Civil Engineering Infrastructures Journal, 52(1): 11 – 22, June 2019
Print ISSN: 2322-2093; Online ISSN: 2423-6691
DOI: 10.22059/ceij.2019.246397.1444

Processing Digital Image for Measurement of Crack Dimensions in Concrete

Barkavi, T.[1*] and Natarajan, C.[2]

[1] Ph.D. Candidate, Research Scholar, Department of Civil Engineering, National Institute of Technology Tiruchirappalli, Tamil Nadu, India.
[2] Professor, Department of Civil Engineering, National Institute of Technology Tiruchirappalli, Tamil Nadu, India.

Received: 22 Nov. 2017; Revised: 05 Oct. 2018; Accepted: 28 Oct. 2018

ABSTRACT: The elements of the concrete structure are most frequently affected by cracking. Crack detection is essential to ensure safety and performance during its service life. Cracks do not have a regular shape, in order to achieve the exact dimensions of the crack; the general mathematical formulae are by no means applicable. The authors have proposed a new method which aims to measure the crack dimensions of the concrete by utilizing digital image processing technique. A new algorithm has been defined in MATLAB. The acquired data has been analyzed to obtain the most precise results. Here both the length and width of the crack are obtained from image processing by removing background noise for the accuracy of measurement. A semi-automatic methodology is adapted to measure the crack length and crack width. The applicability of the program is verified with the past literature works.

Keywords: Algorithm, Crack Length, Crack Width, Cracks, Digital Image Processing.

INTRODUCTION

Image processing is the process of extracting significant information from digitized images by transforming them into other images using various mathematical algorithms Okan Önal et al. (2008). As for automated damage detection, many methods have been created using image processing techniques such as wavelet transforms, edge detection, and/or region-based segmentation. Abdel-Qader et al. (2003) compared various edge detection algorithms and found the Haar Wavelet method is the most reliable among them, for the purpose of crack detection. However, the performance of edge detection algorithms on noisy image data is questionable, and the same is the case with morphological operation based methods. Kabir and Rivard (2007) used Haar's discrete wavelet transform to study the deterioration in concrete structures from the 2D image. Yamaguchi et al. (2010) used scalable local percolation-based image processing techniques, and they proved to be efficient and accurate even for large surface images. Prasanna et al. (2012) developed a histogram-based classification algorithm and used it along with Support Vector Machines to detect cracks on a concrete deck surface. The results of this algorithm on real bridge data highlighted the need for improving the accuracy. Nevertheless, training data from various locations on the bridge could be used

* Corresponding author E-mail: barkavicivil@gmail.com

to build the classifier and testing could be done on data from a different location of the similar structural composition. Similarly, Lattanzi and Miller (2014) developed an automatic clustering method for segmentation based on Canny and K-Means for achieving greater accuracy of crack detection under various environmental conditions at a higher speed. Li and Xuehui (2014) used a feature extraction methodology in image processing to identify segregation of the SCC mix from the photograph of the mix. Yu et al. (2007) developed a graph-based search method to extract crack properties for further assessment and used a ground-based robot for collecting images; however, this method needed the manual input of start and end points of crack. They also have used watershed segmentation algorithm to found the lower boundary of the mix and thresholding to find the upper boundary. They manually picked the boundary points and checked the vertical distance between them and have used it as the visual feature to identify segregation in the mix. The above-discussed works are focused on crack detection alone, but it is also vital in an inspection to understand the crack properties such as width and length because condition ratings are assigned based on such properties. This type of Condition monitoring aims to obtain the signs and indicators revealing the condition of the structure. This procedure is more cost-effective than predictive or regular maintenance or repair (Mohammadizadeh and Yasi, 2018).

SEMI-AUTOMATIC CRACK MEASUREMENT

In this research article, a new method of semi-automation has been proposed. This system doesn't have the provision of automatic image acquisition by external devices. Here the human intervention is needed to capture the image and feed it into the crack detecting system to compute the numerical information of crack images. So the parameters like calibration length and camera focal length are neglected while writing the algorithm. In order to compensate that, the user is instructed to select the first, end and traversing points along the crack length by using a mouse-controlled pointer. Even though it seems to be time-consuming and work-intensive, this approach will give an accurate result. Another advantage is that in case of an image with multiple/continuous cracks the user can retrieve the property of crack of interest alone.

