due to the groups $\mathcal{M}, \mathcal{G}_p^M, \mathcal{G}_p^D, \mathcal{R}$ and evidence image $E_{\mathcal{G}_p^D}$.

3.4. Comparison with LSM algorithm

Since ELSM is built upon and improves the LSM [12] algorithm, it serves as our main comparison baseline. Here, we demonstrate that what is common in both methods and what is new in ELSM.

Grouping. For forming spatioangular proximal groups, angular proximity is used in the same same way, but spatial proximity is updated to have more reference points. This results in more complete spatial proximal groups. As demonstrated in Fig. 5, the LSM algorithm uses only two proximal zones at the endpoints, while ELSM proposes further proximal zones.

Merging. For merging line segments, the overlapping idea from Fig. 7 and Equations (7)–(9) provides more explicit control over the merging/non-merging of parallel, overlapping segments. Close, overlapping parallel segments may need to be merged for line drawings but not for natural imagery.

As demonstrated in Fig. 8b, closest point pair between two segments is found by utilizing an additional midpoint \mathbf{p}_3 on the longer segment. The LSM method uses the endpoints only. The

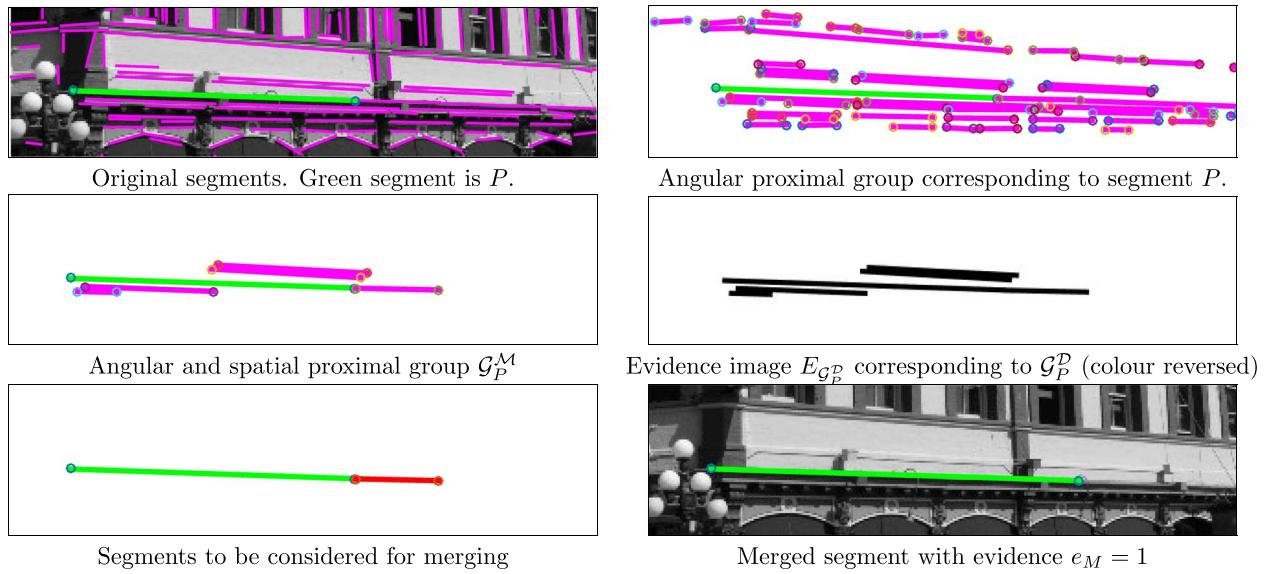


Figure 9. Visualizing line segment grouping and merging steps; spatial and angular proximity measures to group line segments are applied for a selected segment P . Pairs in the group \mathcal{G}_P^M are passed through geometry, perception and evidence-based criterion to get a merged segment M . (Best viewed in colour.)

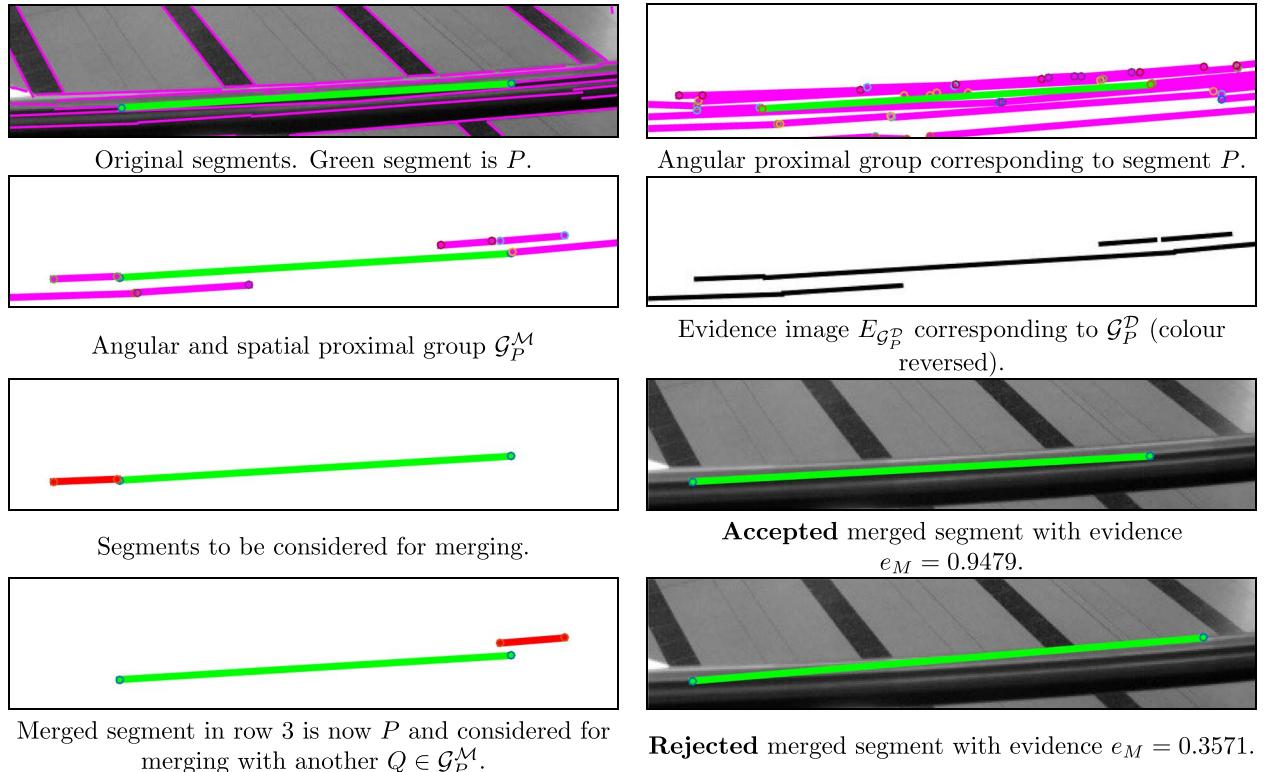


Figure 10. The use of image-based evidence prevents incorrect mergings that mere geometric reasoning cannot prevent. **Rows 1–3:** Merged segment accepted because of underlying support from the original image. **Row 4:** Merged segment rejected because of low support from the underlying image. (Best viewed in colour.)

additional midpoint allows computing more accurate distances between segments. Perception of non-mergeability (Equations (13) and (14)) and angular proximity (Equation (15)) are the same in both methods.

