IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 12, 2019, accepted February 23, 2019, date of publication February 28, 2019, date of current version March 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2902194

Eliminating Lining Seams in Tunnel Concrete Crack Images via Line Segments' Translation and Expansion

ZHONG QU[1,2], SI-QI CHEN[1], YU-QIN LIU[1], AND LING LIU[3]

[1]School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[2]College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[3]College of Mobile Telecommunications, Chongqing University of Posts and Telecommunications, Chongqing 401520, China

Corresponding author: Zhong Qu (quzhong@cqupt.edu.cn)

This work was supported by the Chongqing Basic and Frontier Research Project under Grant cstc2015jcyjBX0090.

ABSTRACT Cracks are the main form of tunnel complications and have drawn much attention in current tunnel safety monitoring. However, automatic crack detection cannot accurately extract cracks from tunnel concrete lining surfaces because of the inference of noises like lining seams, which have similar gray values and textures to cracks, and increase the topological complexity of cracks. In this paper, we proposed a novel method for eliminating lining seams in tunnel concrete crack images, to address the above-mentioned issues. Our contributions are shown as follows: 1) classified arbitrary line segments to better distinguish the edges on the cracks and the seams, and proposed uniform *k*-divided angle model to get accurate classification and 2) introduced two new principles to help effectively eliminate lining seams through the information of translation direction and expansion width of all line segments. The experimental results proved the superior precision and efficiency of our method compared with the existing methods.

INDEX TERMS Lining seams elimination, line segments classification, translation direction, expansion width, crack extraction.

I. INTRODUCTION

In recent years, tunnels have an important role in current transportation infrastructure and tunnel safety monitoring has increasingly received public attention. During the lifetime of tunnels, there are many minor or major complications, such as cracks, deformations, water leakage, etc., which may seriously threaten the safety of tunnels. Cracks are the main issues of tunnels. They not only reduce the solidity and carrying capacity of lining structure but also become liquid leakage channels, which results in corrosion of steel bars and reduction of its supporting capacity and durability [1], [2]. With the help of computer vision, the technique of crack identification based on digital image processing can be used for qualitative and quantitative analysis of tunnel crack defects [3], [4], which avoids human error under manual instrument detection and guarantees the people's safety at the same time.

As shown in Fig.1, crack detection is an important part of the whole crack identification module, it contains image preprocessing, crack extraction and lining seams elimination, thus it provides essential prerequisites and basis for crack analysis, therefore it has been explored by experts or scholars for decades.

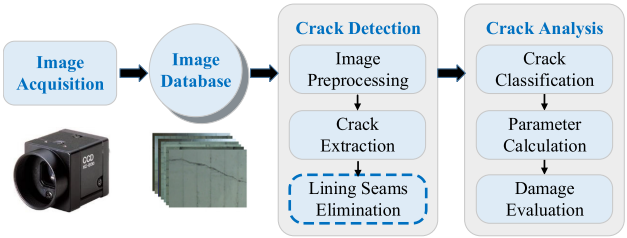


FIGURE 1. Pipeline of the crack identification.

Cha *et al.* [5] proposed a vision-based method using a deep architecture of convolutional neural networks (CNNs) for detecting concrete cracks. The prepared training image set is fed into a CNN to build a CNN classifier for separating crack from intact concrete images in the validation set. Its main advantage is requiring no feature extraction and calculation compared to traditional approaches. Yamaguchi *et al.* [6]–[8]

The associate editor coordinating the review of this manuscript and approving it for publication was Huazhu Fu.

considered the termination and skipped conditions to improve the speed of crack detection on the basis of the proposed algorithm based on the percolation model. Although the local details of the large image and the tiny cracks can be detected in a high accuracy, the process of percolation is inefficient because every pixel needs to be processed. In order to significantly reduce the amount of calculation, Qu *et al.* [9], [10] proposed the second accelerated percolation algorithm by improving the conditions in the percolation, using a de-noising method based on the percolation model, and pre-extracting the dark pixel via sliding window to reduce percolation redundancy of background noise. Zou *et al.* [11], [12] proposed a geodesic-based shadow removal method by analyzing the statistic property and illumination compensation of geodesic levels. Then, tensor voting is used to establish a crack probability map and crack seeds can be sampled to derive minimum spanning trees and determine the real cracks. Shi *et al.* [13] proposed a crack detection method using random structured forests, which can suppress noises by learning the inherent structured information of cracks thus cracks with intensity inhomogeneity can be detected well. Chen *et al.* [14] proposed an approach termed minimal path propagation with backtracking (MPP-BT) and utilized the information in the process of backtracking from reached points, which can efficiently extract geometrically curve-like structures by finding the path with minimal accumulated cost between two given endpoints.

However, these crack detection algorithms target at extracting the simple cracks, i.e., no severe interference of background noise such as lining seams, or they cannot accurately extract cracks from tunnel concrete lining images, because in that kind of environment, cracks are easily connected with lining seams and they also have similar gray values to the lining seams.

(*a*) (*b*)

FIGURE 2. Tunnel concrete structure. (a) The tunnel passage. (b) The tunnel concrete surface with seams and cracks.

Fig. 2(a) shows the tunnel passage with lining structure, and Fig. 2(b) shows the close-up view of the tunnel concrete surface with lining seams and cracks, where the cracks are marked with dashed boxes.

Lining technology is usually applied in tunnel engineering. It refers to a permanent support structure constructed with reinforced concrete and other materials along the periphery of the tunnel body to prevent deformation or collapse of the surrounding rock. Integral mold concrete lining is one of the forms of tunnel lining. In order to handle the expansion and contraction to the tunnel lining structure caused by temperature and humidity, as well as the damage caused by uneven settlement of the foundation or seismic waves, the tunnel structure is divided into several parts in advance and each part can deal with changes independently. These reserved seams that separate the parts of the structure includes anti-deformation seam, anti-settlement seam and anti-seismism seam. In addition, construction cold seams are easily formed by layered and segmented pouring of concrete lining, which mainly include the longitudinal seam and the circumferential seam. All of the above types of seams are collectively called lining seams.

Thus it is very necessary to study how to remove the interference of lining seams in crack detection. Qu *et al.* [15] proposed a lining seam removal algorithm based on unit line feature extraction, which to some extent fills the vacancy in this special studies. The algorithm used the improved progressive probabilistic Hough transform to detect significant linear characteristics, extracting the minimum line feature of the lining seams called Unit-Line, and removed lining seams according to the characteristics of the Unit-Line. However, in some images where cracks similar to linear segments could be removed as the lining seams by mistake.

In this paper, we proposed a novel method for eliminating lining seams in tunnel concrete to help detect actual cracks in complex backgrounds, which enriched studies after initial crack extraction. The module of what our work is addressed is highlighted in the blue dashed box in Fig. 1. On the basis of the second accelerated percolation detection, firstly, salient line segments were extracted by the Line Segment Detector (LSD). Secondly, according to the length feature and oblique angle, every line segment was classified by the proposed uniform k-divided angle model, and its translation direction and expansion width were determined by the proposed rules. Finally, every line segment was translated with a distance of the expansion width and pure cracks can be obtained after percolation de-noising.

The remainder of this paper is organized as follows. In Sec. 2, we briefly reviewed the second accelerated percolation algorithm and the de-noising method related to our research. In Sec.3, our proposed lining seams elimination method was presented in forms of each detailed process, viz., the extraction of salient line segments, the classification of line segments, and the translation and expansion of line segments. In Sec.4, the experimental evaluations and quantitative analysis of the proposed method were reported. Finally, we concluded this paper in Sec. 5.

II. RELATED WORK

In the process of crack detection, as shown in Fig.1, targets in the crack image were first extracted by preliminary crack detection algorithms, then lining seams would be detected and removed by subsequent steps. Therefore, here related work refers to the crack extraction and the noises removing, i.e., the second accelerated percolation algorithm and the de-noising algorithm used in the related processing. The two related algorithms were briefly described as follows.

