IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 21, 2018, accepted October 6, 2018, date of publication October 15, 2018, date of current version November 8, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2875889

Image-Based Crack Detection Using Crack Width Transform (CWT) Algorithm

HYUNWOO CHO[1,2], **HYUK-JIN YOON**[1,2], **AND JU-YEONG JUNG**[2]
[1]Department of Robotics and Virtual Engineering, Korea University of Science and Technology, Daejeon KS015, South Korea
[2]Korea Railroad Research Institute, Uiwang KS010, South Korea

Corresponding author: Hyuk-Jin Yoon (scipio@krri.re.kr)

This work was supported in part by the Smart Civil Infrastructure Research Program through the Ministry of Land, Infrastructure and Transport, Korean Government, under Grant 18SCIP-B065985-06, and in part by the Korea Agency for Infrastructure Technology Advancement.

ABSTRACT This paper describes an image-based methodology for the detection of structural cracks in concrete. The conventional approach based on line enhancement filtering has problems associated with an inaccurate extraction of edge pixels. We therefore propose an edge-based crack detection technique consisting of five steps: crack width transform, aspect ratio filtering, crack region search, hole filling, and relative thresholding. In the first step, crack width transform, opposing edges are identified and classified as crack candidate pixels, and a width map is generated. In the second step, aspect ratio filtering, the width map generated is used to remove noise. In the next two steps, the method searches for and restores missing pixels. In the last step, relative thresholding, residual noise is removed through reclassification of crack regions based on image-adaptive thresholding. The performance of this technique was tested using synthetic and real images. The results show that the proposed technique has greater accuracy and consistency in measuring crack width compared with the conventional technique.

INDEX TERMS Crack detection, crack width measurement, edge-based, crack width transform, crack region search.

I. INTRODUCTION

Structural cracks in concrete can occur for various reasons, including repetitive loading [1], chemical reactions [2], faulty construction, structural factors such as design mistakes, and environmental factors. Such structural cracks greatly affect the load transfer capacity and durability of a structure, posing a substantial risk to its safety. It is therefore crucial to ensure the stability of a structure by detecting defects, analyzing their causes, and providing appropriate remedies. In general, regular safety inspection and maintenance/repair of structures are required by law, given their importance and associated risks [3].

In order to ensure the safety of a structure, it is important to monitor its condition continuously. Currently, however, structural safety monitoring relies on regularly scheduled inspections performed at specified intervals, and on post-accident inspections [4]. In recent years, automatic image-based crack detection has attracted research interest as a promising technique to overcome the limitations of current structure safety inspection schemes. The following section briefly reviews the prior literature concerning crack detection based on image processing.

Reference [5] proposed a percolation-based technique to reduce computation time by calculating circularity during image processing and skipping the percolation processing in subsequent pixels. Reference [6] presented a method for identifying cracks on noisy concrete surfaces by minimizing noise through shade correction, Hessian-based line enhancement filtering, and probabilistic relaxation, and by detecting missing cracks on discontinuous cracks using region expansion through locally adaptive thresholding. Reference [7] proposed a depth perception method enabling contactless remote-sensing quantification of crack width, which was tested to have measurement errors of 23.34% and 5.50% at 0.4 mm and 1.2 mm, respectively.

Reference [8] proposed a method for predicting crack width from the difference in intensity of the crack scale and crack images. The prediction power was tested using real crack images captured under different illumination conditions. The average error was found to be 0.07 mm at the ground sampling distance (GSD) of 0.22 mm/pixel. However, this method has limited applications, given that attaching a crack scale on the test surface is impractical and the intensity distribution on cracks is not as uniform as on the crack scale. Reference [9] proposed techniques for semi-automatic crack width measurement. The crack width of the edge placed in the center of the spot with the lowest intensity in the region between the seed points determined manually. The measured value was then compared with that measured with the crack scale. In that method, by attaining a four- to five-fold increased accuracy relative to the resolution, a semi-automatic crack width measurement was achieved with acceptable results; however, it was still necessary to perform manual crack detection.

Reference [10] measured the width of cracks detected within a predetermined region using labeling based on RGB color information. However, crack detection was limited to specific regions, and automatic crack detection based on noise filtering could not be applied. Reference [2] proposed a system for measuring crack growth using time-series images. Screws were fixed to the specimen surface as reference points for crack identification and width measurement. For crack detection, seed points were determined and a tracing method was employed. Furthermore, the relationship between concrete expansion and cracking caused by the alkali–silica reaction was examined based on the results of crack width measurement using time-series images. The main limitation of this semi-automatic method is the need to select seed points manually.

Reference [11] presented a crack detection method using the Canny edge detection algorithm and the fast Haar transform algorithm, in which a Bayesian approach to edge detection selected a threshold value for crack detection. The main drawbacks are limited applicability of threshold value on rough and uneven surfaces, and false crack detection on non-cracked surfaces. Furthermore, the method could not extract crack pixels or measure crack width following crack edge detection. Reference [12] compared the crack detection performances of existing edge detectors. Their respective edge detection techniques were applied to images that were classified as cracked or non-cracked (25 each), whereby the threshold value was selected according to their average intensity. However, the comparison was achieved through relative analysis, and the crack detection efficiency of the edge detectors could not be compared based on pixel-level analysis.

Reference [13] presented a three-step algorithm for automated crack detection in sewer pipes: crack detection based on CCTV images; gap-filling based on operations to close the adjacent gaps; and directional filtering based on erosion operation, eccentricity, and area information. Directional filtering was also used for connecting discontinuous cracks, and it was claimed that this enabled the detection and measurement of various crack patterns.

Whereas many studies have presented a variety of techniques and algorithms for crack detection, comparatively few have dealt with crack width measurement, which is an important factor for safety diagnosis. In a review of papers on crack perception [14], it was pointed out that many related studies did not measure the crack width, also noted that crack width was measured only in nine of the 50 reviewed papers on image-based defect detection published between 2005 and 2015.

Against this background, we present a crack detection technique with enhanced accuracy in measuring the crack width, which is an important component of safety inspection. After analyzing the limitations of the currently available image processing methods, we present an edge-based crack recognition method designed to overcome these drawbacks. The proposed technique consists of five steps: crack candidate region extraction based on crack width transform, aspect ratio filtering, crack region search, hole filling, and relative thresholding. Its main advantage is more accurate crack pixel extraction compared with other techniques. The efficiency of the proposed technique was tested using synthetic and real imagesand pixel extraction accuracy was verified by comparison with a conventional technique.

II. GUIDELINES FOR MANUSCRIPT PREPARATION

Crack detection methods based solely on intensity are prone to false crack extraction resulting from various noise sources exhibiting similar intensities to that of a crack. Conversely, crack regions with below-average intensities are not properly extracted. These problems can be efficiently addressed by line enhancement filtering designed to enhance the crack region based on morphological as well as intensity information.

Owing to this advantage, line enhancement filtering has been used in many studies of crack detection. The most common approaches are Hessian-based filtering [6], [15]–[19] and morphological filtering [7], [20]. The typical Hessian-based Frangi filtering enhancement, which was designed for blood vessel detection [21], can be expressed by Equation (1).

$$V_o(x) = \begin{cases} 0 & \mathit{if}\ \lambda_2 < 0 \\ \mathit{exp}(-\frac{R_B^2}{2\beta^2})(1 - \exp\left(\frac{s^2}{2c^2}\right)) & \mathit{otherwise}, \end{cases}$$
$$s = \sqrt{\lambda_1^2 + \lambda_2^2}, \quad \beta = 0.5,\ R_B = \lambda_1/\lambda_2,\ \lambda_1 \ll \lambda_2 \quad (1)$$

where $V_o(x)$ represents the line enhancement filtering result, c is the standard deviation, and λ_1 and λ_2 are two eigenvalues of the Hessian matrix.

A line enhancement filter is highly efficient at removing noise sources other than linear patternsand is hence widely used for detecting linear structures such as cracks and blood vessels. However, it performs less well in measuring line widthand has not been dealt with in the related studies. This problem is analyzed in this section.

For problem analysis of line enhancement filtering, we generated a synthetic image as shown in FIGURE 1. The image displays three co-planar lines of differing intensity, such that the intensity-dependent differences in the line enhancement filtering value can be analyzed along with the consequent line width measurement errors. The lines on the synthetic image were configured to have 15-pixel width and 100-pixel length. The intensities of the three lines (#1 to #3)

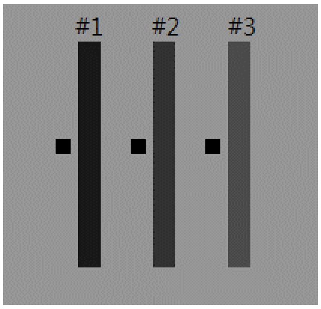


FIGURE 1. A synthetic image depicting three lines of differing intensity.

were configured to be 25, 50, and 75, respectively, against a background intensity of 150. Additionally, to analyze the effects of background noise, we placed a square-shaped noise source (10 × 10 pixels) of zero intensity at a 5-pixel distance from the edge.

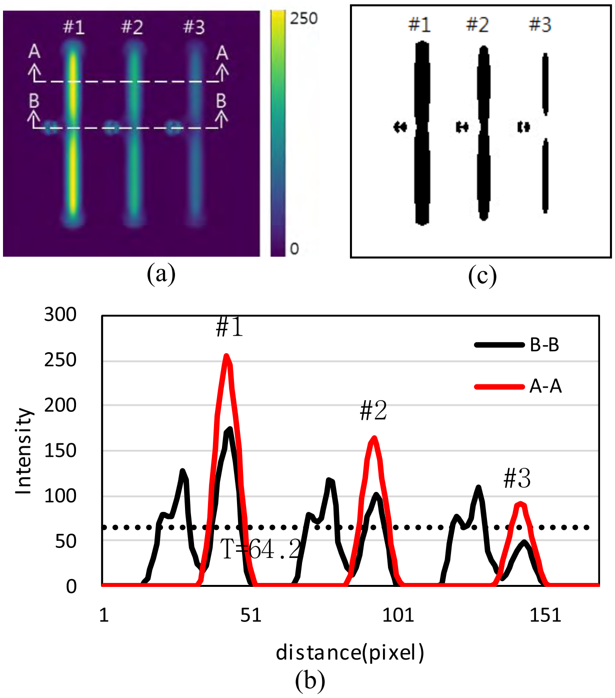


FIGURE 2. Results of line enhancement filtering of the synthetic image (FIGURE 1): (a) Line-enhanced image; (b) Intensity profile of (a); (c) Otsu thresholding image of (a).