Validation. In order to validate a merged segment, angular validation (Equation (16)) is same in both methods, but ELSM introduces an additional evidential validation. This eliminates a major weakness in the LSM method whose local merging decisions can

eventually lead to a globally incorrect segment with no underlying image support. In ELSM, any merged segment without underlying image evidence is discarded so that globally incorrect segments are never formed.

Complexity. Both algorithms share the same $O(n^3)$ time complexity. Space complexity of LSM is $O(n)$, while ELSM takes $O(n + hw)$ space due to the additional use of the evidence image $E_{G_P^D}$.

4. QUANTITATIVE EVALUATION METHODOLOGY

Given an image and three sets $\mathcal{G}, \mathcal{D}, \mathcal{M}$ of ground-truth, detected and merged segments, respectively, we compare detections \mathcal{D} and mergings \mathcal{M} via their line segment Hausdorff distances from ground-truth \mathcal{G} . We begin by defining a distance function between two segments P and Q following [39]. Let $d_{\perp PQ}$ be the directed perpendicular distance from the closest point on P to line segment Q . We can define undirected closest distance

$$CD(P, Q) = \min(d_{\perp PQ}, d_{\perp QP}), \quad (18)$$

modified line Hausdorff distance

$$MLHD(P, Q) = \min(l_p, l_Q) \sin(|\theta_Q - \theta_p|) \quad (19)$$

and translation distance

$$TD(P, Q) = \frac{d_{11} + d_{12} + d_{21} + d_{22}}{4} - \frac{l_p + l_Q}{4}. \quad (20)$$

They are aggregated to compute the straight line distance

$$ST(P, Q) = CD(P, Q) + \frac{MLHD(P, Q)}{4} + TD(P, Q). \quad (21)$$

Then, following [41], directed Hausdorff distance between sets \mathcal{G} and \mathcal{D} is defined as

$$h(\mathcal{G}, \mathcal{D}) = \frac{1}{\sum_{P \in \mathcal{G}} l_p} \sum_{P \in \mathcal{G}} l_p \min_{Q \in \mathcal{D}} ST(P, Q) \quad (22)$$

and undirected Hausdorff distance can be computed as

$$H(\mathcal{G}, \mathcal{D}) = \max(h(\mathcal{G}, \mathcal{D}), h(\mathcal{D}, \mathcal{G})). \quad (23)$$

For a dataset of N images, each with ground-truth \mathcal{G}_n , detections \mathcal{D}_n and mergings \mathcal{M}_n , following [12], we compute the ratio

$$r = \frac{\sum_{n=1}^N H(\mathcal{G}_n, \mathcal{D}_n)}{\sum_{n=1}^N H(\mathcal{G}_n, \mathcal{M}_n)}. \quad (24)$$

If ratio r is greater than 1, then compared with detected segments, the merged segments are closer to the ground-truth on average. If r is less than 1, then merging is inferior to the detected segments on average.

It must be noted that ratio r represents the improvement of merged segments over detected segments in terms of undirected Hausdorff distances from the ground truth. As mentioned earlier, existing ground truths of line segment datasets are incomplete. However, ratio r neutralizes any incompleteness of ground-truth. This is because instead of just measuring the distance of merged segments \mathcal{M}_n from potentially incomplete ground-truth \mathcal{G}_n , it measures improvement in that distance when compared with the original segments \mathcal{D}_n . Specifically, if a correctly detected segment in \mathcal{D}_n is missing in the ground-truth \mathcal{G}_n , it will be unfairly penalised by the numerator in Equation (24). However, the corresponding merged segment in \mathcal{M}_n will also be missing in \mathcal{G}_n and will be similarly penalized in the denominator. Therefore, potential incompleteness in ground-truth will be neutralized. In addition, this metric does not require any thresholds.

Table 3. User-defined parameters and thresholds used to generate all results on YorkUrbanDB images (**York**) and line drawing images (**LD**) in this paper.

Name	Use	York	LD
π_s	Spatial proximity	0.05	0.2
π_t	Thickness	1	1
π_r	Reference points	2	3
τ_θ	Angular proximity	5°	5°
τ_o	Overlap tolerance	0.6	1
τ_e	Segment evidence	0.8	0.6

5. EXPERIMENTS

Datasets. We test our novel, perceptually driven and evidence-based grouping and merging ELSM technique on images from YorkUrbanDB [17]. It includes 102 images (45 indoor, 57 outdoor) of urban environments. The size of each image is 640 × 480 pixels. On average, 118 ground truth line segments are marked per image. However, the ground truth segments are mostly marked in Manhattan directions so that other valid line segments are not marked. As a result, the ground truth underestimates the actual number of line segments that can reduce the validity of quantitative evaluations.

We also test on images of hand-drawn cadastral maps so that the performance of different approaches on line drawings can be compared.

Parameters and thresholds. All results shown in this paper were generated using the parameter and threshold values reported in Table 3. The second last column contains values used for YorkUrbanDB images and the last column is for line drawings.

Detection baselines. To exhibit the general applicability of our proposed ELSM method, we apply it over the outputs of the following four different line segment detectors:

- Line Segment Detector (LSD) [15]¹
- Edge Drawing Lines (EDLines) [20]²
- Markov Chain Marginal Line Segment Detector (MCMLSD) [31]³ and
- Fast Line Segment Detector (FLD) [21]⁴.

Segments obtained from each detection method are input to the merging pipelines of both LSM [12]⁵ and the proposed ELSM approach. All detection and merging statistics are averaged over the four detection methods.

Since deep learning for line segment detection itself is a nascent area of research, there are no existing deep segment merging models. We analyse the effect of the proposed ELSM approach on the outputs of the following four state-of-the-art deep line segment detection models:

- AFD [22]⁶ : a CNN-based line segment detector trained on YorkUrbanDB and Wireframe [18] datasets,
- LETR [23]⁷ : a transformer-based line segment detector trained on YorkUrbanDB,

¹ <http://www.ipol.im/pub/art/2012/gjmr-lsd/>

² https://github.com/CihanTopal/ED_Lib

³ <https://www.elderlab.yorku.ca/mcmlds/>

⁴ Python implementation within OpenCV.