A. THE SECOND ACCELERATED PERCOLATION ALGORITHM

Percolation is a physical model based on the natural phenomenon of liquid permeation. In image processing, the specific gray value of a certain pixel is used as the critical probability in percolation model. The percolation algorithms [9], [10], [15] fully considered the continuity of the gray values of pixels in the neighborhood. According to the cluster characteristics, whether the center pixel in the local region belongs to the crack can be determined in the process of the repeated calculation. The percolation process of the single pixel is briefly described as follows.

i. Update the percolation threshold T and the acceleration parameter w as follows:

$$w^{'} = F_c \times w \tag{1}$$

$$T = \max\left\{\max_{p\in D_p}\left[I(p)\right], T\right\} + w^{'} \tag{2}$$

where F_c is the circularity characteristic of the percolated cluster, $I(p)$ is the gray value of pixel p and D_p is the percolation cluster.

ii. Form a candidate region D_c by adding the eight neighbor pixels of each pixel in D_p. If there is a pixel with the gray value $I(p_c) < T$, then add the pixel p_c into D_p. Otherwise, add the pixel with the lowest $I(p_c)$ into D_p.

iii. F_c is calculated according to the following equation:

$$F_C = \frac{4 \times C_{count}}{\pi \times C_{\max}^2} \tag{3}$$

where C_{count} is the number of pixels in D_p, and $C_{\max}$ is the diameter of the minimum circumscribed circle for D_p.

The flowchart of the second accelerated percolation is briefly provided as follows:

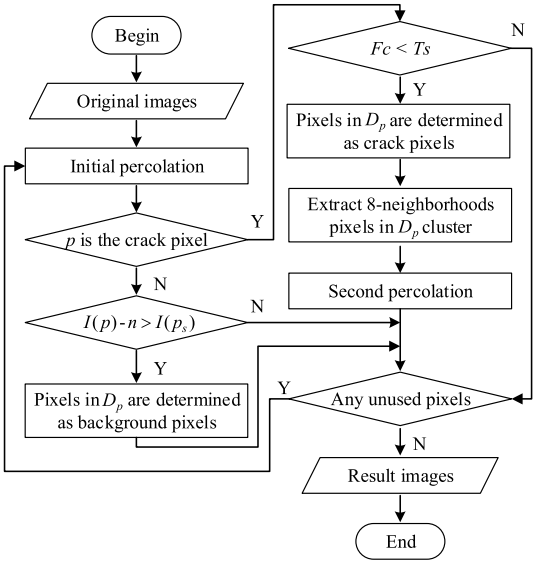


FIGURE 3. Flowchart of the second accelerated percolation.

The second accelerated percolation [9] is based on the single pixel percolation. If the center pixel p_s is determined as a background pixel, and every $p \in D_p$ satisfies $I(p) - n > I(p_s)$, then all the pixels in D_p are directly determined as background pixels, and no secondary percolation is required. The constant $n \geq 0$ is set mainly for avoiding fracture of small cracks; If p_s is determined as a crack pixel, and the circularity characteristic F_c satisfies $F_c < T_s$, then all the pixels in D_p are directly regarded as crack pixels. Then, a second percolation is performed for the 8 neighborhoods of the crack pixels to avoid missed detection, where T_s is the threshold for determining whether the pixel is the crack pixel. The result of the second accelerated percolation is shown as Fig. 4.

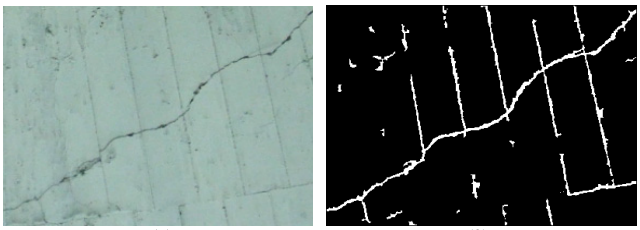


FIGURE 4. Detection result. (a) An original tunnel concrete lining image. (b) Detection of the second accelerated percolation.

B. THE DE-NOISING ALGORITHM

Because the characteristics of the blurred cracks are similar to the background noise, gray features of the cracks have been considered as shape features as well. On the basis of the percolation model, the percolation de-noising [10] can be performed when $I(p) < F_cMax \times 255$, where F_cMax is the upper limit for F_c. The steps are briefly described as follows:

i. Add the center pixel p_s, which satisfies $I(p_s) < F_cMax \times 255$, to the percolation cluster D_p.

ii. Form a candidate region D_c by adding the eight neighbor pixels of each pixel in D_p. For each pixel $p_c \in D_c$, add p_c into D_p if $I(p_c) < F_cMax \times 255$.

iii. Calculate the radius r of the minimum circumscribed circle of D_p, and calculate F_c according to Eq. (3). If F_c and r satisfy Eq. (4), D_p will be regarded as a noise region, and the pixels in D_p will be determined as the background pixels. Otherwise, they will be determined as the crack pixels,

$$F_c > F_cMax || r < rMax \tag{4}$$

where $rMax$ is the upper limit for r.

III. LINING SEAMS ELIMINATION

In this section, lining seams elimination are divided into four parts, and we described each part of our proposed method in detail.

A. THE FRAMEWORK OF THE PROPOSED METHOD

The steps of the proposed method are as follows:

i. Input the pre-extraction image detected by the second accelerated percolation algorithm, which is shown as Fig. 4(b).

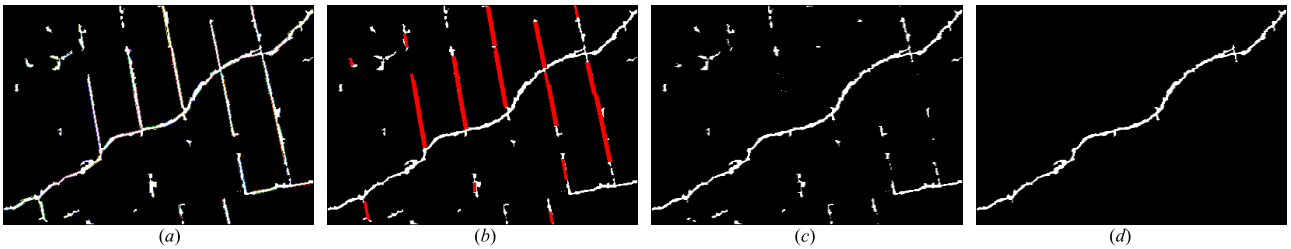


FIGURE 5. Samples of every step. (a) Line segments extraction. (b) Marking the information of the translation and expansion. (c) Result after line segments' translation and expansion. (d) Final result after de-noising.

TABLE 1. Principle of the translation direction.

Oblique Angle	Pixels to Determine	Translation Direction
$\theta_l \in \left[0, \frac{\pi}{4}\right] \cup \left(\frac{3\pi}{4}, \pi\right)$	$P(x, y \pm offset)$	*Vertical*
$\theta_l \in \left(\frac{\pi}{4}, \frac{3\pi}{4}\right]$	$P(x \pm offset, y)$	*Horizontal*

TABLE 2. Principle of the expansion width.

Relationship		Expansion Width
$0 < length_l \le len_1$		0
$len_1 < length_l < len_2$	$N_{\theta_l} < N_{Thr}$	0
	$N_{Thr} \le N_{\theta_l}$	$Width_1$
$len_2 \le length_l$		$Width_2$

ii. Run the Line Segment Detector (LSD) to extract salient line segments. The extraction is shown as Fig. 5(a).

iii. Calculate the length and the oblique angle of all line segments, and classify these line segments according to the proposed uniform k-divided angle model in polar coordinates, and set the accumulator to count the number of line segments in each angle subset.

iv. Traverse the line segment set, determine the translation direction and the expansion width of current line segment according to the two proposed principles shown as Table 1 and Table 2, respectively, by referring the length and the oblique angle calculated in the previous step. By marking these information with red color, the image is shown as Fig. 5(b).

v. Translate and expand all line segments according to their translation direction and expansion width. The result is shown as Fig. 5(c).

vi. De-noise the image and output the final result. The final image is shown as Fig. 5(d).

In this paper, the flowchart of the proposed method is shown as Fig. 6.