FIGURE 2 shows the results of line enhancement filtering of the synthetic image (FIGURE 1) using a Frangi filter (Equation (1)). As a result of filtering, it was found that the filtering value varies according to the contrast between lines and background, with the value decreasing in regions with lower contrast. FIGURE 2(b) shows the intensity profile of FIGURE 2(a). The line segment containing no background noise (A–A) exhibited lower filtering values than the line segment containing background noise (B–B).

FIGURE 2(c) shows the result of line detection using thresholding the lines in FIGURE 2(a). The lines have different widths and each becomes narrower in the segments with background noise. Table 1 presents the width measurements obtained for the lines in FIGURE 2(c). The measured values reveal that line width varies within the same line depending on the line intensity and background noise.

TABLE 1. Line width (true width = 15 pixels) measured using thresholding (FIGURE 2(c)) after line enhancement filtering.

Line no.	A–A		B–B	
	W	Error (%)	W	Error (%)
#1	14	6.7	12	20.0
#2	12	20.0	7	53.3
#3	8	46.7	None	100.0

W: width (pixels)

These analysis results point to the problems associated with crack detection based on line enhancement filtering: it can yield different crack width measurements for cracks of the same width, depending on crack–background contrast and background noise. In other words, the currently used methods of line enhancement filtering have problems of accuracy and consistency in measuring crack width.

III. EDGE-BASED CRACK DETECTION TECHNIQUE

A. CRACK WITH TRANSFORM

To ensure consistent measurement of crack width, we propose the edge-based crack width transform technique (CWT). The Stroke Width Transform (SWT) [22] is a widely used edge-based object extraction technique designed for text recognition. SWT consists of three steps: searching opposing edges, allocating width between the opposing edges, and classifying the letters based on their width consistency.

In the CWT, crack assignment conditions are added to the opposing edge pixel searching process employed in the SWT [22]. FIGURE 3 illustrates the process of searching for an opposing edge pixel (q_j) of any given pixel (q_i) in the direction normal to the edge. Crack width (w) is the number of pixels located between the parallel opposing edges, as expressed by Equation (2).

$$w = card\left(q_{ij}\right) \tag{2}$$

Here, q_{ij} is the set of all pixels located between q_i and q_j, and $card\left(q_{ij}\right)$ is the number of q_{ij} (card = cardinality).

The opposing edge pixel search is iterated until the assigned width falls below the maximum threshold value. Given the limited crack width allowing measurement, setting the maximum threshold value has the effect of preventing the classification of a non-crack region as a crack candidate.

The opposing edge pixels extracted are classified as candidate crack pixels if the set of pixels located between the opposing pixels (q_{ij}) satisfies the following conditions:

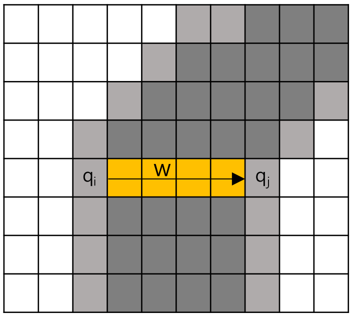


FIGURE 3. Opposing edge pixel searching process.

1. The crack width (w) must be smaller than the maximum threshold value ($\mathrm{w_m}$), as expressed by Equation (3).

$$w < w_m \tag{3}$$

2. If the crack width (w) is greater than or equal to the maximum threshold value ($\mathrm{w_m}$), the average Frangi value (fg_{avg}) should be greater than the threshold value (fg_t), as expressed by Equation (4).

$$fg_{avg} > fg_t \tag{4}$$

Condition 2 enables a more accurate classification of crack candidate regions by considering the Frangi filtering value for crack region enhancement. The average Frangi filtering value can be calculated as that of the opposing edge pixels, as expressed by Equation (5).

$$fg_{avg} = \frac{1}{w}\sum\nolimits_{m=i}^{j} fg\,(m) \tag{5}$$

Equation (6) expresses the process of classifying the opposing edge pixels as the crack candidate region (C).

$$C = C \cup \{q_{ij}\} \tag{6}$$

Narrow-width segments of a crack have a low intensity, which lowers their Frangi filtering values. To prevent such segments from being classified as non-crack regions, we applied only Condition 1 to narrow-width segments without considering their Frangi filtering values.

Additionally, we generated a width map (WM) for aspect ratio filtering of the CWT-based crack width measured. A width map is the alignment of the midpoints of the crack width, as expressed by Equation (7). A width map was initialized as 0, and the crack width was assigned to the width map whenever the conditions for a candidate crack were satisfied.

$$WM\left(\frac{q_i + q_j}{2}\right) \leftarrow w \tag{7}$$

The flowchart in FIGURE 4 explains the proposed CWT method. The process for obtaining crack edge information refers to the process of extracting an edge from a crack image and calculating its gradient information. The CWT is performed on all edge pixels extracted, in order to generate the crack candidate image and width map.

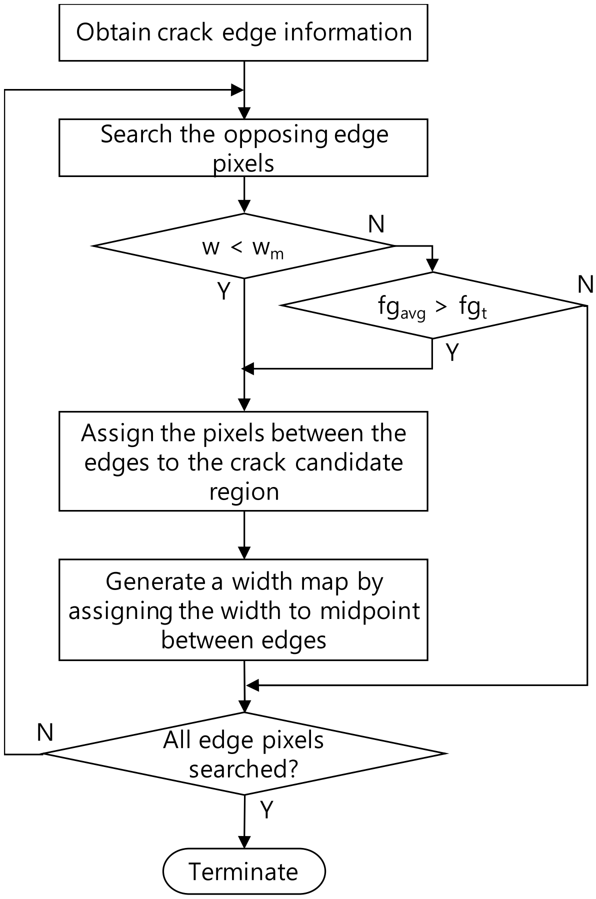

Obtain crack edge information
Search the opposing edge pixels
w < w_m
N
Y
fg_avg > fg_t
N
Y
Assign the pixels between the edges to the crack candidate region
Generate a width map by assigning the width to midpoint between edges
N
All edge pixels searched?
Y
Terminate

FIGURE 4. Flowchart of the crack width transform (CWT) process.

B. ASPECT RATIO FILTERING

The crack candidate regions extracted by the CWT contain various noise sources. For noise removal, we propose aspect ratio filtering, drawing on the typical morphological feature of all cracks, i.e., a high aspect ratio (length>width). To enable object classification based on a morphological feature, numerical values must be assigned. Many studies have employed circularity as a morphological feature for crack classification. Circularity values range between 0 and 1, where values closer to 1 indicate rounder objects, and vice versa. Crack patterns have circularity values closer to 0 than to 1. The morphological feature of a crack can be quantified indirectly by using its circumference and area based on circularity. Use of this indirect measurement method is necessary because it is difficult to measure the crack width and length in the crack detection process.

Crack width and length can be measured using the width map generated during the proposed CWT, as mentioned above. The crack width and length are measured as follows. First, pixels adjacent to a crack candidate region are grouped together. The number of pixels included in a group (G_i) is the group's area ($\mathrm{A_{G_i}}$), and its width (w_{G_i}) and length ($\mathrm{h_{G_i}}$) can be calculated as per Equations (8) and (9).

$$w_{G_i} = mean\left(\left\{WM\left(x_j\right) | WM\left(x_j\right) > 0 \wedge x_j \in G_i\right\}\right) \tag{8}$$

$$h_{G_i} = A_{G_i} / w_{G_i} \tag{9}$$

Here, x_j is the coordinates of a two-dimensional (2D) array or the image.

The aspect ratio of each group (rt_{G_i}) can be calculated as per Equation (10).

$$rt_{G_i} = h_{G_i} / w_{G_i} \tag{10}$$

During the process of aspect ratio filtering, a group with an aspect ratio (rt_{G_i}) greater than the threshold value (rt_t) is classified as a crack region (Q), as expressed by Equation (11).

$$Q = \{G_i \mid rt_{G_i} > rt_t\} \tag{11}$$

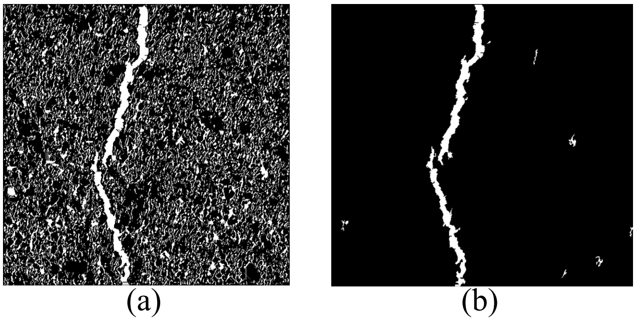


FIGURE 5. Result of noise removal using aspect ratio filtering: (a) a crack candidate region classified by CWT; (b) After aspect ratio filtering. (original image: No.17(TABLE 3))

FIGURE 5 shows the result of noise removal through aspect ratio filtering. The image in FIGURE 5(a) is a candidate crack region extracted by CWT, which consists of the crack region and a large amount of background noise. After applying aspect ratio filtering to FIGURE 5(a), only the crack region is shown in FIGURE 5(b), with practically no background noise.