⁵ http://faculty.pucit.edu.pk/nazarkhan/work/line_merging/LSMDemo.zip

⁶ https://github.com/cherubicXN/afm_cvpr2019

⁷ <https://github.com/mlpc-ucsd/LETR>

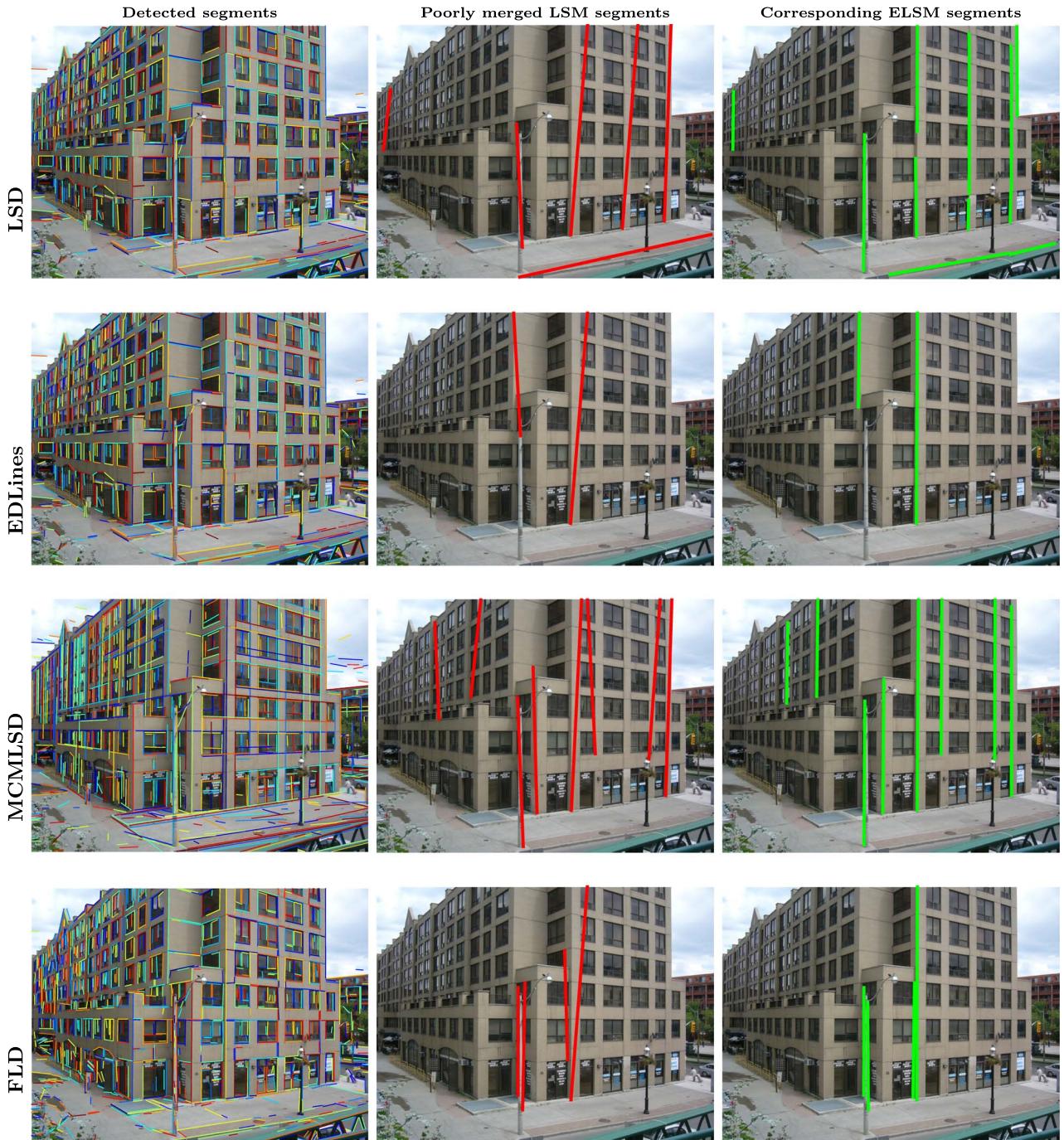


Figure 11. Benefit of image evidence-based merging; **Column 1:** segments detected by four detectors LSD [15], EDLines [20], MCMLSD [31] and FLD [21] on image 90 from YorkUrbanDB; they serve as input to the merging methods; **Column 2:** poorly merged segments by LSM [12] since it considers line geometry only; **Column 3:** use of image evidence in addition to line geometry allows the proposed ELSM method to prevent poor mergings.

- Mobile-LSD [26]⁸: a light-weight model with real-time detection speed, and
- F-Clip [38]⁹: an efficient and accurate fully convolutional line parsing network.

Machine and software specifications. All experiments in this paper were carried out on an HP Envy Model 15T-DR100 Core i7 10th generation system using Matlab R2015a software.

⁸ <https://github.com/navervision/mlsd>

⁹ <https://github.com/Delay-Xili/F-Clip>

6. RESULTS AND ANALYSIS

We present both qualitative and quantitative evaluations of existing methods as well as the proposed ELSM method.

The first columns in Figs 11 and 12 demonstrate weaknesses of the four detection methods, i.e. breaking up perceptually connected lines into multiple segments. The second columns demonstrate weakness of merging via LSM. Since LSM considers geometric cues only, it lacks global control over the merged segments. The segments drawn as thick red lines have incrementally strayed away from the underlying image data. In contrast, the third