B. THE EXTRACTION OF SALIENT LINE SEGMENTS

In this paper, we used the pre-extraction images. i.e., the detections after running the second accelerated percolation algorithm, as the input images. During our processing, the influence caused by the external environment on

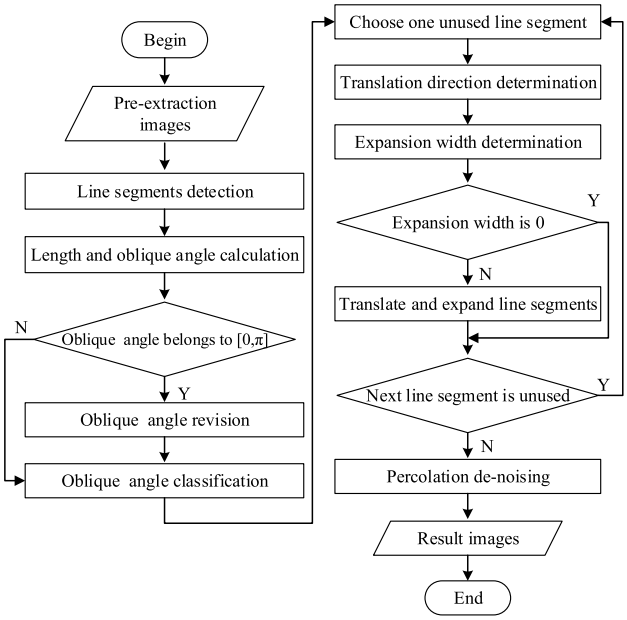


FIGURE 6. Flowchart of the proposed method.

the images, such as illumination, shadow, etc., had been dealt with in the process of the second accelerated percolation, i.e., some necessary pre-processing had been done. As these obtained pre-extractions contained many interconnected cracks and lining seams, and most lining seams have obvious linear characteristics, therefore, utilizing line detection methods [16]–[21] to remove lining seams should be able to achieve better results for getting pure cracks. Here we used the Line Segment Detector (LSD) [22], [23] to extract salient line segments, as the first step.

Because LSD works well for lots of images without parameter tuning and it controls the number of false detections. In addition, LSD is much faster than other classic line segment detection algorithms, and its detections are more accurate. Here, the steps of the LSD algorithm are briefly described as follows:

i. Image scaling: the image is filtered with a Gaussian kernel and then sub-sampled.

ii. Gradient computing: for every pixel (x, y) its level-line angle is computed as follows:

$$\theta_{(x,y)} = \arctan\left(\frac{g_x(x, y)}{g_y(x, y)}\right) \tag{5}$$

and its gradient magnitude is computed as follows:

$$G_{(x,y)} = \sqrt{g_x^2(x,y) + g_y^2(x,y)} \tag{6}$$

where

$$g_x(x,y) = \frac{I(x+1,y) + I(x+1,y+1) - I(x,y) - I(x,y+1)}{2} \tag{7}$$

$$g_y(x,y) = \frac{I(x,y+1) + I(x+1,y+1) - I(x,y) - I(x+1,y)}{2} \tag{8}$$

and $I(x, y)$ is the gray value of pixel (x, y).

iii. Gradient pseudo-ordering: 1024 bins are created corresponding to equal gradient magnitude intervals between zero and the largest observed value on the image. Pixels are classified into the bins according to their gradient magnitude.

iv. Region growing: a line-support region is formed when pixels are added to the region whose level-line angle $\theta_{(x,y)}$ is equal to the region angle θ_{region} up to a tolerance τ, here the region angle θ_{region} is computed as follows:

$$\theta_{region} = \arctan\left(\frac{\sum \sin\left(\theta_{(x,y)}\right)}{\sum \cos\left(\theta_{(x,y)}\right)}\right) \tag{9}$$

Starting from the bin with the largest gradient magnitudes, pixels in each bin are used in a descending order, but pixels with gradient magnitude smaller than ρ are rejected. Recursively, the neighbors of the pixels already in the region are tested. The process is repeated until no other pixel can be added to the region.

v. Rectangular approximation and validation: Every line segment can be regarded as a geometrical region, i.e., a rectangle. The width and the length of the rectangle r are set to the smallest values that make the rectangle r to cover the full line-support region. The *p aligned* point means the pixel in a rectangle whose level-line angle is equal to the rectangle's main angle up to a tolerance $p\pi$, where p is the precision. The density d of the aligned points of the rectangle r is computed as follows:

$$d = \frac{k}{length(r) \cdot width(r)} \tag{10}$$

where k is the number of the p - *aligned* points, and the rectangle r can be accepted when it has a density d not smaller than the threshold D. Here D is set to a specific value which provides good balance between solving the obtuse angle problem, and providing smooth approximations to curves, without over-cutting the line segments. If r is not accepted, it will be cut into two smaller regions at the right place until the condition is satisfied.

vi. *NFA* computing: the number of false alarms (*NFA*) associated with the rectangle r is computed as follows:

$$NFA(r) = (NM)^{5/2} \gamma \cdot B(n,k,p) \tag{11}$$

where N and M are the number of columns and rows of the image (after scaling), p is the precision, γ is the number of different values for p are tried, and $B(n, k, p)$ is the binomial which is computed as follows:

$$B(n,k,p) = \sum_{j=k}^{n} \binom{n}{j} p^j (1-p)^{n-j} \tag{12}$$

and the rectangle r with $NFA(r) \leq 1$ is validated as a line segment.

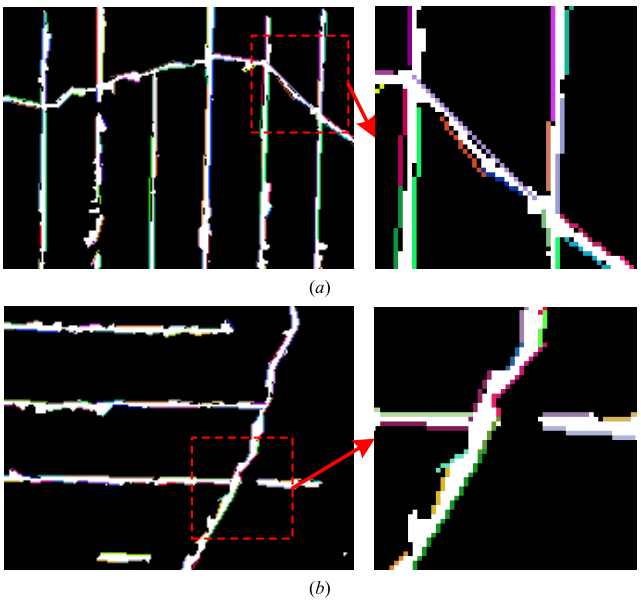


FIGURE 7. Results of salient line segments obtained by the LSD. (a), (b) Examples of salient line segments obtained by the LSD algorithm.

As shown in Fig. 7, there are two examples of salient line segments obtained by the LSD algorithm (line segments are marked with colored lines), and each one of them have a local part magnified in the red dashed boxes on the right side. By observing a large number of images extracted line segments by the LSD, we found a regular phenomenon, as the LSD targets at line segments at edges, the length of the line segment extracted from cracks edges are shorter, compared with length of the line segment extracted from seams edges. Furthermore, seeing with naked eye, line segments from crack edges are oriented in various and irregular directions, however, line segments from seams edges have certain cluster characteristics in the direction distribution, i.e., they are oriented in one or several main directions. Therefore, our lining seams elimination algorithm greatly utilizes these two features.

C. THE CLASSIFICATION OF THE LINE SEGMENTS

After running the LSD algorithm, the line segment set of the image has been obtained. For every line segment l in the set, its length is calculated as $length_l$, and its oblique angle is calculated as θ_l, respectively. We revise θ_l to $(\theta_l + \pi)$ if $\theta_l < 0$, thus all oblique angles can belong to the interval $[0, \pi)$.