C. CRACK REGION SEARCH

During aspect ratio filtering, not only noise, but also those crack segments with low aspect ratios are removed, because such segments have low intensity which impedes effective pixel extraction. Consequently, such cracks are extracted discontinuously, comprising multiple shorter cracks. Since these small crack regions have smaller aspect ratios and are hence likely to be removed during aspect ratio filtering, we propose a crack region search technique to identify and restore such omitted crack regions.

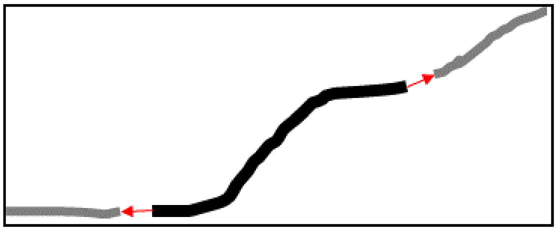


FIGURE 6. Crack region restoration with crack region search.

FIGURE 6 illustrates the crack region search process. The crack regions omitted from a discontinuous (i.e., incompletely extracted) crack are located along the crack propagation path. The omitted crack segments are thereby restored by searching the crack candidate regions located on the crack propagation path. The crack region search consists of two main steps: crack tip search, in which the tip of the discontinuous crack is searched and its direction is determined; and crack region search, in which the removed crack segment is restored by bridging it with the crack candidate region located along the direction of the main crackpropagation.

For the crack tip search, we used the width map tracing the midpoints of the crack width. To enable crack tip search, the crack must be thinned to the one-pixel-wide midline and the pattern of the crack tip must be determined. The width map served this purpose because it represents the midline linking the midpoints of opposing edge pixels placed along the crack propagation path.

However, a midline contains segments that are discontinuous or thicker than one pixel. We propose a crack tip search technique to connect the discontinued midline, thin the midline, and search the crack tip. The proposed crack tip search technique is designed to connect disconnected midpoints by scaling the corresponding coordinates to densify those midpoints. The following depicts the workflow of the crack tip search.

Generation of alignment S with densified midpoints:

1. Compute a new coordinate s by dividing coordinate x of the midpoint pixel by the proportionality constant.

2. Assign variable u if alignment S(s) has no assigned value. Variable u, which refers to the order of assigning value to alignment S, is used as the index of the 2D list U.

3. Store the original coordinates of coordinate s in the 2D list U[u].

Crack tip search on alignment S:

4. Search the crack tip on alignment S.

5. Extract the original coordinate $\mathrm{x_i}$ using value u at the tip $\mathrm{S(s_e)}$.

6. Compute the tip coordinates from the original coordinates ($\mathrm{x_i}$).

FIGURE 7 illustrates the midline extraction process for crack tip search. FIGURE 7(a) is the width map (WD) generated, where x_i is the coordinate value assigned to any given midpoint. The midpoints dispersed in FIGURE 7(a) were mapped on alignment S (FIGURE 7(b)) by scaling x_i to densify the dispersed midpoints. For midpoint coordinate scaling, the value of scaled coordinate (s_i) was computed by dividing x_i by the scale constant (sc), using Equation (12).

$$s_i = x_i / sc \tag{12}$$

If alignment S(s) has no assigned value, variable u is assigned as per Equation (13). Variable u, which refers to the order of assigning value to alignment S, is used as the index of the 2D list U.

$$S(s_i) = u_k \tag{13}$$

As shown in FIGURE 9(b), original coordinate x_i, i.e., the pre-transform coordinate of s_i, is added to list U[uk].

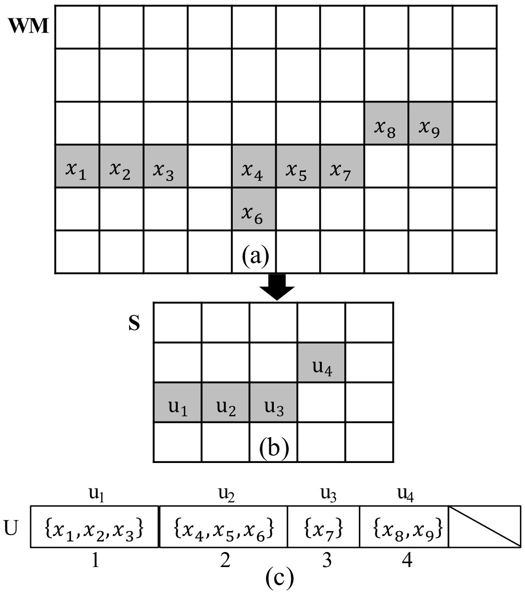


FIGURE 7. Midline extraction process for crack tip search: (a) Width map (WM) and midpoint coordinates (x_i); (b) Alignment S and the values assigned to it (u_i); (c) List U and the assigned midpoint coordinates (x_i).

The 2D list U stores the pre-transform coordinates and is used for searching the crack tip coordinates of the original image (WM) from the crack tip detected in the alignment S region.

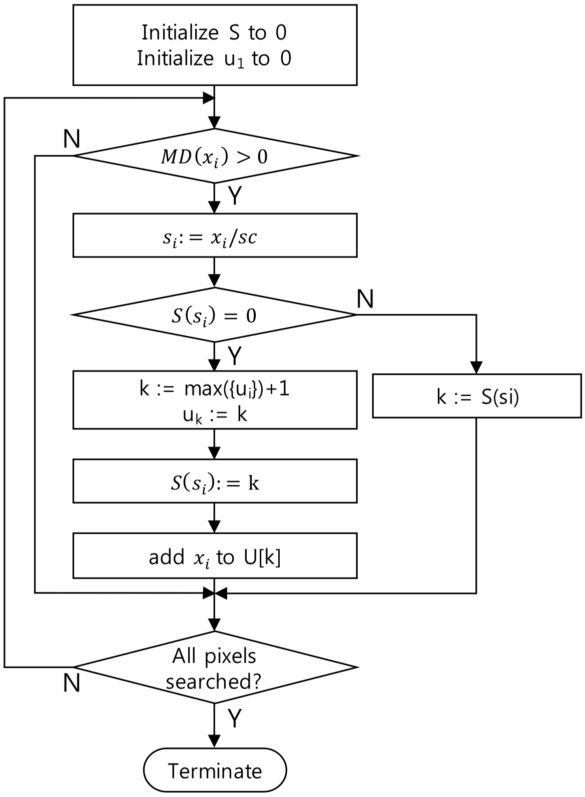


FIGURE 8. Flowchart for generating the arrangement S and list U.

The flowchart in Figure 8 shows the process for generating the midline. This is performed to obtain alignment S of densified pixels and a 2D list U of the original, pre-mapping coordinates of the midline of alignment S.

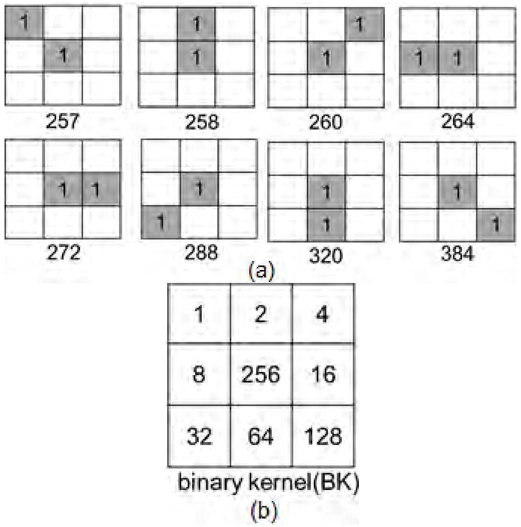


FIGURE 9. Eight crack tip patterns and the binary kernel: (a) Eight crack tip patterns and their respective correlation values; (b) Binary kernel (BK) to locate the crack tip.

To search for the crack tip from the midline, we defined eight crack tip patterns as shown in FIGURE 9(a), and computed the correlations between the binary kernel (BK) and alignment S_B, as expressed by Equation (16).

$$S_B(s_j) = \begin{cases} 1 & S(s_j) > 0 \\ 0 & otherwise \end{cases} \tag{14}$$

$$S_B(s_j) = thinning\,(S_B) \tag{15}$$

$$V = conv\,(S_B,\ BK) \tag{16}$$

Here, S_B is alignment S converted into binary form (values 0 and 1), as expressed by Equation (14), whereby 1 was assigned to the midline; and *thinning* is the thinning operation performed to make the S_B alignment one-pixel-thick. FIGURE 9(b) shows the value table for the binary kernel *BK*. V denotes the convolution result and is an alignment of the same size as S_B.

We used a lookup table (*LT*) to identify the crack tip and determine its direction after defining it as shown in Equation (17) by assigning a direction code (*dc*) to each index (*idx*). Figure 10 shows the direction code.

$$LT\,[idx] = \begin{array}{l} \quad dc \quad idx \\ \begin{cases} 4 & 257 \\ 5 & 258 \\ 6 & 260 \\ 3 & 264 \\ 7 & 272 \\ 2 & 288 \\ 1 & 320 \\ 8 & 394 \\ 0 & otherwise \end{cases} \end{array} \tag{17}$$

On alignment S, the crack tip can be identified using the lookup table (*LT*) and convolution value (*V*), as expressed

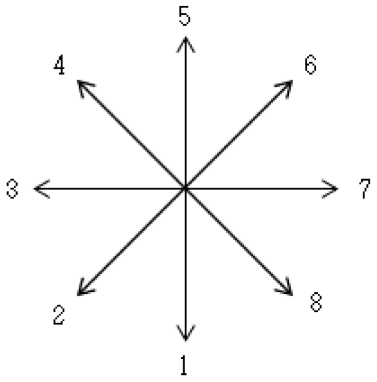


FIGURE 10. Direction code (dc) indicating the crack tip directions.

by Equation (18).

$$E = \{(s_i, dc_i) | LT [V (s_i)] = dc_i, dc_i > 0 \tag{18}$$

Here, E denotes a pairof crack tip coordinates (s_i) and their respective direction codes (dc_i).