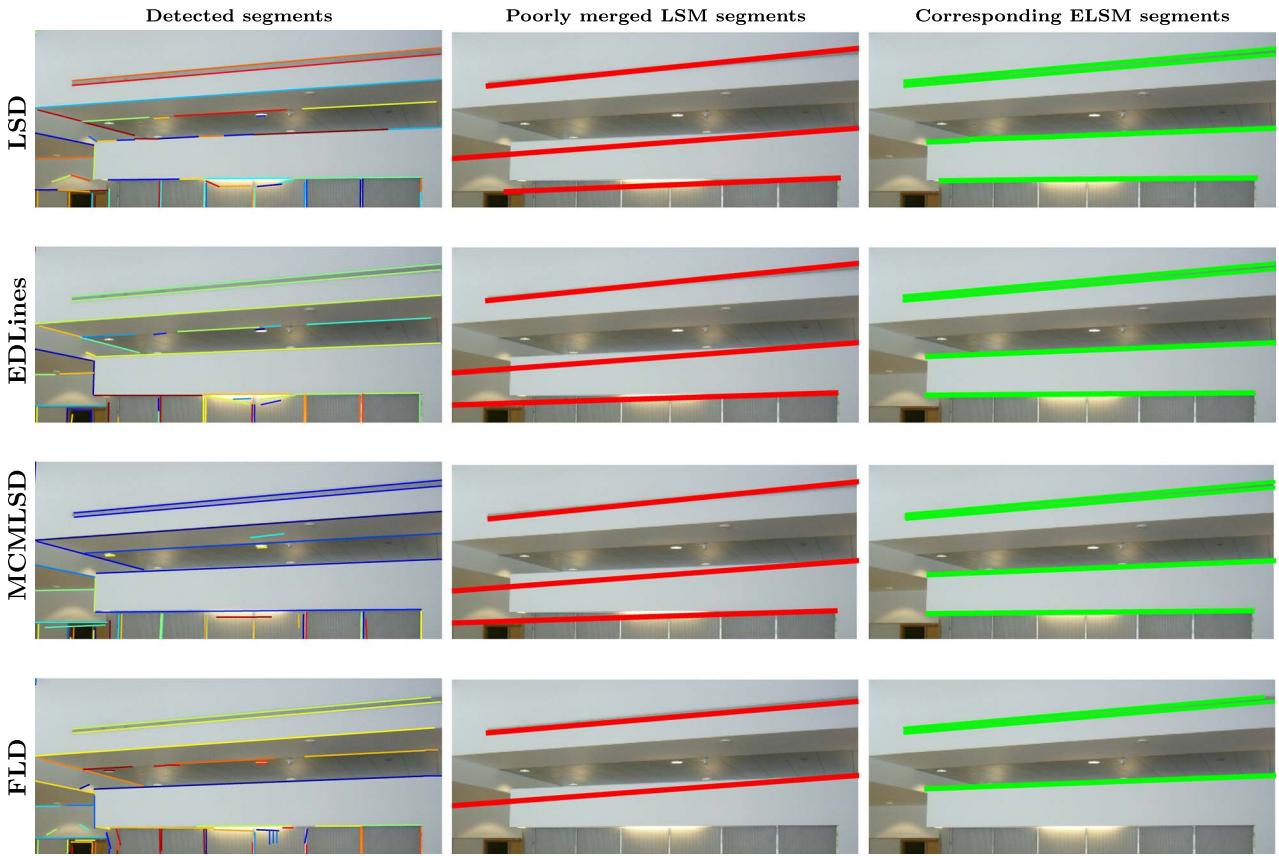


Figure 12. Benefit of image evidence-based merging demonstrated on a cropped portion of image 5 from YorkUrbanDB; the proposed ELSM method does not allow merged segments to move away from detected segments.

columns demonstrate the effectiveness of the proposed ELSM method. Since it considers geometric- as well as image-based cues, it continually rejects any merged segment that starts to stray away from underlying image data. As a result, the merged segments remain faithful to underlying image data and retain perceptual validity as can be seen by comparing the thick green segments in the third columns to the thick red segments in the second columns.

We show in Table 4 how the proposed ELSM method yields a middle ground between over-segmentation of conventional line segment detectors and excessive merging of the LSM method. For all four conventional detectors considered in this study, ELSM reduces the average number of segments indicating that it merges the detected segments. However, it does not merge as much as LSM on average. This is not surprising since LSM does not consider image evidence that could have enabled it to reject incorrect mergings. Table 4 also shows that recent deep learning-based detectors do not suffer from over-segmentation since they learn from human-marked ground-truth annotations that are not broken at intersections and junctions. As a consequence, they do not require as much merging of their segments. However, the weaknesses of existing deep models are highlighted in Figure 14.

The over-segmentation phenomenon of LSD has been addressed by Multiscale LSD [16], which detects and iteratively merges LSD segments at multiple scales. The method produces longer segments compared with LSD, but results are not faithful to the underlying image data. Figure 13 shows Multiscale LSD and the proposed ELSM (on LSD) results on four images from YorkUrbanDB. It can be observed that Multiscale LSD still produces many false positives, while all ELSM segments are true

Table 4. Comparison of mean number of segments per image (rounded to the nearest integer) for all 102 images of YorkUrbanDB averaged over four conventional detection methods LSD [15], EDLines [20], MCMLSD [31] and FLD [21] and four deep learning models AFD [22], LETR [23], Mobile-LSD [26] and FClip [38]; the proposed ELSM approach reduces the overmerging phenomenon of LSM [12] for conventional detectors since they require merging due to over-segmentation; deep learning-based detectors do not oversegment and therefore do not require significant merging.

	Conventional	Deep learning
Detected	637	121
LSM [12]	556	105
ELSM	613	119

representations of the scene since any incorrect mergings are immediately rejected.

Deep learning approaches. Figure 14 shows line segments detected by four state-of-the-art deep learning-based models trained on urban and indoor imagery. We make the following observations.

Observation 1: Deep models do not suffer from over-segmentation seen in conventional gradient-based detectors since they are trained on appropriately segmented ground truth. Therefore, our proposed merging approach will not improve their output significantly. This is corroborated by Table 4 where it can be seen that neither LSM nor ELSM significantly reduced the average numbers of segments detected by deep models.

Observation 2: Using current datasets and their annotations with a strong Manhattan bias, deep models tend to yield