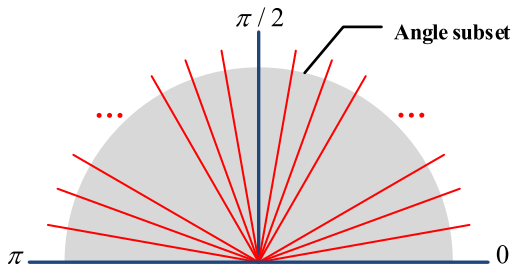


FIGURE 8. The uniform *k*-divided angle model.

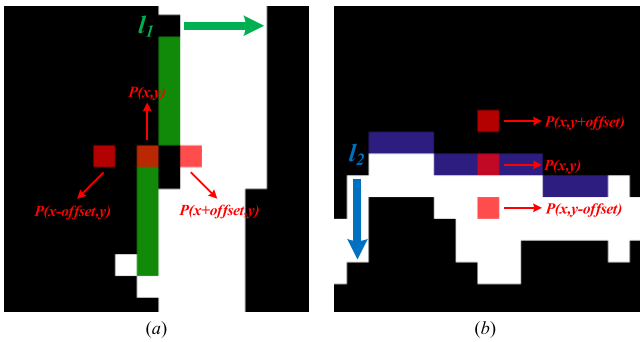


FIGURE 9. Diagram of the determination of translation direction. (a) Translation in horizontal. (b) Translation in vertical.

After that, according to the uniform k-divided angle model we proposed, which is based on the polar coordinate, as shown in Fig. 8.

In Fig. 8, every line segment is classified into its own sub-region, and the accumulator corresponding to each sub-region is set to count the number of line segments in it, we used N_{θ_l} to represent this quantity. Here, the classification is based on the oblique angle θ_l, as we can see in the uniform k-divided angle model, k subsets are created corresponding to equal angle intervals between 0 and π, where k is the number of subsets and here we set $k = 60$, thus line segments classified into the same subset can have the same oblique angle up to an accepted tolerance τ, where $\tau = \pi/60$.

D. TRANSLATION AND EXPANSION OF LINE SEGMENTS

Once the classification of line segments is finished, the next step is line segment's translation and expansion. The core of this step is to determine the direction of translation and the width of expansion. Here, the direction of translation and the width of expansion are determined by two principles we proposed, as shown in Table 1 and Table 2.

1) THE TRANSLATION DIRECTION OF LINE SEGMENTS

We give two diagrams of how we decide the translation direction in Fig. 9, here the two line segments are l_1 and l_2, and we take l_1 as an example to explain in detail. First of all, we calculate the coordinates of the midpoint $p\,(x, y)$ of l_1 according to the coordinates of the two endpoints, i.e., $p_1\,(x_1, y_1)$ and $p_2\,(x_2, y_2)$, the calculation is shown as follows:

$$\begin{cases} x = (x_1 + x_2)/2 \\ y = (y_1 + y_2)/2 \end{cases} \tag{13}$$

Next, we need to determine the brightness of two pixels which are at a fixed distance of the midpoint $p\,(x, y)$, when $\theta_l \in \left(\frac{\pi}{4}, \frac{3\pi}{4}\right]$ as θ_{l_1}, the two pixels are $p\,(x + offset, y)$ and $p\,(x - offset, y)$, respectively, when $\theta_l \in \left[0, \frac{\pi}{4}\right] \cup \left(\frac{3\pi}{4}, \pi\right)$ as θ_{l_2}, the two pixels are $p\,(x, y + offset)$ and $p\,(x, y - offset)$, respectively, where *offset* is the minimum distance threshold.

Assume $\theta_l \in \left(\frac{\pi}{4}, \frac{3\pi}{4}\right]$ as θ_{l_1}, if the brightness of pixel $p\,(x + offset, y)$ satisfies $I\,(p\,(x + offset, y)) = 255$, then the line segment l moves horizontally to the side where the pixel $p\,(x + offset, y)$ is located. Similarly, l moves horizontally to the side where the pixel $p\,(x - offset, y)$ is located if $I\,(p\,(x - offset, y)) = 255$.

Because LSD detects locally straight edges, i.e., it targets at edge of the white pixels in the binary image, every line segment always translates and expands to only one side of the vertical direction or the horizontal direction. The reason why we do not determine the brightness of the midpoint $p\,(x, y)$ but the two pixels which are at a fixed distance of the midpoint $p\,(x, y)$ is that if a line segment is not strictly horizontal or vertical, it is segmented like l_1 when magnified thus not all pixels on the line segment are white target pixels, as we can see the midpoint $p\,(x, y)$ of l_1 is a black pixel. Therefore, it may be not accurate to verify the brightness of $p\,(x, y)$. And we set $offset = 2$ in this paper, meaning a distance of two pixels, based on the fact that almost all lining seams have a wider width than two pixels.

Here, as shown in Fig. 9(a), l_1 is the green line segment with $\theta_{l_1} \in \left(\frac{\pi}{4}, \frac{3\pi}{4}\right]$, as its orientation is closer to the vertical direction inspected with the naked eye. We mark $p\,(x, y)$, $p\,(x + offset, y)$ and $p\,(x - offset, y)$ with red color, and we can clearly see that $p\,(x + offset, y)$ is a white target pixel and $p\,(x - offset, y)$ is a black background pixel, so l_1 moves horizontally along the side of $p\,(x + offset, y)$ or the direction of the green arrow. To determine the direction of translation for l_2 is similarly (shown as Fig. 9(b)), l_2 moves vertically along the side of the blue arrow.

2) THE EXPANSION WIDTH OF LINE SEGMENTS

As same as previous part, we also give an example of how we decide the expansion width of the line segments. In section 3.2, we have calculated the length of every line segment and obtained the number of line segments in every angle subset. Here we mainly use these two properties according to the principle shown in Table 2.

In Table 2, $length_l$ and N_{θ_l} are the length of the line segment l and the number of the line segments in the angle subset to which l belongs, respectively.

len_1 and len_2 are two predefined length thresholds of the current line segment. len_1 is the maximal suppression length

threshold for a line segment which has a great probability to form the edge of a crack, and len_2 is the minimal suppression length threshold for a line segment which is most likely to be on the edge of a lining seam. len_1 is used to directly judge if line segments are on the edges of the cracks or the lining seams. If len_1 is set too short, some edges of crack will easily be identified as the edges of lining seams, thereby it is possible to remove short line segments on the cracks by mistake; if len_1 is set too long, some edges on the unsmooth lining seams will be recognized as the crack edges, causing lining seams are retained rather than removed. len_2 is used to judge line segments located on lining seams are the smooth edges or the unsmooth edges. If len_2 is set too short, the number of classified line segments applied in the uniform k-divided angle model will be greatly reduced, thus the determination of the edges on the lining seams will be inaccurate; if len_2 is set too long, the number of classified line segments applied in the uniform k-divided angle model will be sharply increased, thus time cost of our proposed algorithm will increase a lot;

N_{Thr} is the maximal suppression threshold, which is the limit number of line segments that are classified into the same angle subset and form the edge of the cracks. If N_{Thr} is set too small, an amount of not very curved cracks may be removed as lining seams; If N_{Thr} is set too large, a number of not very smooth lining seams being wrongly preserved as cracks.

$Width_1$ is a predefined smaller expansion width for line segments located on unsmooth edges, and $Width_2$ is a predefined larger expansion width for line segments located on smooth edges. For $Width_1$, if it is set too small, line segments' translation and extension process is likely to leave short and thin areas, thus unsmooth lining seam cannot be broken; if $Width_1$ is set too large, it may cause excessive removal and will increase time consumption at the same time; For $Width_2$, if it is set too small, long and thin areas are probable to be left, and then lining seam with long length cannot be broken; if $Width_2$ is set too large, on the one hand, it may cause excessive removal and increase execution time, on the other hand, such kind of removal cannot completely preserve cracks and may even add a lot of non-crack areas, therefore it makes no sense.