We used list U to identify the crack tip of the original image that matches that found on alignment S. The coordinate x_i of the original image stored in list U [$\mathrm{u_k}$] can be retrieved based on the value ($\mathrm{u_k}$) assigned to the crack tip (s) on alignment S. Of the coordinates on the original image, one is selected for the crack tip that matches its direction.

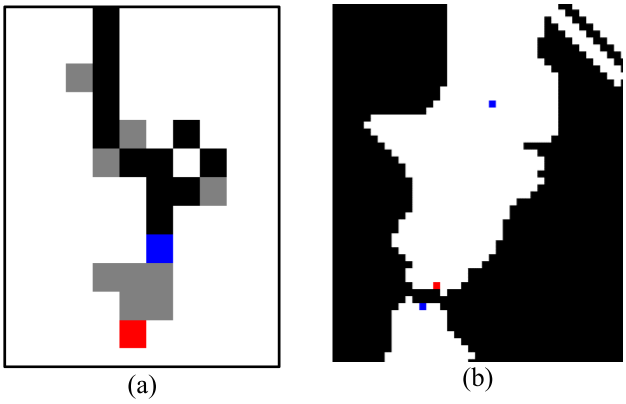


FIGURE 11. Result of crack tip search: (a) Crack tip search on alignment S; (b) Crack tip position on the original image (■ midline on S, ■ $\mathrm{S_B}$, 🟦 crack tip, 🟥 crack tip detected in the repeated search).

FIGURE 11 shows the results of the crack tip search. FIGURE 11(a) shows the pixel-level crack tip search result on alignment S. Image $\mathrm{S_B}$ is generated for the crack tip search by converting and thinning alignment S to a binary format. Some of the midline pixels are removed in the thinning process, which can lead to errors. To minimize such thinning-induced errors, we repeated the crack tip search, starting from the pixel located at the farthest end along the crack propagation path in a region selected to take account of the crack direction. The position of the crack tip extracted in this way matches well with the position of the crack tip detected on the original image in FIGURE 11(b).

A crack region is darker (lower intensity) than its surrounding area. Therefore, after identifying the pixels located along the crack propagation path, we added the pixel with the lowest intensity as a region expanded from the crack tip.

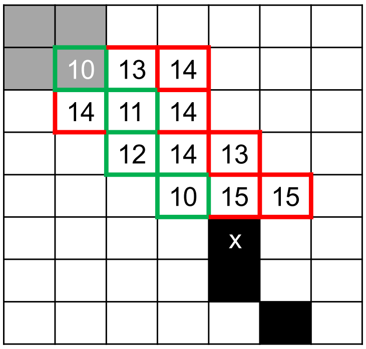


FIGURE 12. Crack region restoration through region search (x crack tip, ■ : crack region, □ search region, □ expanded region, ■ missing crack region).

FIGURE 12 illustrates the process of searching for and restoring the missing crack regions. In this example, we searched three pixels adjacent to crack tip X in the crack region, and added the pixel with the lowest intensity as the expanded region. From this expanded region, we repeated the pixel search and addition until we found all omitted (removed) crack regions. The three adjacent pixels searched can be expressed by Equation (19).

$$N_3^{dc}(p), \quad p = (r, c) \tag{19}$$

where $N_3^{dc}(p)$ represents the three pixels adjacent to the current pixel (p) as per the direction code (dc). Pixels corresponding to the direction code (dc) are expressed as in Equation (20).

$$N_3^{dc}(p) = \{(r+i, c+j)\}, \tag{20}$$

$$\begin{cases} dc = 1, & i \in 1,\ j \in [-1,\ 1] \\ dc = 2, & i \in [0,\ 1], j \in [-1,\ 0] \\ dc = 3, & i \in [-1,\ 1], j \in -1 \\ dc = 4, & i \in [-1,\ 0], j \in [-1,\ 0] \\ dc = 5, & i \in -1, j \in [-1,\ 1] \\ dc = 6, & i \in [-1,\ 0], j \in [0,\ 1] \\ dc = 7, & i \in [-1,\ 1], j \in 1 \\ dc = 8, & i \in [0,\ 1], j \in [0,\ 1], \end{cases}$$

$$p \notin N_3^{dc}(p) \tag{21}$$

Of the three adjacent pixels, the pixel with the lowest intensity was taken for region expansion. The region search was iterated until a new region was found. The process of searching the adjacent pixels and extending the region can be expressed by Equation (22).

$$\begin{aligned} \varepsilon_{k+1} &= argmin\,(g\,(x_i)), \quad x_i \in N_3^{dc}(\varepsilon_k) \\ k &= 1, \ldots, n, \quad \varepsilon_1 = x,\ x \in E \end{aligned} \tag{22}$$

Here, ε is the expanded region and g is the grayscale value of the image. The region search is iterated until a new group is found as per Equation (23), and the crack tip was assigned as the initial coordinate (ε_1)

$$\{\varepsilon_k\} \cap G_j \neq \emptyset, \quad \varepsilon_1 \in G_i, \; G_i \neq G_j \tag{23}$$

where $\mathrm{G_j}$ is the crack candidate group identified by region search.

An expanded region was classified as a new region if it satisfied the following conditions:

1. The length of the expanded region is shorter than the minimum length threshold (T_L) (Equation (24)).

$$\#\{\varepsilon_k\} < T_L \tag{24}$$

2. If condition 1 is not satisfied, the average Frangi filtering value ($f_{g\varepsilon}$) of the expanded region must be greater than the Frangi threshold value (fg_t) (Equation (25)).

$$fg_\varepsilon > fg_t \tag{25}$$

A group searched satisfying conditions 1 and 2 was classified as a crack region, as expressed by Equation (26).

$$Q = Q \cup \{G_j \tag{26}$$

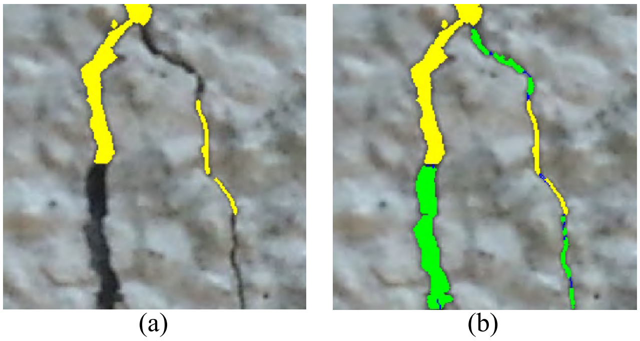


FIGURE 13. Crack region restored through region search: (a) Before restoration, (b) After restoration (🟨 crack region, 🟩 restored crack region, 🟦 expanded region).

FIGURE 13 shows the result of restoring the crack region that was removed during the region search process. The blue and green regions represent the region expanded from the crack tip and the restored region, respectively. The image shows that the missing region could be restored efficiently through region search.

D. CRACK REGION RESTORATION THROUGH HOLE FILLING

CWT-based classification of crack candidate regions is conducted on the edges detected. This suggests that the segments in which no crack edges are extracted will likely be excluded from being classified as candidate crack regions. We propose-hole filling as a method of restoring such missing segments. Hole filling refers to the process of filling up the foreground that forms a closed loop.

First, we generated a closed loop by bridging the gap of the broken edge in a missing (undetected) segment, and restored the closed loop by carrying out a hole filling operation. Bridging the gap of the broken edge in a missing segment must be preceded by detecting the start and end points of the broken edge. To detect the start and end points, we used alignment (Y), on which a larger value is assigned to the broken edge than other regions, and the contour tracing technique, as expressed by Equation (27).

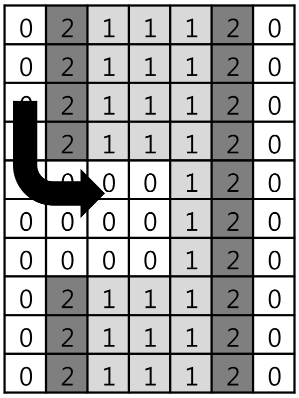
$$\begin{aligned} L = \{&(b_i,\ b_j)\,|Y(b_i)-Y(b_{i+1})>0,\ Y(b_j)\\ &-Y(b_{j+1})<0\},\\ i+1 < j,&\quad b_i \in Contour(Q),\\ Y(edge) &> Y(otherwise) \end{aligned} \tag{27}$$

where *Contour* (I) refers to the contour tracing technique by which a contour pixels b_i is extracted from an object in crack region Q. L is a set of the pairs of start and end points.

A continuous edge is generated by connecting the pairs of start and end points, and such a continuous edge (b_{ij}) is included in the crack region (Q).

$$\begin{aligned} Q &= Q \cup \{b_{ij}\} \\ Q &= FillHoles(Q) \end{aligned}$$

where b_{ij} is a set of pixels connecting the start point (b_i) and the end point (b_j). *FillHoles*(Q) is the hole filling technique, which is performed in the crack region (Q) with a continuous edge.



0	2	1	1	1	2	0
0	2	1	1	1	2	0
0	2	1	1	1	2	0
	2	1	1	1	2	0
	0	0	0	1	2	0
0	0	0	0	1	2	0
0	0	0	0	1	2	0
0	2	1	1	1	2	0
0	2	1	1	1	2	0
0	2	1	1	1	2	0

FIGURE 14. The start and end point search operation for edge connection.

FIGURE 14 illustrates the process of start and end point detection of a discontinued edge. In the alignment, 1 is assigned to a crack region, and 2 to an edge region. In contour tracing, the pixels located on the contour are extracted sequentially along the contour of the crack region. In the discontinued parts, adjacent contour pixels show differences larger or smaller than 0 in the discontinued parts, and this principle is used for extracting the start and end points.