MANUAL MEASUREMENT OF CRACK

The manual measurement is conducted by a technician with measuring tools such as strings, graduated scale or cracks comparator. The main problems about those measuring by comparing methods could be enumerated as low accuracy and traceability, subjectivity on reading and difficulties on recording data (Martins et al., 2013). Another instrument with better resolution applied is digital pachymeter, for which, the technician must have the knowledge to choose and insert the metallic blade in the crack opening. The conventional dimensional instruments have subjectivity on manual positioning during measurement. Another factor is the uncertainness associated with the handling of the instrument by a technician. These inconsistencies demand a higher number of repetitive measures to have more accurate values (Koch et al., 2015). With the purpose of reducing the technician workload and the generation of accurate results, the work aims to identify the crack dimensions on concrete.

DIGITAL IMAGE PROCESSING

Digital Image Processing is the application of various operators or filters on the image to eliminate or reduce noise, some to heal or improve the image, while others may be used to separate required information from the image. The processing methods available are from low-level processing up to high-level processing methods as illustrated in Figure 1.

This research undergoes intermediate level processing of images, i.e. extraction features from the image. The block diagram showing the processing methodology adopted in this research work is represented as a flowchart in Figure 2.

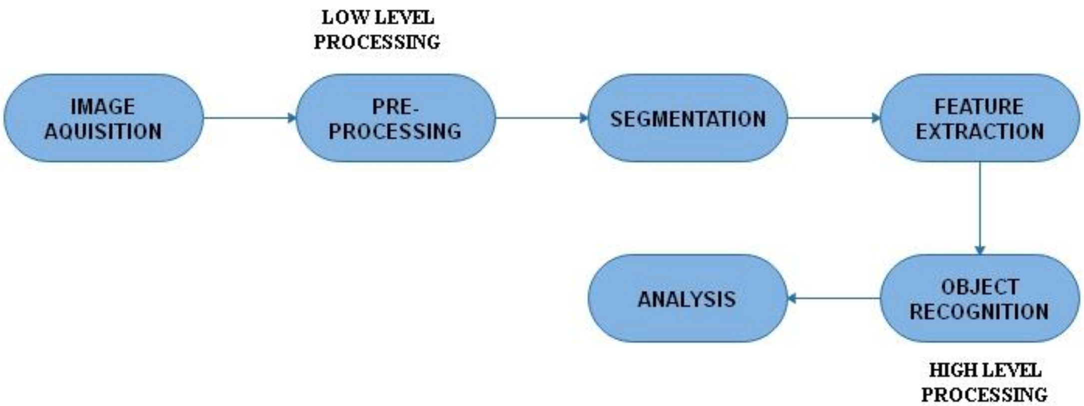


Fig. 1. Pictorial representation of low to high-level image processing methods

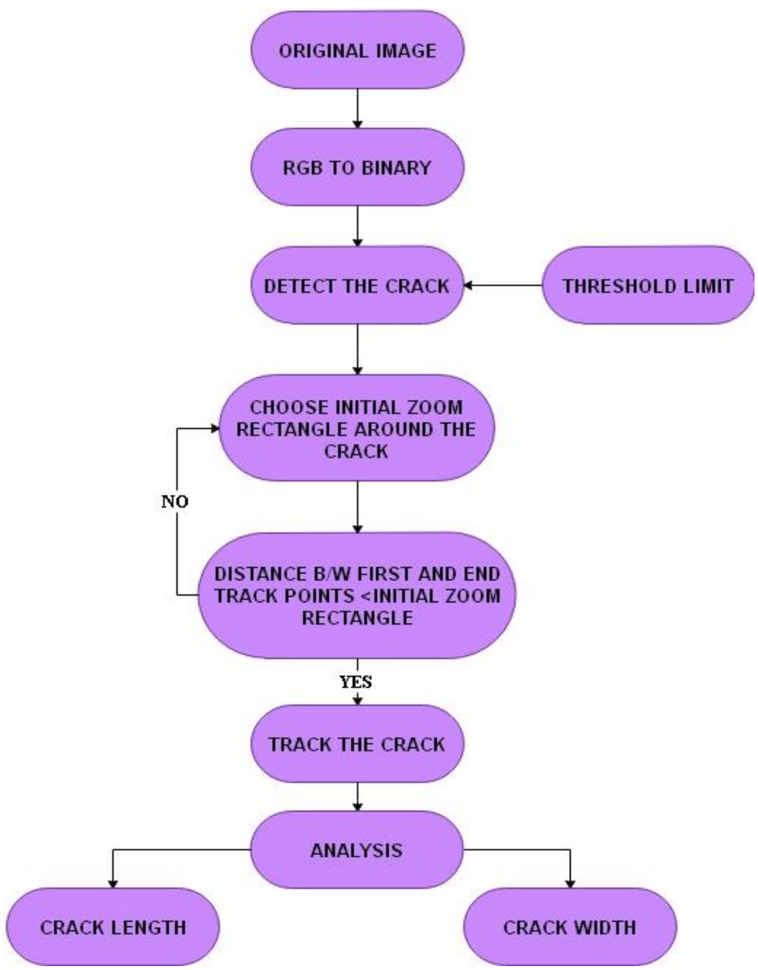


Fig. 2. Block diagram of the image processing methodology adopted in the research

DEFINITION OF FLOWCHART TERMS

Original Image: The original image of the cracked concrete surface in .jpeg format is uploaded. An RGB image is also referred to as True Color Image, which is stored in MATLAB as 24- bit image in which each color component occupies 8-bit each.

The threshold is the simplest segmentation method of converting a grey-scale image to a binary image using a value. Pixels brighter than the threshold value are white pixels in the resulting image; pixels darker than the threshold value are black pixels. After conversion, connected areas of white pixels are interpreted as single objects. The manual selection of the threshold value can be made by otsu, iterative, local enthropy, joint enthropy, relative