Here we define $0 < len_1 < len_2$, $N_{\theta_l} > 0$, $Width_1 > 0$, $Width_2 > 0$. In this paper, we set the above thresholds to the following empirical values:

$$\begin{aligned} 6 \le len_1 \le 9\,[pixel]\,, \quad len_2 = 35\,[pixel]\,, \\ N_{Thr} = 10, \\ Width_1 = 3\,[pixel]\,, Width_2 = 5\,[pixel]\,. \end{aligned}$$

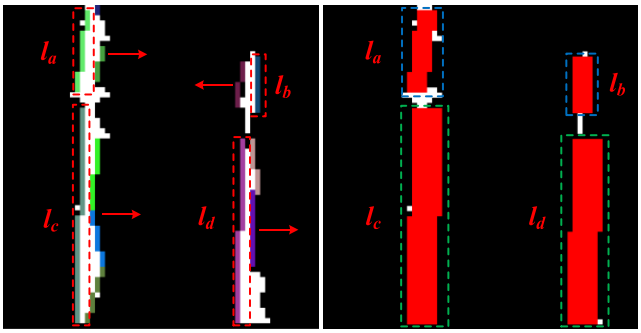


FIGURE 10. Diagram of the determination of expansion width. (a) Expanding preparations. (b) Result of different width of expansions.

If $0 < length_l \le len_1$, l has a great probability to form the edge of a crack, so it will not be expanded and the expansion width for l is just zero; If $len_1 < length_l < len_2$ and $N_{\theta_l} < N_{Thr}$, the length of l is not very large and there are few line segments which have the same oblique angle with l, or it is possible that l is the edge of not very curved cracks, so l will not be expanded, either; If $len_1 < length_l < len_2$ and $N_{Thr} \le N_{\theta_l}$, it means, although the length of l is not very large, there are some line segments which have the same oblique angle with l, or it is possible that l is the edge of a lining seam which is not smooth, so l will be expanded for a width of $Width_1$; If $len_2 \le length_l$, it means l is most likely to be located on the edge of a lining seam, so l will be expanded for a width of $Width_2$.

As shown in Fig. 10(a), there are four line segments in the red dashed boxes, i.e., l_a, l_b, l_c and l_d, and they are in the state before being translated and expanded. Then in Fig. 10(b), we can see that after translation and expansion operations, l_a and l_b were expanded for a distance of $Width_1$ in the blue dashed boxes, and l_c and l_d were expanded for a distance of $Width_2$ in the green dashed boxes, and all results are marked with red color to been clearly seen.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. THE EXPERIMENTAL EVALUATION INDEXES

In this paper, the classification accuracy evaluation method based on confusion matrix [24], [25] is used to conduct the quantitative analysis of the experiment. Table 3 shows the terms of the confusion matrix.

Precision (P), recall (R), F-measure (F), accuracy (Acc), false positives rate (FPR), noise rate (NR) are the comprehensive evaluation indexes [26], [27] which are defined

TABLE 3. The confusion matrix.

Real Image	Detection Result		
	Positive	**Negative**	**Total**
Positive	TP=True Positives	FN = False Negatives	TP+FN
Negative	FP=False Positives	TN = True Negatives	FP+TN
Total	TP+FP	FN+TN	—

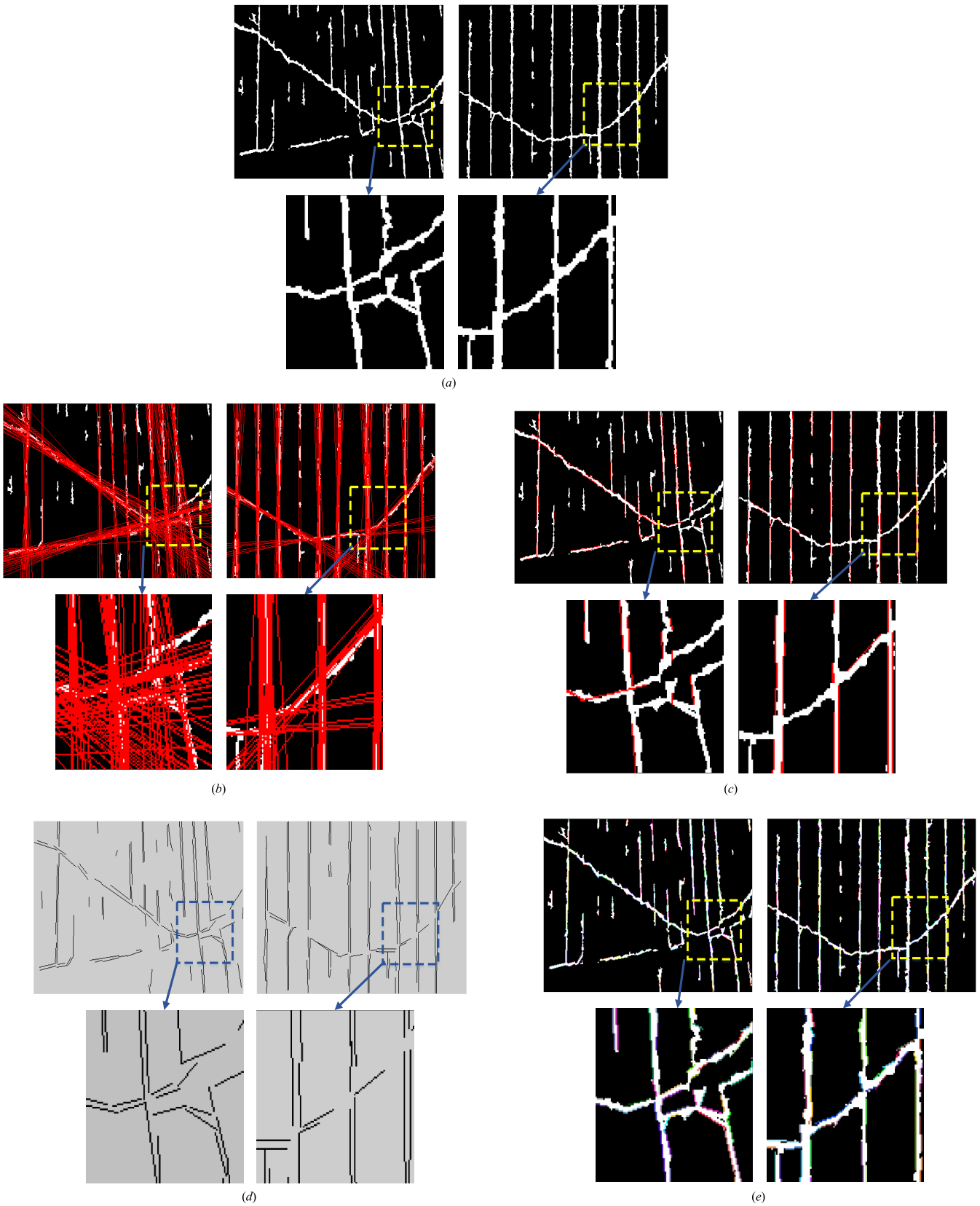


FIGURE 11. Comparisons of different line detection methods. (a) Detections of the second accelerated percolation. (b) HT line detections. (c) PPHT line detections. (d) EDLines line detections. (e) LSD line detections.

TABLE 4. Execution time and quantity of average in each group images.