Missing regions can be restored by performing a hole filling operation on closed loops generated by connecting discontinued segments of an edge, as illustrated

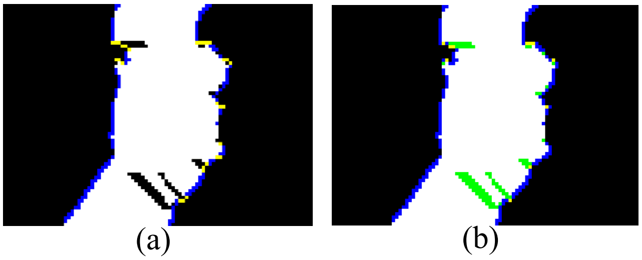


FIGURE 15. Restoration of discontinued parts of an edge through: (a) Connection of discontinued parts of the edge; (b) Restored crack region after the hole filling operation (🟦 detected edge, 🟨 connected edge, 🟩 restored edge).

in FIGURE 15. FIGURE 15(a) shows the connected segments highlighted in yellow. FIGURE 15(b) shows the crack region restored by carrying out the hole filling operation on FIGURE 15(a), where the green regions indicate the restored regions. It could be thus verified that hole filling improved the crack detection accuracy by restoring the missing regions.

E. RELATIVE THRESHOLDING

There remains residual noise after aspect ratio filtering. In addition, not only do pixels added in the region search and hole filling process enhance the true positive rate (TPR), but they also enhance noise. To eliminate this problem of noise, we propose relative thresholding

Cracks are characterized by lower intensities (gray values) and higher Frangi filtering values compared with the background. Selection of an appropriate threshold value is an efficient way to distinguish cracks from background, but determining the threshold value is not a simple task. For the proposed relative thresholding, we selected the threshold value from the previously classified crack regions in the following manner:

The average Frangi filtering value (fg_{G_i}) and intensity distribution of each group (G_i) in the crack region are computed using Equation (28).

$$\begin{aligned} fg_{G_i} &= mean\left\{fg\left(x_j\right)|x_j \in G_i\right\}, \\ G_i &\in Q \end{aligned} \tag{28}$$

The highest (mf) of the average Frangi filtering values of individual groups and the lowest gray value (mg) are determined using Equation (29) and Equation (32), respectively.

$$mf = max(\{fg_{G_i}) \tag{29}$$

The threshold value is determined by multiplying the maximumFrangi filtering value (mf) and the minimum gray value(mg) by their respectiveproportionality constants (kf and kg) as expressed by Equation (30) and Equation (33).

$$T_{fg} = mf \times kf \tag{30}$$

Crack region reclassification was performed using both Frangi filtering values and the intensities (gray values) of the classified regions, and the threshold value for the reclassification was computed using the corresponding intensity (gray value) information of the classified regions as per Equations (31)-(33).

$$g_{G_i} = mean\left\{g\left(x_j\right)|\, x_j \in G_i,\, G_i \in Q\right\} \tag{31}$$

$$mg = min\left(\{g_{G_i}\}\right) \tag{32}$$

$$T_g = max(mg \times kg, g_{back} - g_{std}) \tag{33}$$

where g_{back} and g_{std} denote the background intensity and standard deviation, respectively, and T_{g} denotes the threshold value.

Based on both threshold values, we reclassified each group as a crack region, as expressed by Equation (34).

$$Q = \{G_i | fg_{G_i} > T_{fg} \wedge g_{G_i} < T_g\} \tag{34}$$

The classification ofcrack regions detected in the previous processhas a relatively high accuracy. Of the classified crack regions, the one with the highest Frangi filtering value is the region with the highest likelihood of being a crack. Since the Frangi filtering value varies depending on the foreground–background contrast, a fixed threshold value cannot be applied to all images. The proposed relative thresholding was designed to set the threshold value taking account of image-dependent Frangi filtering value of a crack region.

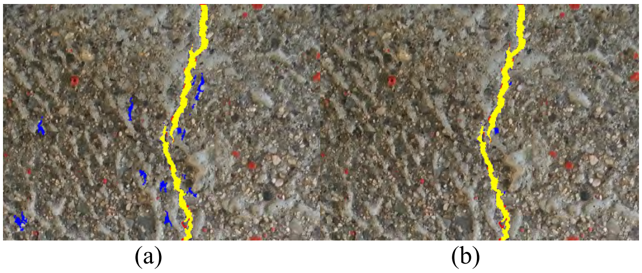


FIGURE 16. Noise removal using relative thresholding: (a) Image captured before relative thresholding; (b) Image captured after removing noise by applying relative thresholding to (a)(🟨 True Positive(TP), 🟦 False Positive(FP), 🟥 False Negative(FN)).

FIGURE 16 shows pre- and post-thresholding images. In FIGURE 16(a), a pre-thresholding image, contains noise leading to false positive (FP) errors. FIGURE 16(b), a pre-thresholding image with crack regions reclassified by applying relative thresholding, shows that sources of FP errors are removed and classification accuracy is improved.

IV. DATASET PREPARATION

We collected real crack image data to test and analyze the image-based crack detection results. We used 18 images with 1920 × 1080 resolution sequentially numbered from 1 to 18 (see the appendix).

The existing crack image dataset does not have labels that indicate the true value [23]. To analyze the performance of a crack detection technique, the true values for the crack region on the image data should be known. To obtain true values, we performed labeling operation to classify the crack image into crack and background regions. It is not easy to distinguish a crack from its background on an image because of irregular intensities of the crack and difficulties and ambiguities associated with selecting the edge taking account of the gradient and morphology of the region inside the crack. The accuracy of the true value manually determined plays a crucial role in testing a crack detection technique. Given the near impossibility of deriving a perfectly accurate true value, we used specific labeling rules to classify crack and background regions.

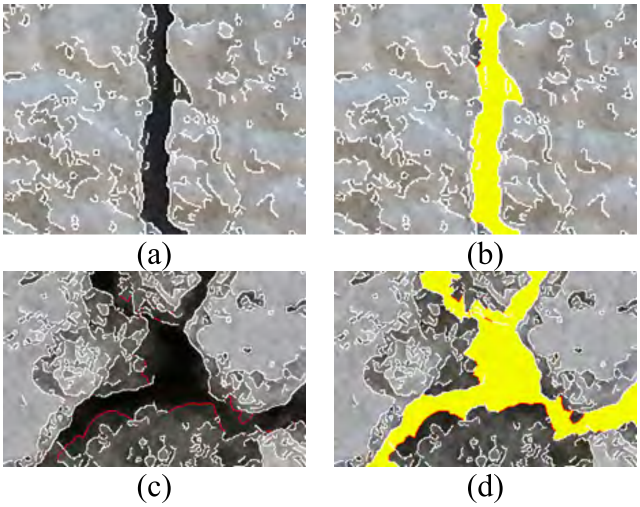


FIGURE 17. Crack image data labeling operation: (a) Image with clear edges; (b) Result of labeling (a); (c) Image with unclear edges; (d) Result of labeling (c)(🟨 labeled crack region, 🟥 randomly selected edge).

FIGURE 17 illustrates the process of crack image labeling. For the selection of a crack region, we applied a simple rule of extracting a crack edge and selecting the segment matching the true value. The inner region of the selected crack edge (i.e. the space between a pair of edges) was subject to hole-filling, thereby bridging the discontinued segments arbitrarily (red line in FIGURE 17). Compared with FIGURE 17(a), FIGURE 17(b) has more arbitrarily selected regions due to the presence of many regions with missing segments. Given the difficulties associated with accurately defining and classifying crack regions, errors in true values should be considered when analyzing the accuracy of crack detection using real images. Since the image data used in this study consist of original crack images and labeling, we named these images "edge-based labeled crack images" (ELCI), and these are available online for free use [24].

V. RESULT

We used a synthetic image to compare the accuracy of crack width measurement between the conventional and proposed techniques. Compared with real images, a synthetic image has the advantage of defining a known crack width.

FIGURE 18 compares the performance of CWT technique and line enhancement filtering based on a synthetic image. The synthetic image was generated in a manner that allows accurate measurement of crack width by generating images against a real concrete background. The concrete background has a blob-shaped noise distribution with irregular illumination intensity and low gray level due to shades, and the synthetic crack was configured to have a width of 30 pixels and gray value of 50.

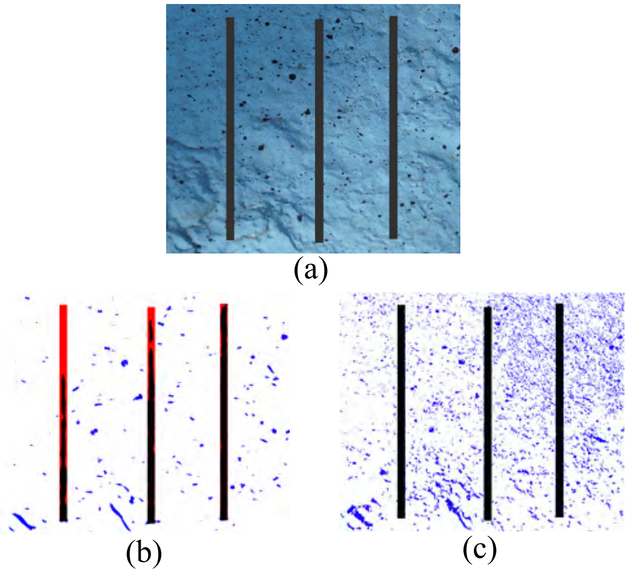
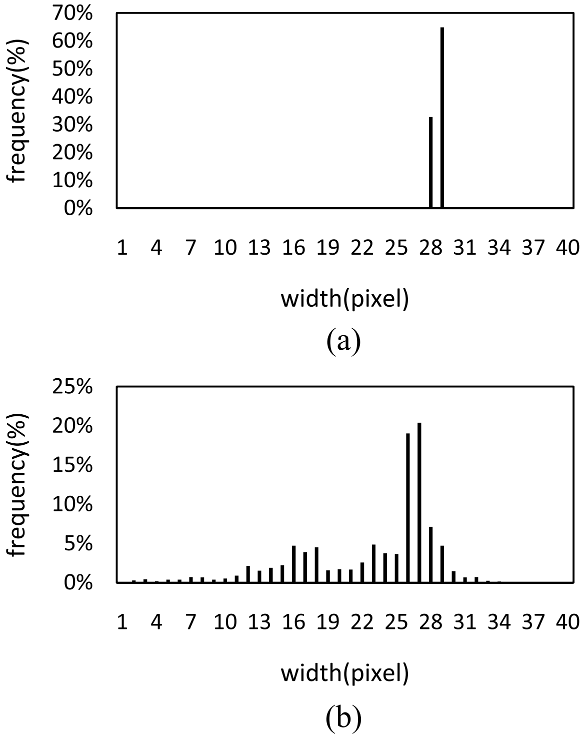


FIGURE 18. Performance comparison between line enhancement filtering and crack width transform (CWT) using a synthetic image: (a) Synthetic image; (b) Detection result using line enhancement filtering; (c) Detection result of CWT (■ True Positive, 🟦 False Positive, 🟥 False Negative).