enthropy- and renyis enthropy-method. When using a manual selection of the threshold value, it is possible to select a threshold value between 0 (black) and 1 (white). When using one of those methods a factor can be selected to modify the automatically determined value. Also, it is programmed to have an inbuilt binary factor of closing and thickening. Eq. (1) shows the general method of thresholding an image for binarization.

$$\text{If } I(x, y) > T(x, y), \text{ Then } b(x, y) = 1, \text{ Else } b(x, y) = 0 \quad (1)$$

where $b(x,y)$: is the binarized image and $I(x,y) \in [0,1]$: is the intensity of a pixel at location (x, y) of the image I (Firdousi and Parveen, 2014).

The otsu's local thresholding is represented in Eq. (2)

$$t = Arg \underset{0 \le t < L}{Max} \begin{Bmatrix} w_1(t)\mu_1^2(t) + \\ w_2(t)\mu_2^2(t) \end{Bmatrix} \quad (2)$$

The found threshold t is multiplied with the scale factor k(0 to 1) for getting new threshold t^*, see Eq. (3)

$$t^* = t * k \quad (3)$$

Select an initial zoom rectangle around object of interest. The user is advised to select an initial zoom rectangle around the crack of interest; so that the particular crack can be tracked to analyze its crack properties i.e. crack length and width.

The distance between first and end track points < initial zoom rectangle. The crack track runs with the condition that if the distance between a tracking point and the end point is greater than the length initial zoom rectangle. If the condition fails, then the program goes to the previous step. If the condition is satisfied the crack track will be continued.

Crack-Track: In this process the user will be advised to select manually the first, traversing and end points of crack whose length has to be measured. The endpoint has to be selected by pressing the Shift+end point. The algorithm for measuring the crack length and crack width is explained in the following section.

PSEUDO CODE OF THE PROPOSED ALGORITHM

1. Read the RGB original cracked concrete surface image: uiget file('*.*',OpenFile)
2. Convert the RGB image to grayscale. rgb2gray()
3. Choose a threshold value lying between 0-1, so that all branches of crack are visible with low/no noise in the background.
4. Binarize the image: im2bw(img,val)
5. The binarized image get stored as 'crackfind.jpg'
6. Read 'crackfind.jpg'
7. Check the size of the image [Sx, Sy] > 10
8. Select an initial zoom rectangle around the crack (object of interest) getrect()
9. Plot the selected object in a separate

window with X and Y axis showing the resolution of the selected region/image.