Group	HT		PPHT		EDLines		LS-T&E					Unit-Line
							LSD		Translation and Expansion	De-noising	Total	
	Time	Num	Time	Num	Time	Num	Time	Num				
1st	3.88ms	10.4	3.38ms	19.2	25.1ms	68.5	27.3ms	197.4	721ms	32.7ms	781ms	847ms
2nd	3.39ms	9.9	2.97ms	18.8	20.3ms	55.7	29.6ms	183.0	726ms	32.4ms	788ms	857ms
3th	3.30ms	14.1	3.28ms	26.8	20.6ms	65.1	31.2ms	201.2	721ms	29.8ms	782ms	868ms
4th	3.02ms	19.5	3.59ms	34.2	21.0ms	80.9	32.1ms	244.4	714ms	30.9ms	777ms	891ms
5th	3.90ms	14.2	4.00ms	24.1	27.7ms	95.9	25.9ms	232.9	712ms	33.1ms	771ms	834ms
6th	3.19ms	12.2	3.20ms	23.6	19.4ms	64.0	30.4ms	174.2	722ms	30.6ms	783ms	850ms
7th	3.40ms	13.8	3.39ms	25.0	22.1ms	71.4	30.7ms	208.8	718ms	29.3ms	778ms	863ms
8th	3.89ms	13.7	3.90ms	23.5	27.3ms	91.2	25.7ms	228.3	711ms	31.3ms	768ms	838ms

as follows:

$$P = \frac{TP}{TP + FP} \tag{14}$$

$$R = \frac{TP}{TP + FN} \tag{15}$$

$$Acc = \frac{TP + TN}{TP + FN + FP + TN} \tag{16}$$

$$NR = \frac{FP}{TP + FP} \tag{17}$$

$$F = \frac{(\alpha^2 + 1) \times P \times R}{\alpha^2 \times (P + R)} \tag{18}$$

$$FPR = \frac{FP}{FP + TN} \tag{19}$$

where TP is the number of correctly classified crack pixels in the detection result, i.e., the number of pixels determined as the actual crack in the detection result. TN is the number of correctly classified background pixels in the detection result, i.e., the number of pixels determined as the actual background in the detection result. FP is the number of background pixels that are detected as crack pixels in the detection result. FN is the number of crack pixels that are detected as background pixels in the detection result. $TP + FP$ is the number of pixels detected as cracks in the detection result. $TP + FN$ and $FP + TN$ are the number of crack pixels and the number of background pixels in the actual segmentation result.

In crack detection, precision indicates the percentage of correct target in detection result, recall indicates the percentage of target in detection result that can be recalled to the actual target class, and noise rate indicates the percentage of false target in the detection result. A lower precision means more false crack pixels in the detection result, a lower recall means more missing crack pixels in the detection result. Owing to the interaction between P and R, the global performance of an algorithm cannot be fully reflected. Consequently, according to Eq. (18), we integrate the two measurements into a comprehensive evaluation index, viz., the weighted harmonic mean named $F - measure$ [28]–[30]. F_1 can comprehensively reflect the performance of an algorithm when $\alpha = 1$, as shown in Eq. (20).

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{20}$$

B. THE EXPERIMENTAL ANALYSIS

The experimental image set was provided by the engineering project. There were thousands of images taken under the natural environment using a regular Nikon camera with back-illuminated complementary metal-oxide semiconductor (CMOS) and 14 times optical zoom by professional engineers. Among them, about 500 tunnel concrete lining surface images were used for laboratory research. We divided these images into 8 groups based on the following considerations: similar complications or issues, directions of lining seams, and complexity of cracks. The number of images in each group ranges from 50 to 80, and the size of images used for the experiments is 400 × 300 [pixel].

In order to find out the most suitable line detection algorithm in crack images, we performed line detections on the results of the second accelerated percolation using Hough transform [17] (HT), progressive probabilistic Hough transform [20] (PPHT), EDLines [21] and LSD, respectively, and four kinds of results were given in Fig. 11. Besides, we counted the number of detected line segments and the execution time of each methods, shown in the first few columns of the Table 4.

Fig.11 consists five subfigures, they were in order, the initial detections by the second accelerated percolation, HT line detections, PPHT line detections, EDLines line detections and LSD line detections. Especially, we enlarged one local position of each image, so that the details of different methods can be clearly seen.

In Fig.11, it can been seen that HT detected a lot of straight lines, as it has no limit on the length of lines, lines in the images were very long. Therefore, not only the adhesion between lines was severe, but also the detections cannot accurately mark the edges of lining seams. Compared with HT, PPHT added a variable that controls line segment

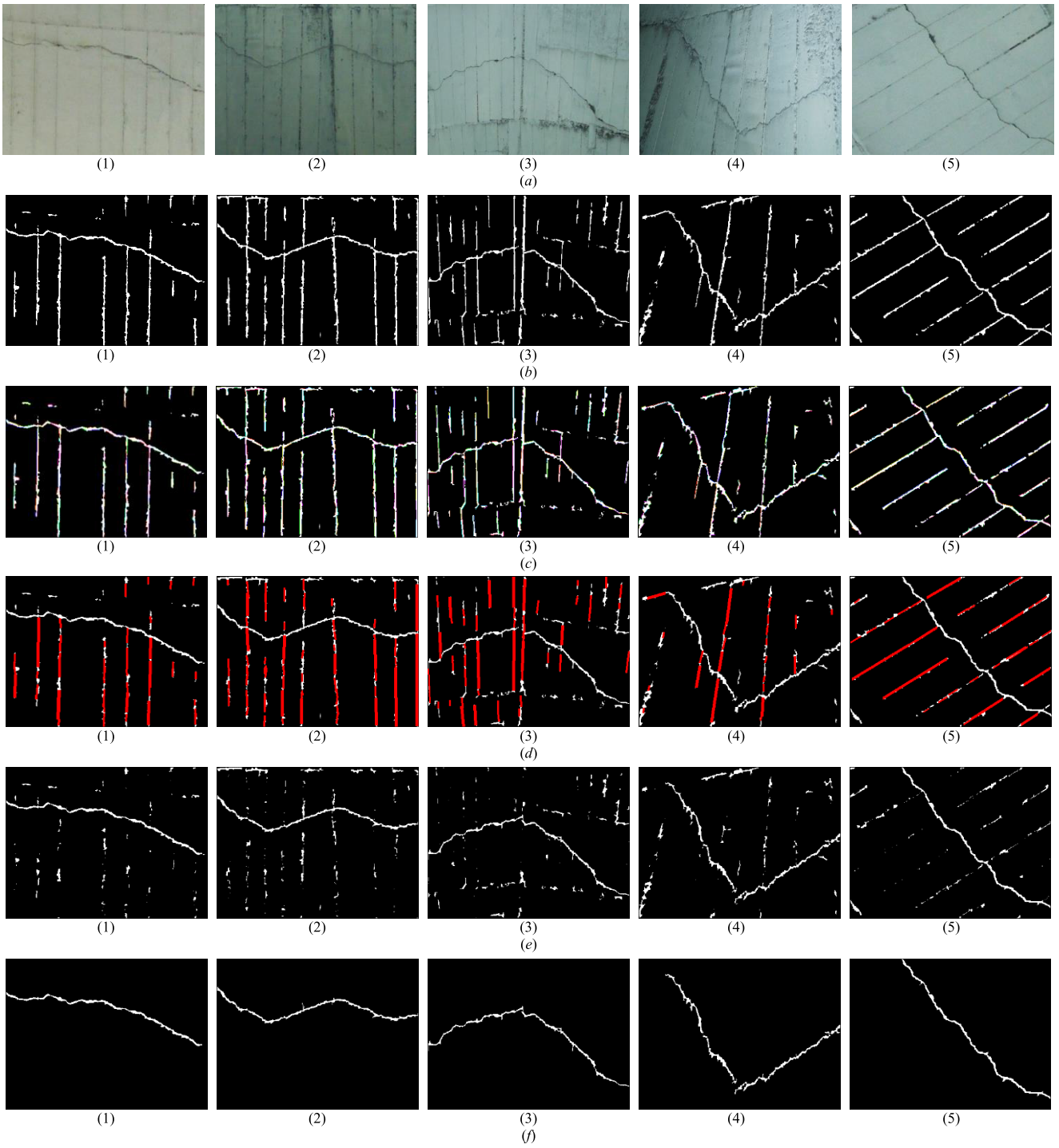


FIGURE 12. The results of different steps in LS-T&E. (a) Original tunnel concrete lining images. (b) Results of the second accelerated percolation. (c) Results of the LSD. (d) Marking information of the translation and expansion of line segments. (e) Results after line segments' translation and expansion. (f) Final results of the LS-T&E.

misdetection, so that the length and number of line segments were greatly improved, and rough edges were detected. The disadvantage is that for the edges with complex details, there were much missing. Detections of EDLines were more accurate than HT and PPHT, but compared with LSD, results of EDLines were rougher, especially on the locations where lining seams and cracks were connected, or we can say its details were not as rich as LSD. Obviously, LSD results were generally more accurate and detailed, even in complicated areas. These indicated that LSD can better meet the requirements of line segments' translation and expansion in our proposed method.