Looking at the line enhancement filtering result, false positive (FP) pixel extraction errors are displayed near the edges. In particular, a large undetected portion is displayed in the upper left part where the background intensity is low due to shadow. By contrast, CWT yielded consistent and accurate results for crack region extraction irrespective of background intensity changes or noise.

In FIGURE 19, the histograms representthe results of crack width measurement shown in FIGURE 18. The histogram for CWT technique (FIGURE 19(a)) shows a very high concentration of pixels closest to the width of the real crack (30 pixels), whereas with line enhancement filtering (FIGURE 19(b)) the measured width values are broadly dispersed: 98% of the CWTmeasurements have an error range within 6.7%, compared with only 15% of the line enhancement filteringmeasurements. This result indicates that CWT is capable of more consistent and accurate crack width measurement than line enhancement filtering, which showed greatmeasurement variation between crack segments.

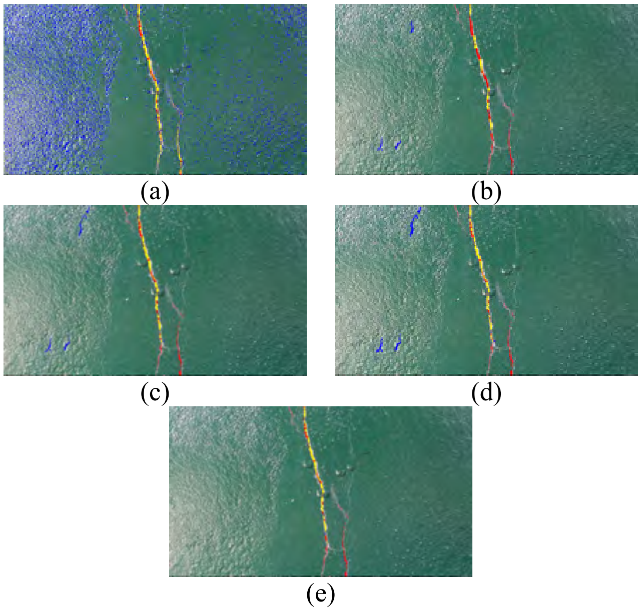
We then analyzed crack detection performance using a real image. FIGURE 20 shows the step-wise detection results using the proposed technique, and FIGURE 21 presents the changes in accuracy and true positive rate throughout the five crack detection steps of the proposed technique. In the crack candidate extraction (CWT_C) process, accuracy of crack extraction is very low due to the noise sources extracted with cracks, as shown in FIGURE 20(a); however, most of the extracted cracks coincided with the real ones. In the crack



(a)

(b)

FIGURE 19. Crack width measurement histograms for CWT and line enhancement filtering (Frangi).



(a)

(b)

(c)

(d)

(e)

FIGURE 20. Crack detection at each step using the proposed method: (a) CWT-based extraction of candidate crack region; (b) aspect ratio filtering; (c) crack region search; (d) hole filling; (e) relative thresholding (🟨 True Positive, 🟦 False Positive, 🟥 False Negative).

candidate extraction process, average true positive rate(TPR) and accuracy were 0.80 and 0.15, respectively.

Aspect ratio filtering (ARF) was performed to eliminate FP errors from the extracted candidate cracks. We computed the crack width and region area obtained in the CWT process, and eliminated the regions with low aspect ratios. Aspect ratio filtering markedly improved the accuracy from 0.15 to 0.84 by removing most noise, as shown in FIGURE 20(b), but TPR decreased to 0.59.

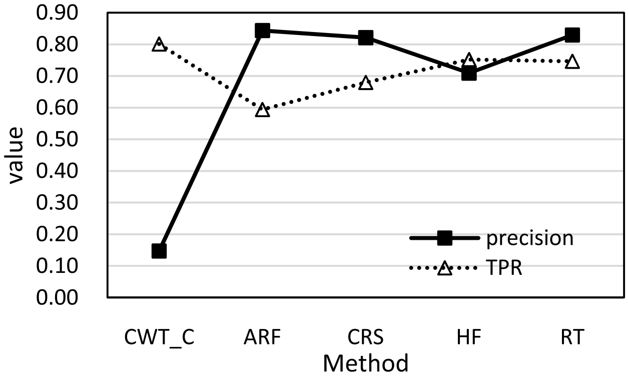


FIGURE 21. Changes in precision and true positive rate (TPR) (CWT_C: CWT-based crack candidate extraction, ARF: aspect ratio filtering, CRS: crack region search, FH: hole filling, RT: relative thresholding).

To restore the crack regions removed in the aspect ratio filtering process, we performed crack region search (CRS) according to the following workflow: identifying the crack tip and its direction in the extracted crack region searching for missing (removed) crack regions starting from the crack tip, based on its direction and intensity information restoring the missing sections detected as crack regions if specified conditions are satisfied. By applying the crack region search technique, the accuracy was maintained and TPR increased by approximately 9%.

We carried out hole filling (HF) to add crack pixels that could not be extracted in the CWT process. One reasonfor the failure to extract candidate crack pixels using CWT is the incomplete extraction of the edges. To solve this problem, we searched for discontinuous edges and connected the regions intended for restoration into closed loops. A region turned into a closed loop was classified as a crack region by the hole filling algorithm, which then improved TPR to 0.75 on average.

Hole filling is a process of adding pixels. By expanding the crack regions, the TPR increases whereas accuracy decreases inversely. To compensate for the decreased accuracy, we performed relative thresholding (RT). This processchecked whether the intensity (gray value) and Frangi filtering value of the detected region satisfied their respective threshold ranges, which were obtained by multiplying the minimum gray valueand the maximumFrangi filtering value by their respectiveproportionality constants. By applying this post-treatment, accuracy increased from 0.71 to 0.83 while maintaining the average TPR at 0.75.

Table 2 compares the accuracy of CWT and a conventional technique based on line enhancement filtering and circularity, showing that the proposed technique achieves 0.36 greater accuracy on average.

TABLE 2. Accuracy of crack width transform (CWT) technique versus line enhancement filtering technique.

No.	TPR	PR_c	PR_f	ΔP
1	0.93	0.89	0.57	0.32
2	0.80	0.90	0.65	0.25
3	0.55	0.85	0.91	-0.06
4	0.92	0.74	0.10	0.64
5	0.62	0.83	0.31	0.52
6	0.61	0.86	0.80	0.06
7	0.37	0.56	0.81	-0.25
8	0.89	0.87	0.18	0.69
9	0.82	0.83	0.66	0.17
10	0.78	0.87	0.50	0.37
11	0.93	0.85	0.10	0.75
12	0.66	0.82	0.36	0.46
13	0.61	0.82	0.76	0.06
14	0.81	0.78	0.50	0.28
15	0.90	0.78	0.08	0.7
16	0.58	0.80	0.79	0.01
17	0.96	0.90	0.03	0.87
18	0.88	0.87	0.14	0.73
Mean	0.76	0.82	0.46	0.36

PRc: Precision of the CWT; PRf: Precision of line enhancement filtering; ΔP: PRc-PRf

FIGURE 22 shows the crack detection results of the two techniques, with the TPR being the same. Both have a TPR of 0.82, which means that they extracted 82% of the actual crack regions. Looking at FIGURE 22(a) and (c), CWT (FIGURE 22(a)) seemingly produces more segments with missing crack regions(red) than does line enhancement filtering (FIGURE 22(b)). In the close-up images, however, CWT (FIGURE 22(c)) has more accurately extracted the pixels located along the edges, with practically no errors, whereas the line enhancement filtering technique (FIGURE 22(d)) produces many missing pixels (red) along the edges.

This is a typical result indicative of the difference between these two techniques. CWT is a technique designed for accurate crack width, aiming at accurate crack detection along the edges by using edge-based processes. CWT incurred detection errors by missing certain crack segments, but detected almost all edge pixels, demonstrating that it is a highly efficient method for crack width measurement. The detection results for all images are listed in the appendix.

VI. CONCLUSION

This study proposes crack width transform (CWT), an edge-based technique for automatic detection and accurate width measurement of cracks in concrete. We analyzed the limitations of the conventional line enhancement filteringtechnique and showed that line enhancement filtering cannot consistently measure cracks under conditions of variable illumination contrast.

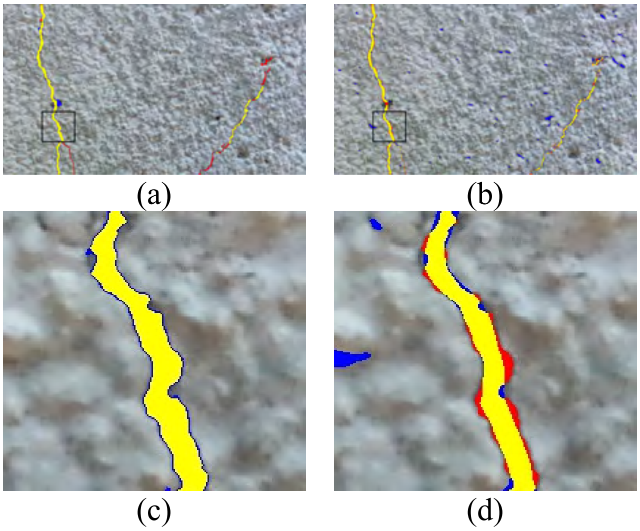


FIGURE 22. Comparison of crack detection results between CWT and line enhancement filtering: (a) CWT; (b) Line enhancement filtering and circularity; (c) Expansion of the box region in (a); (d) Expansion of the box region in (b) (🟨 TP, 🟦 FP, 🟥 FN).