10. Mark control points along the crack length and the points are saved as a 2-dimensional vector – Xi = [X,X0] and Yi = [Y,Y0]
11. Use 'interparc' function to interpolate points between the control points marked to form an arc with piece-wise approximations.
12. To calculate the arc/crack length use 'arclength' function.
13. In a similar way, the crack width is measured for a particular area selected in the crack.

VALIDATION AND DISCUSSION

To check the real accuracy of the proposed algorithm developed using MATLAB, it has been validated with a crack image acquired from the concrete surface. For proper validation, the crack length is measured on the real crack by using string, and it is found as 83 mm. The original image without any scale change is fed into the MATLAB algorithm. For doing morphological corrections, the threshold limit is not standardized as it may vary for each image. Hence the user is instructed to select the best-suited threshold value, and fine changes can also be done. Figure 3 shows the screenshot of the program window.

After opening the image, it can be converted to a grayscale, i.e. binary image by selecting the best-suited threshold value. The program is validated with a crack image of known crack length and crack width. The best-suited threshold value for the image is selected by the user is shown in Figure 4.

Crack Length

In the new window, as shown in Figure 5, the user will be instructed to select the object/region of interest in order to focus on the crack pixels without changing the scale of the image. After that the user will be advised to select manually the first, traversing and end seed points of crack whose length has to be measured. The endpoint has to be selected by pressing the Shift+end point. Further, the crack length will be measured automatically. For the concrete surface crack image, the crack length found from the proposed algorithm is 82.15 mm.