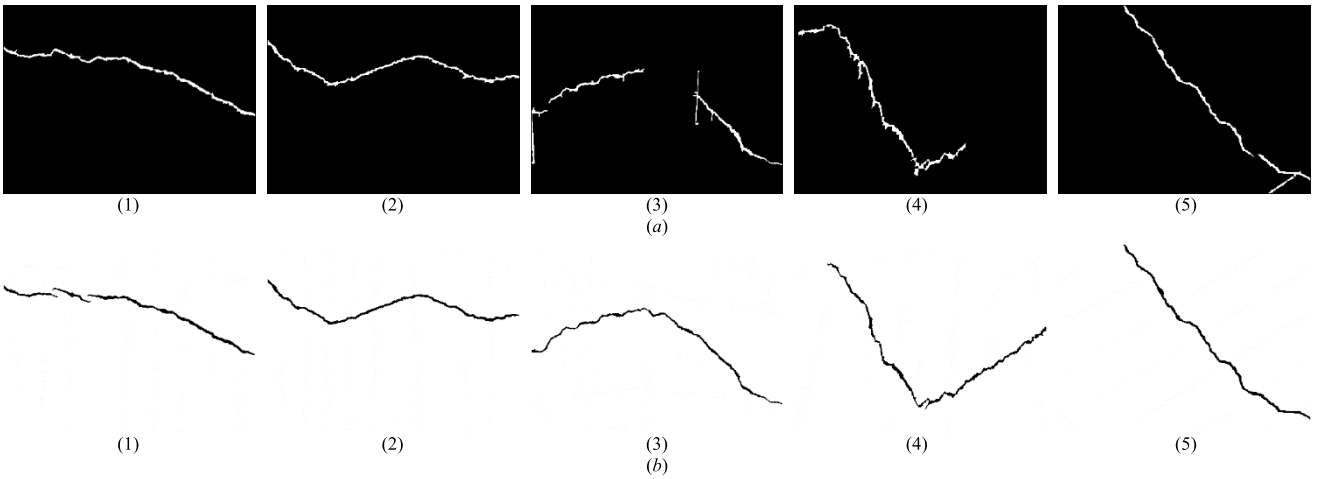


FIGURE 13. Comparisons of final results. (a) Final results of the Unit-Line [15]. (b) Ground-truth images.

In the first few columns of the Table 4, the average execution time and average number of detected line segments of HT, PPHT, EDLines and LSD in eight groups of crack images are shown. It can be seen that the average number of line segments detected by the four algorithms increased successively, which proved that the detailed information obtained in the results is more and more abundant. Among them, LSD had the maximum average number of line segments in each group of images, and it conformed to the conclusion drawn from Fig. 11.

As for the average execution time, since LSD generated line support areas by using the gradient of the pixels in the images and detected a large number of line segments, it took a little more time compared with HT, PPHT, and EDLines, however, the execution time of LSD was still very short, or we can say that it is satisfactory for line edge detection of the lining seams.

Therefore, we used LSD in the line segments extraction.

Next, to evaluate the elimination result and quantitative analysis of our presented method (line segments' translation and expansion, LS-T&E), Unit-Line method [15] was implemented as the comparison. These two methods were running under the same conditions (i.e. 2.5GHz Intel core i5-3210M CPU, Microsoft Visual Studio 2013, Opencv 3.0.0). Here we selected 5 images to show up in Fig. 12 and Fig. 13. Additionally, for LS-T&E, results of the method [9] were used as the input images, and images after line segments' translation and expansion were de-noised by method [10] in the last step.

Fig. 12 shows LS-T&E experimental images of different steps. They are, in order, the original tunnel concrete lining images, the results of the second accelerated percolation, the results of the LSD, images by marking the information of the translation and expansion of line segments, the results after line segments' translation and expansion, final results of the LS-T&E. Fig. 13 shows the comparison images. They are, in order, the final results of the Unit-Line [15] and the ground-truth images created manually by human experts, as it is believed that cracks should be identifiable in images via the naked human eyes.

From Fig. 12(f) and Fig. 13(a), we can see the final results between our proposed LS-T&E and the Unit-Line method [15] in these 5 selected images. For Fig. 12(a)(1) and Fig. 12(a)(2), both final results of the two methods were good. However, for Fig. 12(a)(3), Fig. 12(a)(4) and Fig. 12(a)(5), the results of the Unit-Line [15] had some noises and fractures whereas the results of the proposed LS-T&E had an approximately complete preservation of the cracks, although there were still a few small burrs or fractures in some certain locations.

As the lining seam is not a perfect geometrical region, i.e., a rectangle, some edges on lining seams were not smooth at all, these led to the broken short line segments extraction in LSD directly. When these small line segments' translations and expansions were operated, tiny burrs located among those edges cannot be influenced. Besides, due to the influences of external factors such as illumination, camera dithering and so on, initial image processing before LS-T&E may have a small amount of error detection, which may also affected the accuracy of LS-T&E as it processed behind them, that's why there were a few small burrs or fractures remained.

The last few columns of Table 4 show the computational efficiency between our proposed LS-T&E and the Unit-Line method [15] in total 8 groups of images. LS-T&E aims to make appropriate operations for each detected line segment, i.e., expanded or not expanded, so the execution time was mainly composed of three parts, namely, line segment extraction and classification, line segments' translation and expansion, and noises removing. LSD has an execution time proportional to the number of pixels in the images, i.e., line segments extraction worked in a linear time, after that length and oblique angle calculation and classification in the uniform k-divided angle model were followed, so the operation time in this part was also linear; Then, line segment's

TABLE 5. Performance analysis of average in each group images.

Group	LS-T&E						Unit-Line	
	P(%)	R(%)	F_1(%)	Acc(%)	FPR(%)	NR(%)	P(%)	R(%)
1st	85.95	96.62	90.95	99.75	0.19	14.05	81.52	78.87
2nd	88.77	94.35	91.41	99.74	0.17	11.23	83.68	82.41
3th	89.10	96.72	92.74	99.75	0.19	10.90	80.54	78.45
4th	85.78	96.70	90.89	99.75	0.19	14.22	81.75	78.34
5th	91.04	92.26	91.63	99.76	0.12	8.96	89.17	88.49
6th	90.27	93.54	91.83	99.75	0.15	9.73	84.62	84.83
7th	88.82	92.30	90.44	99.71	0.18	11.18	83.08	80.94
8th	86.48	96.63	91.27	99.73	0.21	13.52	79.49	79.49

translation and expansion involved determining the right translation direction and appropriate expansion width, and doing the defined width expansion, so time in this part was proportional to the number of line segments; Finally, percolation de-noising could be performed with operations proportional to the number of white pixels in the images, time in this part was also short.

All in all, compared with the Unit-line method in [15], the main workload of LS-T&E is the process of line segments' translation and expansion, while that for the Unit-Line method [15] is to judge the length information of unit-lines in an adjacent range. Besides, LS-T&E used the LSD to extract line characteristics, which is much faster than PPHT used in the Unit-Line method [15]. Therefore, although the translation and expansion of each line segment is time-consuming, the total time of LS-T&E is still less.

Table 5 show the average quantitative performance between our proposed LS-T&E and the Unit-Line method [15] in total 8 groups of images. From Table 5, we can see that the LS-T&E performed better, as the precisions were all over 85%, the recalls and the accuracies were high, the weighted harmonic mean F_1 were about 90%, the false positive rates were low, and the noise rates were under 15%, which confirmed the advantage of making an accurate angle classification of each line segment before eliminating process. Although the Unit-Line method [15] also had relatively high recalls and precisions, they fluctuate constantly in different images, i.e., they were not as stable as the LS-T&E.

V. CONCLUSIONS

In this paper, we proposed a novel method for eliminating lining seams in tunnel concrete crack images to achieve extracting pure cracks without interferences caused by complex backgrounds, to some extent it enriched the existing tunnel lining surface crack detection technologies. Unlike other methods, our proposed method was based on line segments' translation and expansion, and the complex linear characteristics on the tunnel lining surface were fully considered.