To overcome the limitations of the conventional technique, we proposed an edge-based crack detection technique, which consists of five steps: CWT, aspect ratio filtering, crack region search, hole filling, and relative thresholding. CWT can efficiently extract edge pixels by identifying the opposing edge pixel for any given edge pixel and classifying it as a candidate crack pixel. Aspect ratio filtering is proposed for removing noise and improving accuracy using the width map generated in CWT. Crack region search was proposed for restoring the crack regions removed during aspect ratio filtering. In this process, the removed pixels could be identified and restored by searching the candidate pixels on the crack propagation path, thus improving the true positive rate (TPR) of crack detection. Hole filling restored crack regions that were undetected due to non-extraction of the corresponding edges. This process contributed to enhancing TPR by bridging the discontinuous edges and carrying out the hole filling operations. Relative thresholding was designed to remove residual noise based on threshold values determined using the intensity information of the extracted crack regions.

The proposed technique was tested using synthetic and real crack images. Its main advantages are three-fold: i) It can measure crack width consistently along all crack segments irrespective of visual contrast between crack and background; ii) It has higher measurement accuracy than the conventional technique; iii) It can extract crack edge pixels with high accuracy. The proposed edge-based CWT technique aims at accurate crack width measurement, with its most salient features being consistent crack width measurement and accurate edge pixel extraction. It is expected to contribute to ongoing safety monitoring of concrete structures by providing accurate crack width measurements.

APPENDIX

See Table 3.

TABLE 3. Results of step-wise crack detection using the proposed technique(🟨 TP, 🟦 FP, 🟥 FN) (Precision, TPR).

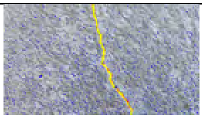
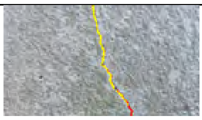
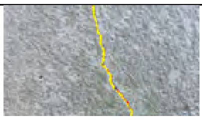
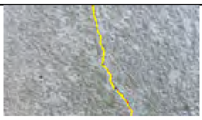
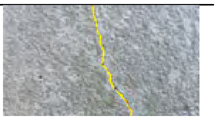
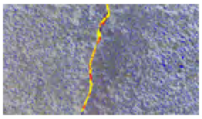
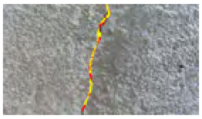
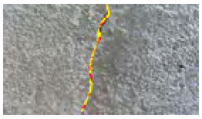
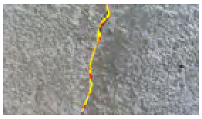
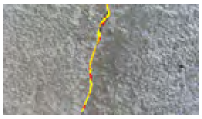
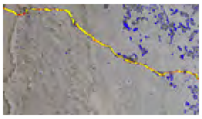
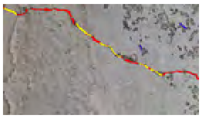
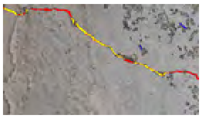
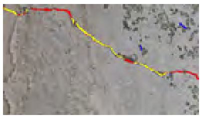
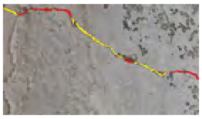
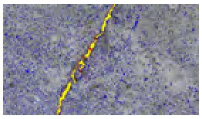
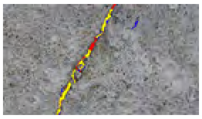
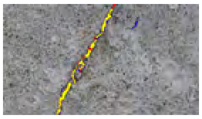
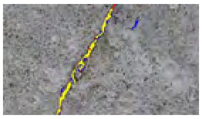
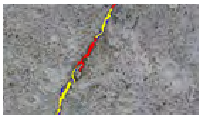
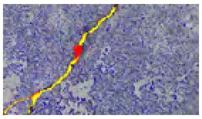
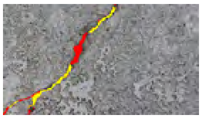
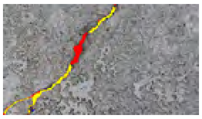
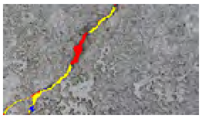
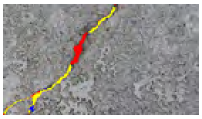
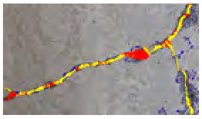
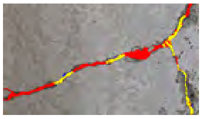
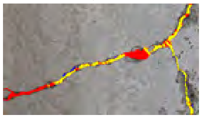
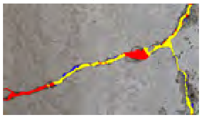
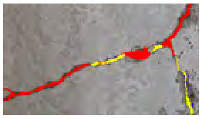
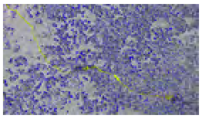
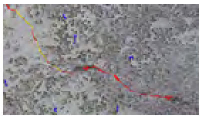
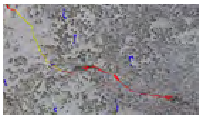
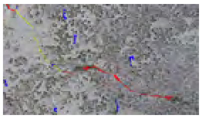
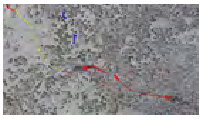
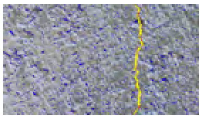
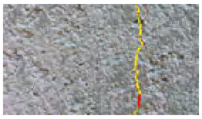
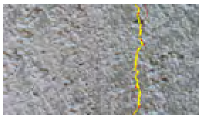
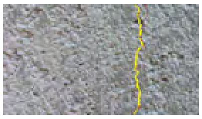
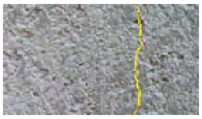
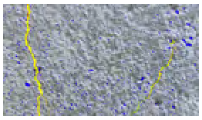
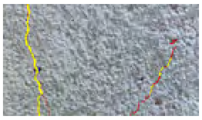
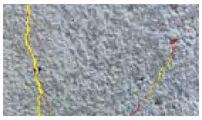
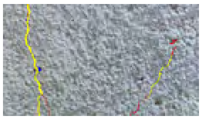
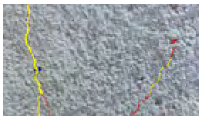
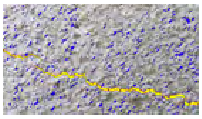
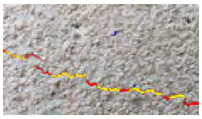
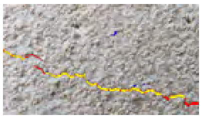
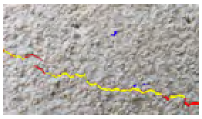
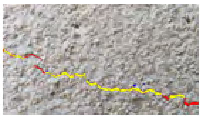
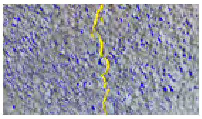
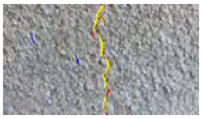
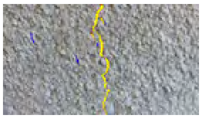
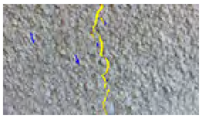
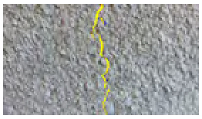
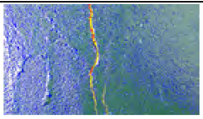
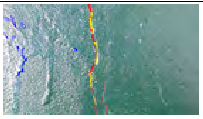
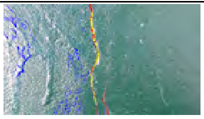
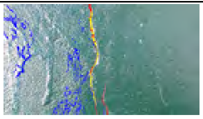
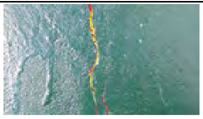
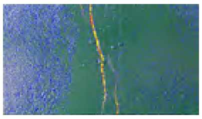
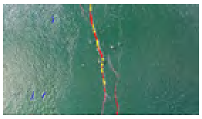
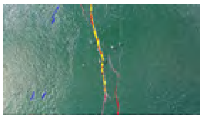
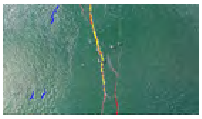
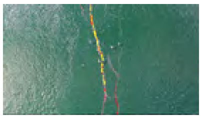
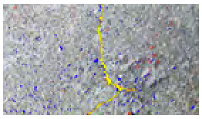
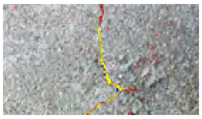
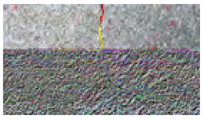
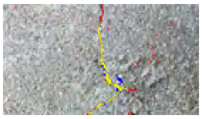
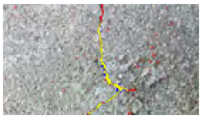
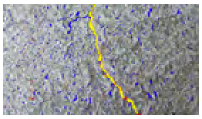
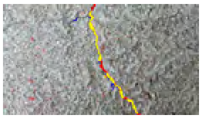
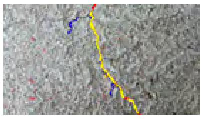
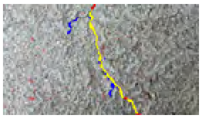
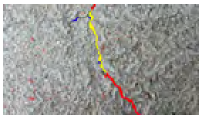
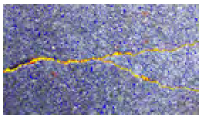
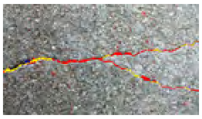
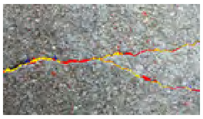
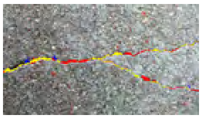
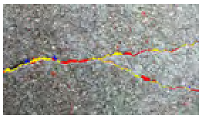
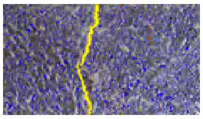
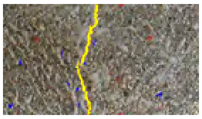
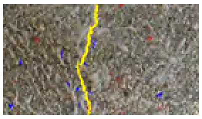
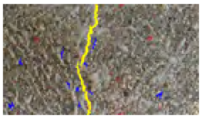
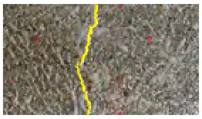
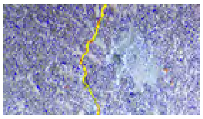
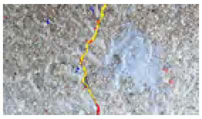
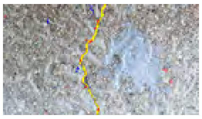
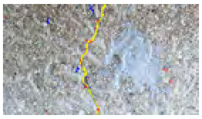
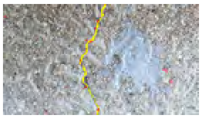
No	CWT_C	ARF	CRS	HF	RT
1	 0.16，0.87	 0.99，0.79	 0.99，0.86	 0.89，0.93	 0.89，0.93
2	 0.08，0.74	 1.00，0.68	 0.99，0.73	 0.89，0.80	 0.90，0.80
3	 0.37，0.79	 0.87，0.37	 0.88，0.50	 0.77，0.57	 0.85，0.55
4	 0.14，0.84	 0.83，0.73	 0.83，0.82	 0.73，0.92	 0.78，0.63
5	 0.11，0.71	 0.97，0.51	 0.95，0.57	 0.83，0.62	 0.83，0.62
6	 0.71，0.61	 0.92，0.35	 0.93，0.52	 0.86，0.62	 0.85，0.26
7	 0.04，0.74	 0.45，0.25	 0.45，0.31	 0.35，0.37	 0.56，0.37
8	 0.13，0.85	 0.99，0.71	 0.99，0.82	 0.87，0.89	 0.87，0.89
9	 0.33，0.90	 0.96，0.73	 0.96，0.79	 0.83，0.85	 0.83，0.82
10	 0.26，0.82	 0.93，0.56	 0.93，0.67	 0.84，0.78	 0.87，0.78
11	 0.12，0.88	 0.90，0.78	 0.87，0.87	 0.75，0.94	 0.85，0.93