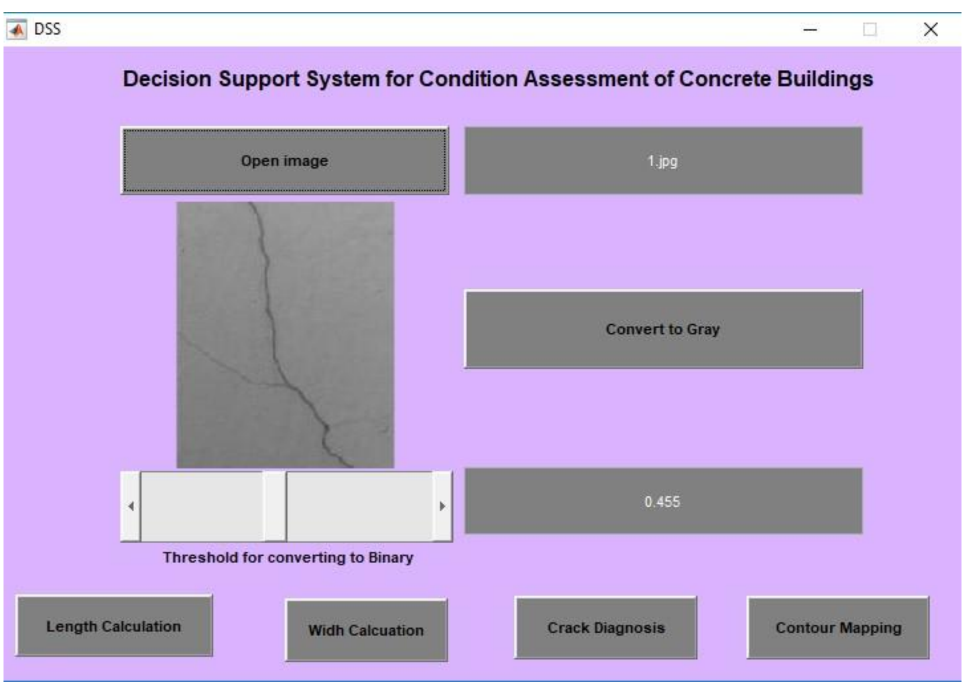


Fig. 3. Original images for crack property retrieval

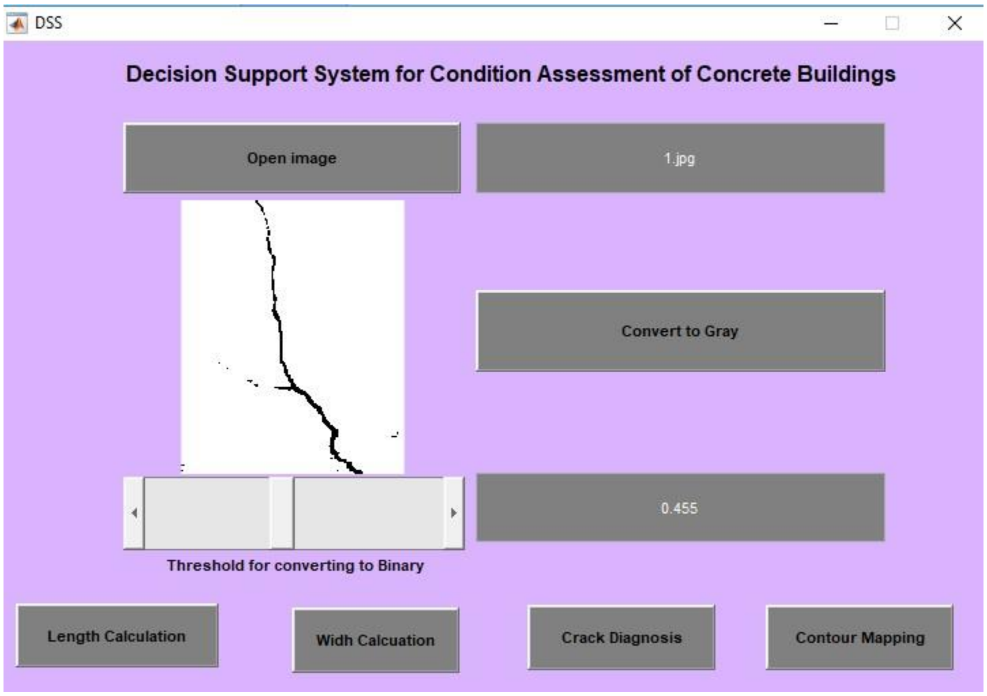


Fig. 4. Binary image of the crack image with user chosen threshold value 0.45

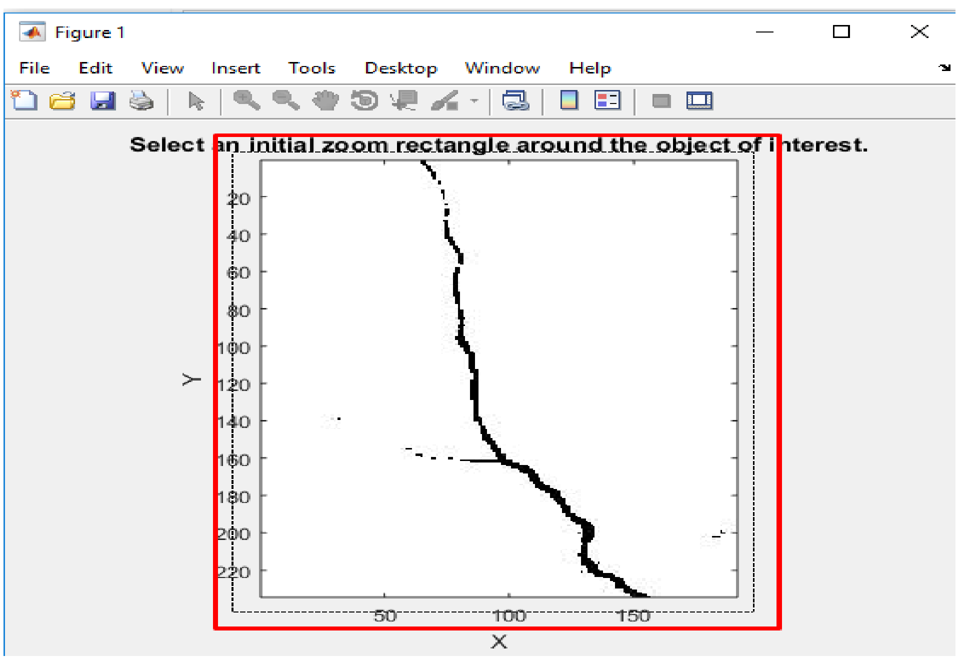


Fig. 5. Selecting the object of interest in the image

Tracking Process

The tracking process starts with the first seed point. The two conditions that stop tracking are: 1) if the distance between a tracking point and the end point is less than a profile span, 2) The number of tracking points is more than triple of the proper number of profiles in the selected region of interest. Table 1 shows the crack length investigation result of all four samples tested using the program. The table also includes the error percentage for the difference in natural length of crack and measured length. This is shown in Figure 6.

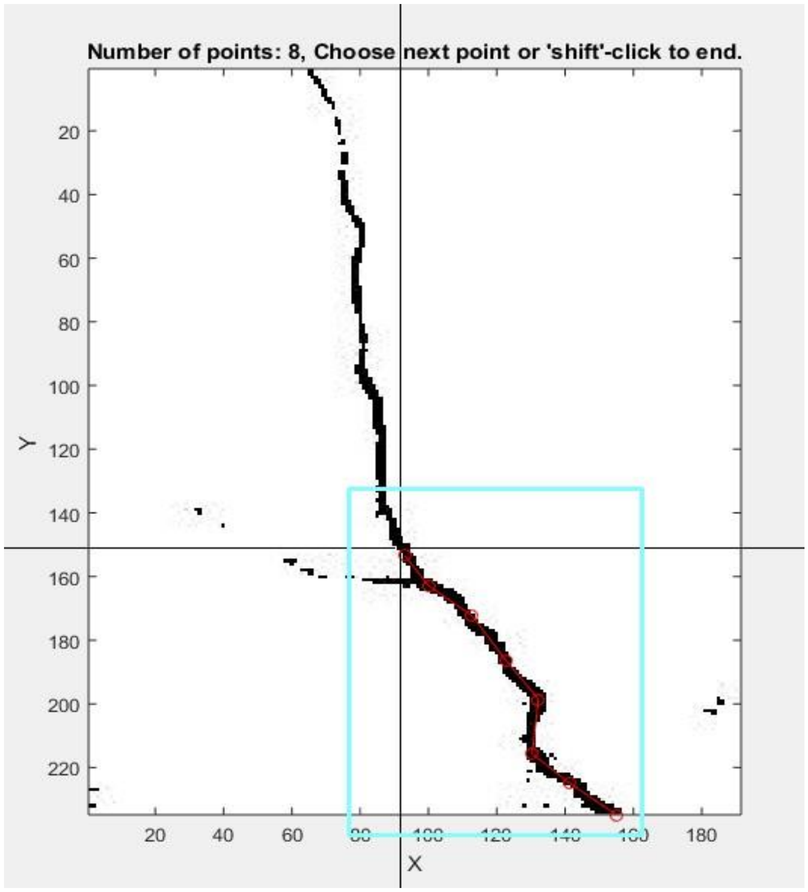
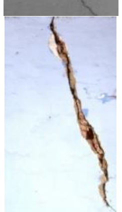
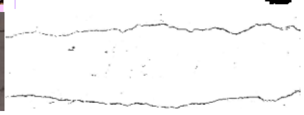


Fig. 6. Selecting the first, traversing and endpoint in the crack image

Table 1. Showing the properties of crack images

Image No.	Original image of crack	Binary image of crack	Manual measurement of crack length (mm)	Crack length provided by proposed algorithm (mm)	Error (%)
1			83	82.15	1.02
2			46.05	45.65	0.88
3			121.6	120.92	0.56
4 a, b			65 62	64.25 61.05	1.15 1.50
				Average Error	1.02

Crack Width

Crack width is measured for five random points along the crack, and the average width is proposed as the measured crack width. The region / object of interest along the crack for width detection has to be manually chosen by the user. Further, the start and end point of a particular chosen pixel have to be selected to know the crack width at the location. Figure 7 shows the selection of a particular area of interest along the line of crack. Figure 8 shows the zoomed pixel which its first and endpoint are chosen to know the crack width.