Based on results of the second accelerated percolation, significant line segment extraction was first obtained by using line detection methods. Then line segments were marked and classified according to different features, i.e., the length and the oblique angle. After that, line segments on the edges of lining seams were removed according to two introduced translation and expansion principles. Finally the pure cracks can be received after percolation de-noising. Only after the complete cracks in the images are obtained, classifying these cracks and calculating necessary parameters for tunnel maintenance can be done later.

All in all, our proposed method can perform accurate lining seams elimination, and has excellent time efficiency. By analyzing different characteristics of line segments on lining seams edges and crack edges, and using the proposed uniform k-divided angle model, our proposed method accurately distinguished and classified these two kinds of edges. And because of precisely the processing, the translation and extension operations were successfully to handle almost all tiny line segments without excessive removal or missing removal, even when they located at the joint of cracks and lining seams. In terms of time efficiency, our proposed method adopted the efficient line detection algorithm, and time of the translation and extension depends directly on the number of line segments, so the total time consumption is small. For our proposed method, the disadvantage is that after line segments' translation and extension, there are still a few small burrs or fractures in some certain locations, probably because the widths of the lining seam in these locations are larger than the average width of most lining seams, this is one of the special cases, we will further study to deal with it in the future.

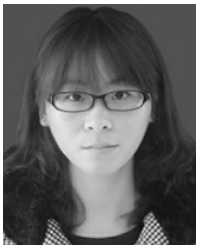
REFERENCES

[1] L. Zhang, G. Zhou, Y. Han, H. Lin, and Y. Wu, "Application of Internet of Things technology and convolutional neural network model in bridge crack detection," *IEEE Access*, vol. 6, pp. 39442–39451, 2018.
[2] P. Prasanna *et al.*, "Automated crack detection on concrete bridges," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 591–599, Apr. 2016.
[3] H. Wang, Z. Xiong, A. M. Finn, and Z. Chaudhry, "A context-driven approach to image-based crack detection," *Mach. Vis. Appl.*, vol. 27, no. 7, pp. 1103–1114, Oct. 2016.
[4] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart, "Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2718–2729, Oct. 2016.

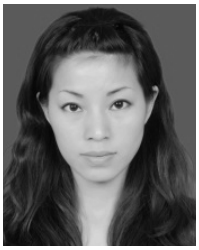
[5] Y. J. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 5, pp. 361–378, May 2017.
[6] T. Yamaguchi and S. Hashimoto, "Image processing based on percolation model," *IEICE Trans. Inf. Syst.*, vol. E89-D, no. 7, pp. 2044–2052, Jul. 2006.
[7] T. Yamaguchi, S. Nakamura, R. Saegusa, and S. Hashimoto, "Image-based crack detection for real concrete surfaces," *IEEJ Trans. Elect. Electron. Eng.*, vol. 3, no. 1, pp. 128–135, Jan. 2008.
[8] T. Yamaguchi and S. Hashimoto, "Fast crack detection method for large-size concrete surface images using percolation-based image processing," *Mach. Vis. Appl.*, vol. 21, no. 5, pp. 797–809, Aug. 2010.
[9] Z. Qu, Y. Guo, F.-R. Ju, L. Liu, and L.-D. Lin, "The algorithm of accelerated cracks detection and extracting skeleton by direction chain code in concrete surface image," *Imag. Sci. J.*, vol. 64, no. 3, pp. 119–130, Apr. 2016.
[10] Z. Qu, L.-D. Lin, Y. Guo, and N. Wang, "An improved algorithm for image crack detection based on percolation model," *IEEJ Trans. Elect. Electron. Eng.*, vol. 10, no. 2, pp. 214–221, Mar. 2015.
[11] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, "CrackTree: Automatic crack detection from pavement images," *Pattern Recognit. Lett.*, vol. 33, no. 3, pp. 227–238, Feb. 2012.
[12] Q. Zou, Z. Hu, L. Chen, Q. Wang, and Q. Li, "Geodesic-based pavement shadow removal revisited," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Shanghai, China, Mar. 2016, pp. 1761–1765.
[13] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, "Automatic road crack detection using random structured forests," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3434–3445, Dec. 2016.
[14] Y. Chen *et al.*, "Curve-like structure extraction using minimal path propagation with backtracking," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 988–1003, Feb. 2016.
[15] Z. Qu, L. Bai, S.-Q. An, F.-R. Ju, and L. Liu, "Lining seam elimination algorithm and surface crack detection in concrete tunnel lining," *Proc. SPIE*, vol. 25, no. 6, Nov. 2016, Art. no. 063004.
[16] P. Bao, L. Zhang, and X. Wu, "Canny edge detection enhancement by scale multiplication," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 9, pp. 1485–1490, Sep. 2005.
[17] H.-Y. Chen, Y.-Y. Lin, and B.-Y. Chen, "Co-segmentation guided Hough transform for robust feature matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 2388–2401, Dec. 2015.
[18] L. A. F. Fernandes and M. M. Oliveira, "Real-time line detection through an improved Hough transform voting scheme," *Pattern Recognit.*, vol. 41, no. 1, pp. 299–314, Jan. 2008.
[19] Z. Xu, B.-S. Shin, and R. Klette, "Accurate and robust line segment extraction using minimum entropy with hough transform," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 813–822, Mar. 2015.
[20] C. Galamhos, J. Matas, and J. Kittler, "Progressive probabilistic Hough transform for line detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Fort Collins, CO, USA, Jun. 1999, pp. 554–560.
[21] C. Akinlar and C. Topal, "EDLines: A real-time line segment detector with a false detection control," *Pattern Recognit. Lett.*, vol. 32, no. 13, pp. 1633–1642, Oct. 2011.
[22] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, Apr. 2010.
[23] Q. Wang, T. W. Chen, and L. Si, "Sparse representation with geometric configuration constraint for line segment matching," *Neurocomputing*, vol. 134, pp. 100–110, Jun. 2014.
[24] K. Lahiri and L. Yang, "Confidence bands for ROC curves with serially dependent data," *J. Bus. Econ. Statist.*, vol. 36, no. 1, pp. 115–130, 2018.
[25] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
[26] D. Ai, G. Jiang, L. S. Kei, and C. Li, "Automatic pixel-level pavement crack detection using information of multi-scale neighborhoods," *IEEE Access*, vol. 6, pp. 24452–24463, 2018.
[27] L. Sun, X. Jianchun, and Z. Xun, "An Algorithm for Concrete Crack Extraction and Identification Based on Machine Vision," *IEEE Access*, vol. 6, pp. 28993–29002, 2018.
[28] I. Giakoumis, N. Nikolaidis, and I. Pitas, "Digital image processing techniques for the detection and removal of cracks in digitized paintings," *IEEE Trans. Image Process.*, vol. 15, no. 1, pp. 178–188, Jan. 2006.
[29] H. I. Koo, "Text-line detection in camera-captured document images using the state estimation of connected components," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5358–5368, Nov. 2016.
[30] P. Giri and S. Kharkovsky, "Detection of surface crack in concrete using measurement technique with laser displacement sensor," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 8, pp. 1951–1953, Aug. 2016.

ZHONG QU received the M.S. degree in computer architecture and the Ph.D. degree in computer application technology from Chongqing University, in 2003 and 2009, respectively. He is currently a Professor with the Chongqing University of Posts and Telecommunications. His research interests are in the areas of digital image processing, digital media technology, and cloud computing.

SI-QI CHEN is currently pursuing the M.S. degree with the School of Software Engineering, Chongqing University of Posts and Telecommunications. Her main research interest is digital image processing.



YU-QIN LIU is currently pursuing the M.S. degree with the School of Software Engineering, Chongqing University of Posts and Telecommunications. Her main research interest is digital image processing.



LING LIU received the M.S. degree in computer technology from the Chongqing University of Posts and Telecommunications, in 2014. Her main research interest is digital image processing.