TABLE 3. ***(Continued.)*** **Results of step-wise crack detection using the proposed technique(🟨 TP, 🟦 FP, 🟥 FN) (Precision, TPR).**

No	CWT_C	ARF	CRS	HF	RT
12	 0.03, 0.70	 0.40, 0.47	 0.20, 0.57	 0.15, 0.66	 0.82, 0.66
13	 0.04, 0.70	 0.69, 0.35	 0.70, 0.55	 0.55, 0.63	 0.82, 0.61
14	 0.31, 0.87	 0.88, 0.74	 0.84, 0.76	 0.72, 0.82	 0.78, 0.81
15	 0.23, 0.86	 0.89, 0.75	 0.81, 0.83	 0.69, 0.90	 0.81, 0.49
16	 0.08, 0.70	 0.92, 0.32	 0.90, 0.47	 0.78, 0.59	 0.80, 0.58
17	 0.08, 0.92	 0.82, 0.91	 0.78, 0.92	 0.68, 0.96	 0.90, 0.96
18	 0.06, 0.83	 0.88, 0.73	 0.87, 0.79	 0.75, 0.88	 0.87, 0.88

REFERENCES

[1] F. P. Zhou, "Time-dependent crack growth and fracture in concrete," Ph.D. dissertation, Dept. Building Technol., Lund Univ., Lund, Sweden, 1992.

[2] L.-C. Chen, Y.-C. Shao, H.-H. Jan, C.-W. Huang, and Y.-M. Tien, "Measuring system for cracks in concrete using multitemporal images," *J. Surv. Eng.*, vol. 132, no. 2, pp. 77–82, May 2006.

[3] J. M. W. Brownjohn, "Structural health monitoring of civil infrastructure," *Philos. Trans. Roy. Soc. A*, vol. 365, no. 1851, pp. 589–622, 2007.

[4] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Adv. Eng. Inform.*, vol. 29, no. 2, pp. 196–210, 2015.

[5] 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.

[6] Y. Fujita and Y. Hamamoto, "A robust automatic crack detection method from noisy concrete surfaces," *Mach. Vis. Appl.*, vol. 22, no. 2, pp. 245–254, 2011.

[7] M. R. Jahanshahi and S. F. Masri, "A new methodology for non-contact accurate crack width measurement through photogrammetry for automated structural safety evaluation," *Smart Mater. Struct.*, vol. 22, no. 3, p. 035019, Mar. 2013.

[8] T. Yamaguchi and S. Hashimoto, "Practical image measurement of crack width for real concrete structure," *Electron. Commun. Jpn.*, vol. 92, no. 10, pp. 1–12, 2009.

[9] L.-C. Chen, H.-H. Jan, and C.-W. Huang, "Mensuration of concrete cracks using digitized close-range photographs," in *Proc. 22nd Asian Conf. Remote Sens.*, Singapore, 2001, pp. 1248–1253.

[10] L. Barazzetti and M. Scaioni, "Crack measurement: Development, testing and applications of an automatic image-based algorithm," *ISPRS J. Photogramm. Remote Sens.*, vol. 64, no. 3, pp. 285–296, 2009.

[11] T. C. Hutchinson and Z. Chen, "Improved image analysis for evaluating concrete damage," *J. Comput. Civil Eng.*, vol. 20, no. 3, pp. 210–216, May 2006.

[12] I. Abdel-Qader, O. Abudayyeh, and M. E. Kelly, "Analysis of edge-detection techniques for crack identification in bridges," *J. Comput. Civil Eng.*, vol. 17, no. 4, pp. 255–263, Oct. 2003.

[13] M. R. Halfawy and J. Hengmeechai, "Efficient algorithm for crack detection in sewer images from closed-circuit television inspections," *J. Infrastruct. Sys.*, vol. 20, no. 2, p. 04013014, Jun. 2014.

[14] A. Mohan and S. Poobal, "Crack detection using image processing: A critical review and analysis," *Alexandria Eng. J.*, to be published.

[15] M. Hao, C. Lu, G. Wang, and W. Wang, "An improved neuron segmentation model for crack detection—Image segmentation model," *Cybern. Inf. Technol.*, vol. 17, no. 2, pp. 119–133, 2017.

[16] Y. Hu, C.-X. Zhao, and H.-N. Wang, "Automatic pavement crack detection using texture and shape descriptors," *IETE Tech. Rev.*, vol. 27, no. 5, pp. 398–405, 2010.

[17] H.-N. Nguyen, T.-Y. Kam, and P.-Y. Cheng, "An automatic approach for accurate edge detection of concrete crack utilizing 2D geometric features of crack," *J. Signal Process Syst.*, vol. 77, no. 3, pp. 221–240, Dec. 2014.

[18] W. Wang and Y. Liang, "Rock fracture centerline extraction based on Hessian matrix and Steger algorithm," *KSII Trans. Internet Inf. Syst.*, vol. 9, no. 12, pp. 5073–5086, 2015.

[19] H. Ji and H. Hong, "Automatic detection of internal and surface cracks of metal products using two and three-dimensional crack enhancement filtering methods in industrial CT volume data," *J. KISS, Softw. Appl.*, Jan. 2014. Accessed: Mar. 8, 2018. [Online]. Available: http://www.dbpia.co.kr

[20] Y. Kim, "Development of crack recognition system for concrete structure using image processing method," *J. Korean Inst. Inf. Technol*, vol. 14, no. 10, pp. 163–168, 2016.

[21] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," in *Proc. Int. Conf. Med. Image Comput. Computl-Assisted Intervent*, 1998, pp. 130–137.

[22] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. CVPR*, 2010, pp. 2963–2970.

[23] M. Maguire, S. Dorafshan, and R. J. Thomas. (2018). *SDNET2018: A concrete crack image dataset for machine learning applications*. Accessed: Oct. 8, 2018. [Online]. Available: https://digitalcommons.usu.edu/all_datasets/48

[24] H. Cho and H.-J. Yoon, "ELCI: Edge based labeled crack image," *IEEE Dataport*, 2018. Accessed: Aug. 21, 2018, doi: 10.21227/5z39-7318.

HYUNWOO CHO was born in Chuncheon, South Korea, in 1984. He received the B.S. and M.S. degrees in mechatronics engineering from Kangwon National University in 2009 and 2012, respectively, and the Ph.D. degree in robotics and virtual engineering from the Korea University of Science and Technology, Daejeon, South Korea, in 2018. His research interests include image processing and artificial intelligence.

HYUK-JIN YOON received the M.S. and Ph.D. degrees in aerospace engineering from KAIST, Daejeon, South Korea, in 2002 and 2006, respectively. Since 2006, he has been a Principal Researcher with the Korea Railroad Research Institute, Uiwang, South Korea. Since 2011, he has been an Associate Professor with the Korea University of Science and Technology, Daejeon. His research interests include structural health monitoring applications with optical fiber sensor and vision camera, crack propagation, and distributed strain monitoring.

JU-YEONG JUNG received the B.S. and M.S. degrees from Hanbat National University, Daejeon, South Korea, in 2014 and 2016, respectively. In 2016, he joined the Korea Railroad Research Institute, Gyeonggi-do, South Korea, where he is currently a Post-Master. His research interests include microthruster, CFD, image processing, and artificial intelligence.