Among the five points of crack width, the average crack width is found to be 0.65 mm. Table 2 shows the average crack width of all four cracks studied.

Validation for Crack Width Detection by the Algorithm

The minimum detectable crack size is reported as the most critical parameter of the inspection method (Laefer et al., 2010). Hence the minimum width detectable by the algorithm has to be checked. The concrete surface crack with two parallel horizontal cracks is taken for the study. In the first crack nine points are chosen for crack width measurement, and in second crack four points are studied. In the first crack among nine points, three points namely 2, 4 and 6 are not detected by the algorithm. In the second crack

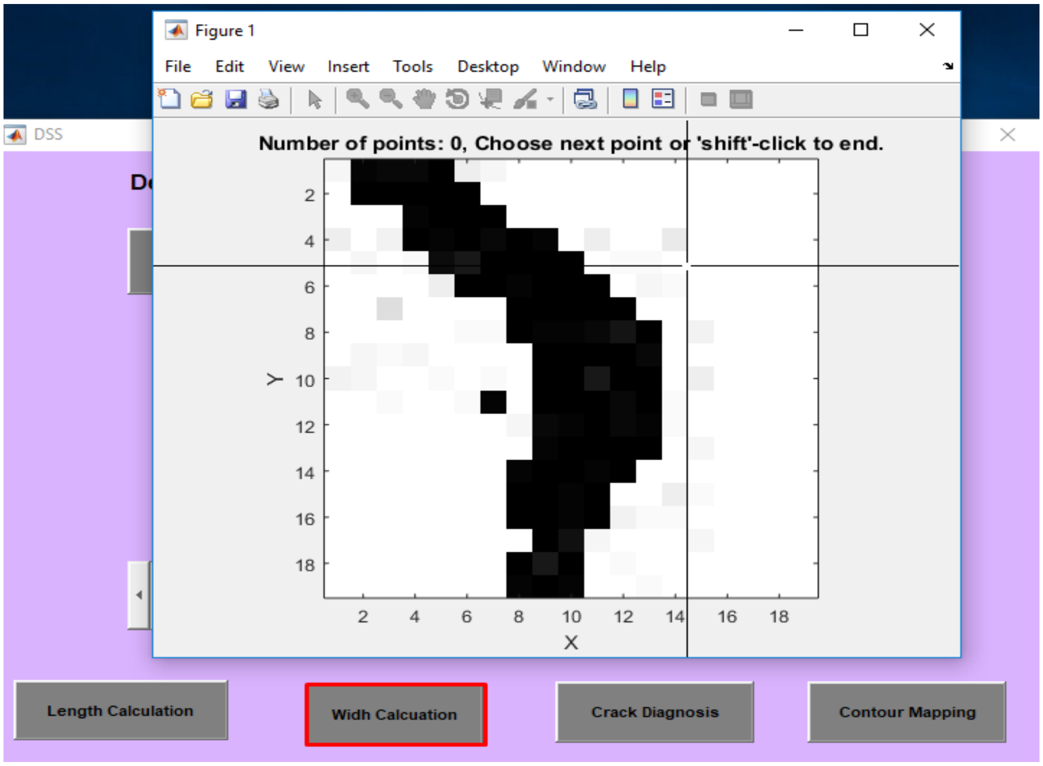
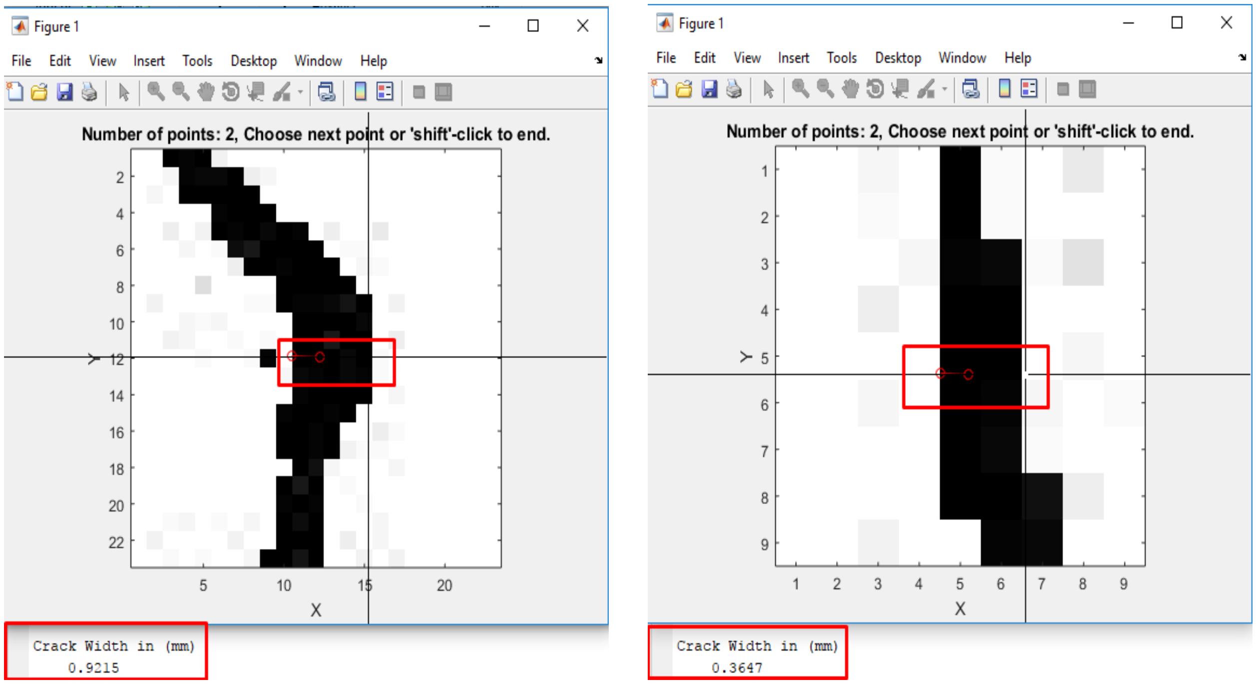