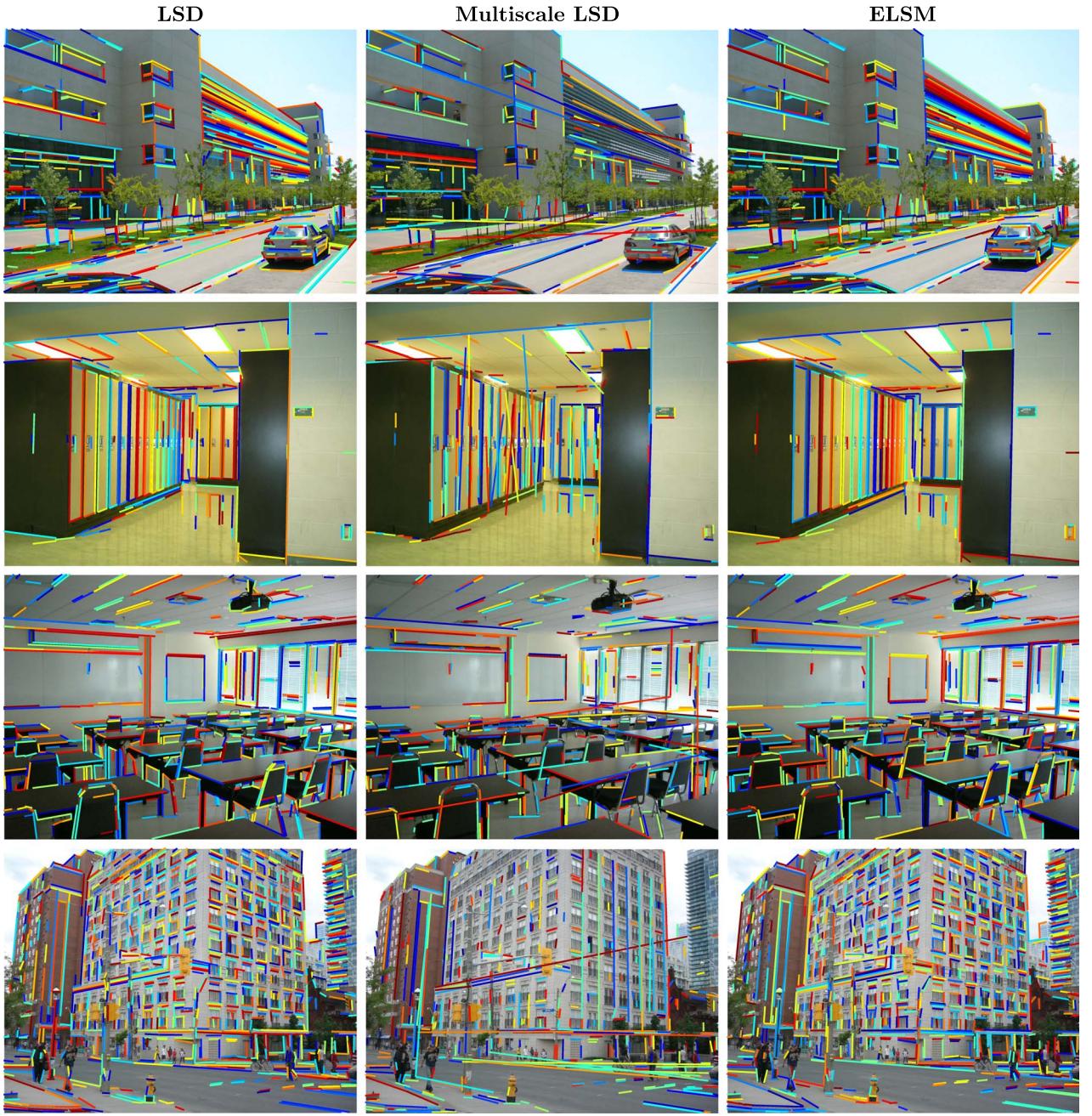


Figure 13. Weakness of applying Multiscale LSD [16] on images 3, 8, 10 and 102 from YorkUrbanDB; **Column 1:** LSD [15] segments; **Column 2:** Multiscale LSD segments; **Column3:** ELSM results on LSD segments, and many false positives can be observed in Multiscale LSD segments, while our proposed ELSM approach reports segments which are faithful to the underlying image data; for example, in the last row, windows in the building facade do not support long line segments and ELSM, accordingly, yields smaller segments.

inconsistent false negatives and false positives that can be hard to explain away.

Observation 3: Such dataset bias causes poor generalization that a simple gradient-based detector will not suffer from. For example, column 4 shows results on a map image containing grid lines. Such an image does not belong to the distribution of the training sets of deep learning models that contain predominantly urban images.

Quantitative evaluation. In Table 4, it can be seen that averaged over four different conventional line detection methods, the proposed ELSM method did not suffer from the over-merging phenomenon of LSM [12]. This is true for four different deep learning

models as well, but not as significantly. This is because deep learning models are trained on appropriately segmented ground truth and therefore produce segments that do not require much merging. Accordingly, ELSM did not perform unnecessary merging when it was not required.

In Table 5, we report merging success r computed from Equation (24). Here too, compared with segments merged by LSM [12], ELSM produces merged segments that are closer to ground-truth than detected segments on average.

Runtime statistics. Average runtime in seconds per image for detection and merging techniques is reported in Table 6 for YorkUrbanDB [17]. Ignoring the fact that ELSM is implemented

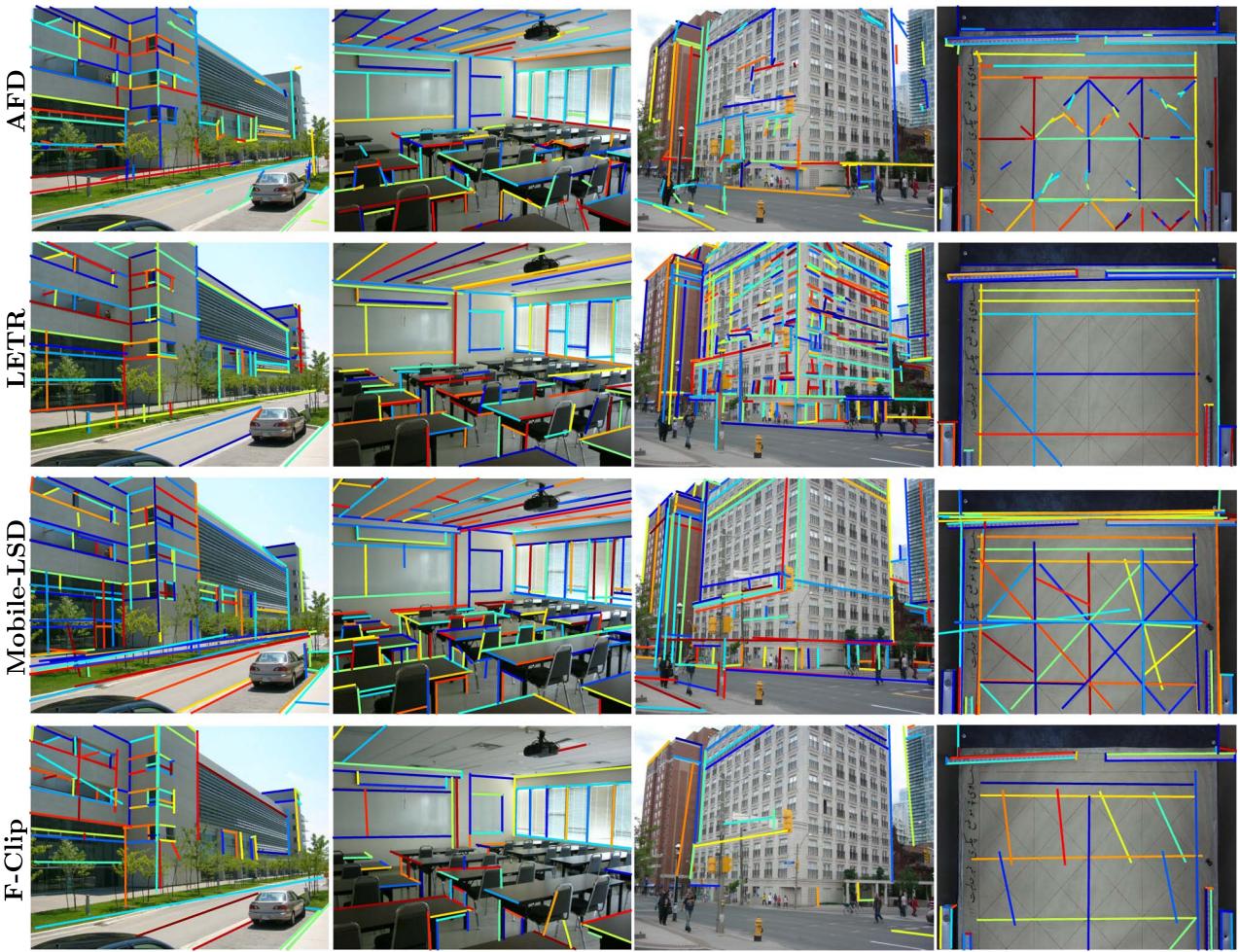


Figure 14. Deep learning-based line detection models trained on urban scenes; **Observation 1:** They do not suffer from over-segmentation seen in conventional gradient-based detectors. Therefore, our proposed merging approach will not improve their output significantly; **Observation 2:** However, using current datasets and their annotations with a strong Manhattan bias, they tend to yield inconsistent false negatives and false positives that can be hard to explain away; **Observation 3:** Such dataset bias causes poor generalization (for example, on a line drawing in column 4) that a simple gradient-based detector will not suffer from.

Table 5. Average merging success r for LSM and ELSM corresponding to the scenario of Table 4; proposed ELSM method yields segments that are closer to the ground-truth on average.

	Conventional	Deep learning
LSM [12]	0.9982	0.9790
ELSM	1.0167	0.9966

in Matlab which is much slower than compiled C++ code, its merging times are not suitable for real-time applications yet. However, the qualitative superiority of ELSM output makes it useful for non-real-time applications. For instance, it can be used to construct more complete ground-truths for training deep learning models for line segment detection. This is important since current ground-truths are incomplete. ELSM can also be useful for document understanding tasks such as cadastral map extraction [42, 43].

Line drawings. Line drawings are unlike natural/urban imagery. One significant difference is that their lines are in the form of roof-edges in contrast to the step-edge lines found in natural/urban imagery. Such roof-edges have large gradients on both sides and produce two detections per line. It is important, therefore, to

Table 6. Average runtime (seconds/image) for detection and merging techniques computed over 102 YorkUrbanDB images with resolution 640×480 ; LSD and EDLines are C++ implementations, FLD is a Python implementation within OpenCV and MCMILSD, LSM and ELSM are Matlab implementations.

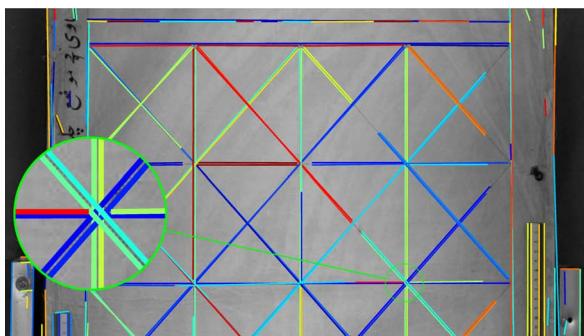
Detection			Merging		
LSD	EDLines	MCMILSD	FLD	LSM	ELSM
0.05	0.002	48.46	0.037	3.7	4.8

tune the parameters and thresholds in Table 3 to work well for line drawings.

The most important threshold is the overlap ratio τ_0 that needs to be set to its maximum value of 1 to allow merging of close, parallel segments as exhibited in Fig. 15 for the case of hand-drawn reference grid lines in a historical cadastral map.

The next important parameter is the number of reference points π_r to decide spatial proximity of line segments. Using only two reference points ($\pi_r = 2$) on the longer segment P in Equations (3) and (4) leads to an incomplete spatio-angular proximal group G_p^M , which leaves some segments unmerged. Using three reference points, ($\pi_r = 3$) forms a more complete proximal group

Without merging overlapping parallel segments



After merging parallel overlapping segments

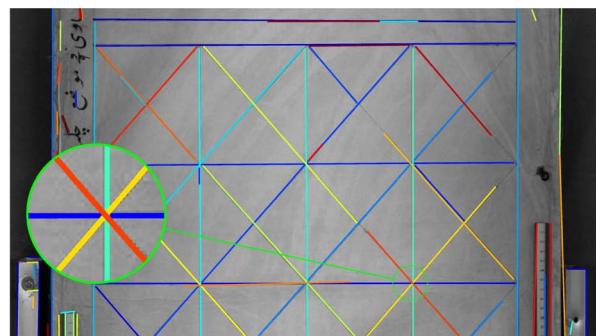


Figure 15. Left: Line drawings should be treated differently from natural imagery; gradient change on both sides of a line yields two parallel overlapping segments; such segments are perceptually redundant but geometrically and computationally problematic; **Right:** They can be allowed to merge by setting our overlap threshold τ_0 to the maximum possible value of 1.

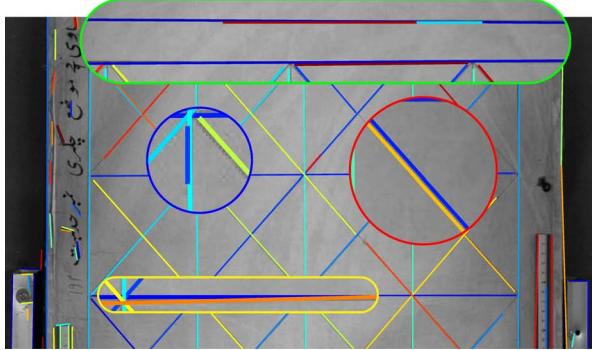
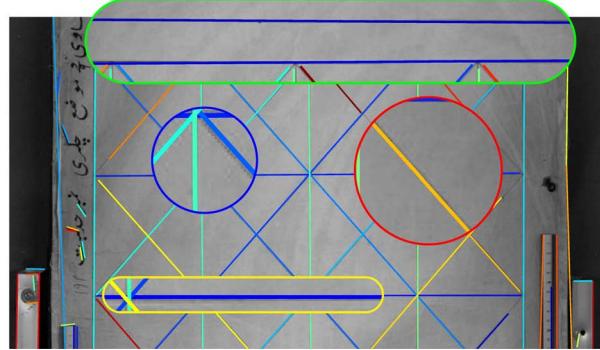
Using two reference points on P Using three reference points on P 

Figure 16. Benefit of more reference points; **Left:** Result from Fig. 15 (right) annotated to show that small line segments remain problematic; using only two reference points ($\pi_r = 2$) on the longer segment P in Equations (3) and (4) leads to the formation of an incomplete spatial and angular proximal group G_p^M . This causes some line segments (shown magnified) to remain unmerged leading to 81 line segments; **Right:** Using three reference points ($\pi_r = 3$) forms a more complete proximal group G_p^M that leads to 68 merged line segments that are less redundant and perceptually improved.

G_p^M leading to less redundant and perceptually improved merged segments as shown in Fig. 16.

Figure 17 demonstrates the benefit of using image evidence in obtaining globally accurate mergings. LSM [12] considers the geometry of line segments only and can therefore iterate to a merged segment with no underlying image support. This is prevented in the proposed ELSM method by requiring every intermediate as well as final merged segment to have underlying image support via condition (17). As a result, merged segments never iterate away from detected segments. Not surprisingly, increasing π_r from 2 to 3 had no impact on visual as well as quantitative results on YorkUrbanDB images since they are not characterized by roof-edges.

7. CONCLUSION

We have described a post hoc solution to the over-segmentation phenomenon of gradient-based line segment detectors. The proposed evidence-based line segment merging algorithm ELSM restricts geometrically merged segments to have underlying image support as well. On images from YorkUrbanDB, we have shown that our method yields perceptually accurate segments and that incremental, local merging does not stray toward globally incorrect segments.

While deep learning models for line segment merging do not exist yet, models for line segment detection already yield reasonably segmented lines. As a consequence, the proposed ELSM method does not unnecessarily merge the output of deep models.

However, we do highlight how existing deep models suffer from the Manhattan bias of line segment datasets and give poor generalization that a gradient-based line segment detector will not suffer from.

We also experiment with parameter settings for line drawings that are characterised by roof-edges in contrast to natural imagery characterized by step-edges. Perceptually relevant segments can be obtained by allowing the parallel segments on both sides of roof-edges to merge into one segment.

Finally, we describe a novel evaluation measure for line segment merging algorithms based on Hausdorff distance. Evaluation of our results demonstrates that ELSM merges segments in a way that brings them closer to ground truth marked according to human perception.

As for future directions of research, in order for ELSM to be applicable in real-time applications, the running time needs to be improved. It will also be interesting to measure the benefit of ELSM in 3D reconstruction, document analysis and other applications that rely on line segments.

FUNDING

Higher Education Commission National Research Program for Universities (8329 to N.K.).

DATA AVAILABILITY

Source code and demo for the proposed ELSM method can be found at <https://nailahamidpucit.github.io/ELSM/>.

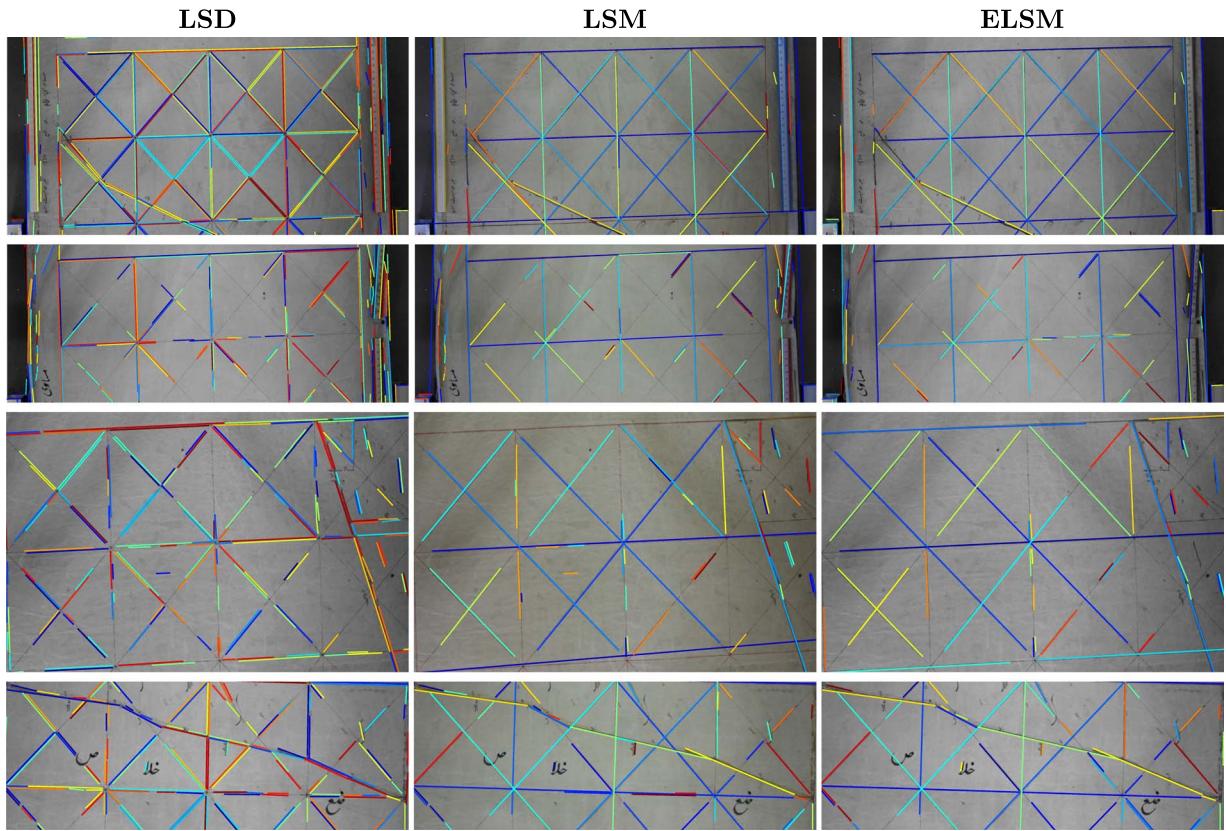


Figure 17. Line drawings require the merging of significantly overlapping parallel line segments returned by LSD [15] due to gradient change on both sides of each line; the LSM [12] method sometimes converges to globally incorrect segments with no underlying image support; use of image evidence prevents the proposed ELSM method from straying away from image gradients; this results in globally correct segment merging.

REFERENCES

1. Hofer, M., Wendel, A. and Bischof, H. (2013). Incremental line-based 3D reconstruction using geometric constraints. In British Machine Vision Conference (BMVC), Bristol, England, September 9–13. BMVA Press, Durham, UK.
2. Micusik, B. and Wildenauer, H. (2017) Structure from motion with line segments under relaxed endpoint constraints. *Int. J. Comput. Vis.*, **124**, 65–79.
3. Zhang, L. and Koch, R. (2014) Structure and motion from line correspondences: representation, projection, initialization and sparse bundle adjustment. *J. Vis. Commun. Image Represent.*, **25**, 904–915.
4. Holt, R.J. and Netravali, A.N. (1996) Uniqueness of solutions to structure and motion from combinations of point and line correspondences. *J. Vis. Commun. Image Represent.*, **7**, 126–136.
5. Al-Shahri, M. and Yilmaz, A. (2014) Line matching in wide-baseline stereo: a top-down approach. *IEEE Trans. Image Process.*, **23**, 4199–4210.
6. Chen, M., Yan, S., Qin, R., Zhao, X., Fang, T., Zhu, Q. and Ge, X. (2021) Hierarchical line segment matching for wide-baseline images via exploiting viewpoint robust local structure and geometric constraints. *ISPRS J. Photogramm. Remote Sens.*, **181**, 48–66.
7. Lu, J., Fang, Z., Gao, Y. and Chen, J. (2021) Line-based visual odometry using local gradient fitting. *J. Vis. Commun. Image Represent.*, **77**, 103071, 1–10.
8. Yoo, J.H., Lee, S.-W., Park, S.-K. and Kim, D.H. (2017) A robust lane detection method based on vanishing point estimation using the relevance of line segments. *IEEE Trans. Intell. Transp. Syst.*, **18**, 3254–3266.
9. Wildenauer, H. and Vincze, M. (2007). Vanishing point detection in complex man-made worlds. In 14th Int. Conf. on Image Analysis and Processing (ICIAP), Modena, Italy, September 10–14, pp. 615–622. IEEE.
10. Li, Z., Meng, C., Zhou, F., Ding, X., Wang, X., Zhang, H., Guo, P. and Meng, X. (2019) Fast vision based autonomous detection of moving cooperative target for unmanned aerial vehicle landing. *J. Field Robot.*, **36**, 34–48.
11. Czyzowski, M.A., Laskowski, A. and Wasik, S. (2020) Chessboard and chess piece recognition with the support of neural networks. *Foundations of Computing and Decision Sciences*, **45**, 257–280.
12. Hamid, N. and Khan, N. (2016) LSM: perceptually accurate line segment merging. *J. Electron. Imaging*, **25**, 061620, 1–11.
13. Bandera, A., Pérez-Lorenzo, J.M., Bandera, J. and Sandoval, F. (2006) Mean shift based clustering of Hough domain for fast line segment detection. *Pattern Recognit. Lett.*, **27**, 578–586.
14. Jang, J.-H. and Hong, K.-S. (2002) Fast line segment grouping method for finding globally more favorable line segments. *Pattern Recognit.*, **35**, 2235–2247.
15. Von Gioi, R.G., Jakubowicz, J., Morel, J.-M. and Randall, G. (2012) LSD: a line segment detector. *Image Process. Line*, **2**, 35–55.
16. Salaun, Y., Marlet, R. and Monasse, P. (2016). Multiscale line segment detector for robust and accurate SfM. In Proc. of the 23rd Int. Conf. on Pattern Recognition (ICPR), Cancun, Mexico, December 4–8, pp. 2000–2005. IEEE.
17. Denis, P., Elder, J. H. and Estrada, F. J. (2008). Efficient edge-based methods for estimating Manhattan frames in urban imagery. In European Conf. on Computer Vision (ECCV), Marseille, France, October 12–18, pp. 197–210. Springer.

18. Huang, K., Wang, Y., Zhou, Z., Ding, T., Gao, S., and Ma, Y. (2018). Learning to parse wireframes in images of man-made environments. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, June 18–23, pp. 626–635. IEEE.
19. Duda, R.O. and Hart, P.E. (1972) Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM*, **15**, 11–15.
20. Akinlar, C. and Topal, C. (2011) EDLines: a real-time line segment detector with a false detection control. *Pattern Recogn. Lett.*, **32**, 1633–1642.
21. Liu, D., Wang, Y., Tang, Z. and Lu, X. (2013). A robust and fast line segment detector based on top-down smaller eigenvalue analysis. In *Fifth Int. Conf. on Graphic and Image Processing*, Hong Kong, China, October 26–27, Vol. **9069**, pp. 906916. SPIE, Bellingham, WA, USA.
22. Xue, N., Bai, S., Wang, F., Xia, G.-S., Wu, T. and Zhang, L. (2019). Learning attraction field representation for robust line segment detection. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, California, USA, June 15–20, pp. 1595–1603. IEEE, New Jersey.
23. Xu, Y., Xu, W., Cheung, D. and Tu, Z. (2021). Line segment detection using transformers without edges. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, held virtually, Nashville, TN, USA, June 19–25, pp. 4257–4266. IEEE.
24. Tong, X., Ying, X., Shi, Y., Wang, R. and Yang, J. (2022). Transformer Based Line Segment Classifier with Image Context for Real-Time Vanishing Point Detection in Manhattan World. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, June 19–25, pp. 6093–6102. IEEE.
25. Teplyakov, L., Erlygin, L. and Shvets, E. (2022) LSDNet: trainable modification of LSD algorithm for real-time line segment detection. *IEEE Access*, **10**, 45256–45265.
26. Gu, G., Ko, B., Go, S., Lee, S.-H., Lee, J. and Shin, M. (2022). Towards Light-weight and Realtime Line Segment Detection. In *Proc. of the 36th AAAI Conf. on Artificial Intelligence*, held virtually , February 22–March 1, pp. 726–734. AAAI Press, Palo Alto, CA, USA.
27. Huang, S., Qin, F., Xiong, P., Ding, N., He, Y. and Liu, X. (2020). TP-LSD: Tri-Points based line segment detector. In *Computer Vision–ECCV*, 16th European Conf., Glasgow, UK, August 23–28, Proceedings, Part XXVII 16, pp. 770–785. Springer.
28. Zhang, H., Luo, Y., Qin, F., He, Y. and Liu, X. (2021). ELSD: Efficient Line Segment Detector and Descriptor. In *Proc. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Montreal, Canada, October 11–17, pp. 2969–2978. IEEE.
29. Guerreiro, R.F. and Aguiar, P.M. (2012) Connectivity-enforcing Hough transform for the robust extraction of line segments. *IEEE Trans. Image Process.*, **21**, 4819–4829.
30. Lu, X., Yao, J., Li, K. and Li, L. (2015). CannyLines: a parameter-free line segment detector. In *IEEE Int. Conf. on Image Processing (ICIP)*, Canada, September 25–28, pp. 507–511. IEEE.
31. Almazan, E. J., Tal, R., Qian, Y. and Elder, J. H. (2017). MCMLSD: a dynamic programming approach to line segment detection. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Honolulu, Hawaii, July 21–26, pp. 2031–2039. IEEE.
32. Zaheer, A., Rashid, M. and Khan, S. (2012). Shape from angle regularity. In *12th European Conference on Computer Vision (ECCV)*, Florence, Italy, October 7–13. Springer, Berlin Heidelberg, Germany.
33. Viero, T. and Jeulin, D. (1995) Morphological extraction of line networks from noisy low-contrast images. *J. Vis. Commun. Image Represent.*, **6**, 335–347.
34. Kim, N.W., Lee, J., Lee, H. and Seo, J. (2014) Accurate segmentation of land regions in historical cadastral maps. *J. Vis. Commun. Image Represent.*, **25**, 1262–1274.
35. Trucco, E. and Verri, A. (1998) *Introductory techniques for 3-D computer vision* (Vol. **201**). Prentice Hall Englewood Cliffs, New Jersey.
36. Zheng, X., Xiang, C., Lu, H., Chen, B. M. and Lee, T. H. (2018). Human vision inspired multi-scale line segments merging and filtering. In *2018 IEEE 14th Int. Conf. on Control and Automation (ICCA)*, Anchorage, Alaska, USA, June 12–15, pp. 259–264. IEEE.
37. Yu, Q., Xu, G., Cheng, Y. and Zhu, Z.H. (2020) PLSD: a perceptually accurate line segment detection approach. *IEEE Access*, **8**, 42595–42607.
38. Dai, X., Gong, H., Wu, S., Yuan, X. and Ma, Y. (2022) Fully convolutional line parsing. *Neurocomputing*, **506**, 1–11.
39. Wirtz, S. and Paulus, D. (2016) Evaluation of established line segment distance functions. *Pattern Recognit. Image Anal.*, **26**, 354–359.
40. Huttenlocher, D.P., Klanderman, G.A. and Rucklidge, W.J. (1993) Comparing images using the Hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, **15**, 850–863.
41. Gao, Y. and Leung, M.K. (2002) Line segment Hausdorff distance on face matching. *Pattern Recognit.*, **35**, 361–371.
42. Khan, S., Aslam, A., Ahmad, S., Rehan, A., Gul, A., Alam, U., Naqvi, H., Bukhari, Z., Iqbal, I. and Sherdil, K. (2011) *Mapping rural Pakistan: bottlenecks and solutions*. International Growth Centre, London, UK.
43. Iftikhar, T. and Khan, N. (2023) Background grid extraction from historical hand-drawn cadastral maps. *Int. J. Doc. Anal. Recognit.* 1–14.