Fig. 7. Selection of the area of interest along the crack length

Table 2. Crack width inspection

Image No.	Average crack width (mm) (Manual)	Average crack width proposed algorithm (mm)	Errors (%)
1	0.65	0.64	1.53
2	1.2	1.12	6.7
3	1.75	1.68	4
4a	1.24	1.24	0
4b	1.3	1.29	0.77
	Average error		2.6



(a) (b)

Fig. 8. Zoomed crack pixel which its first and last points are measured to know the crack width, 0.92 mm (maximum) and 0.365 (minimum)

among 4 points, one crack point, i.e. 2 was not detected by the algorithm. The respective crack width values for each point on both the crack lines are tabulated in Table 3. The concrete surface crack studied for crack width validation is illustrated in Figure 9.

Form Table 3, it is understood that the proposed algorithm can measure cracks thicker than 0.15 mm. A naked eye can detect cracks whose width is lesser than 0.1 mm, but it is unable to detect and measure with proposed algorithm. According to ACI 224R-90, tolerable crack width is designed with regard to the exposure condition as shown in Table 4 below.

From Table 4, we can infer that the proposed approach can be applied with confidence for concrete buildings in Tropical countries. This algorithm is not recommended for offshore buildings, water retaining structures and buildings at cold weather.

Table 3. Crack width measurement for each crack points

Crack No.	Location	Manual measurement (mm)	Proposed algorithm (mm)
	1	1.2	1.1
	2	0.13	NA
	3	1.2	1.18
	4	0.11	NA
	5	1.2	1.21
I	6	0.1	NA
	7	1.3	1.27
	8	1.25	1.22
	9	1.3	1.26
	1	1.3	1.3
	2	0.11	NA
II	3	1.3	1.28
	4	1.3	1.3

Table 4. Tolerable crack width for several exposure conditions as per ACI 224R-90

Exposure	Tolerable crack width (mm)
Dry air protective membrane	< 0.40
Humidity, moist air, soil	< 0.30
Deicing chemicals	< 0.18
Seawater and seawater spray; Wetting and drying	< 0.15
Water retaining structures	< 0.10

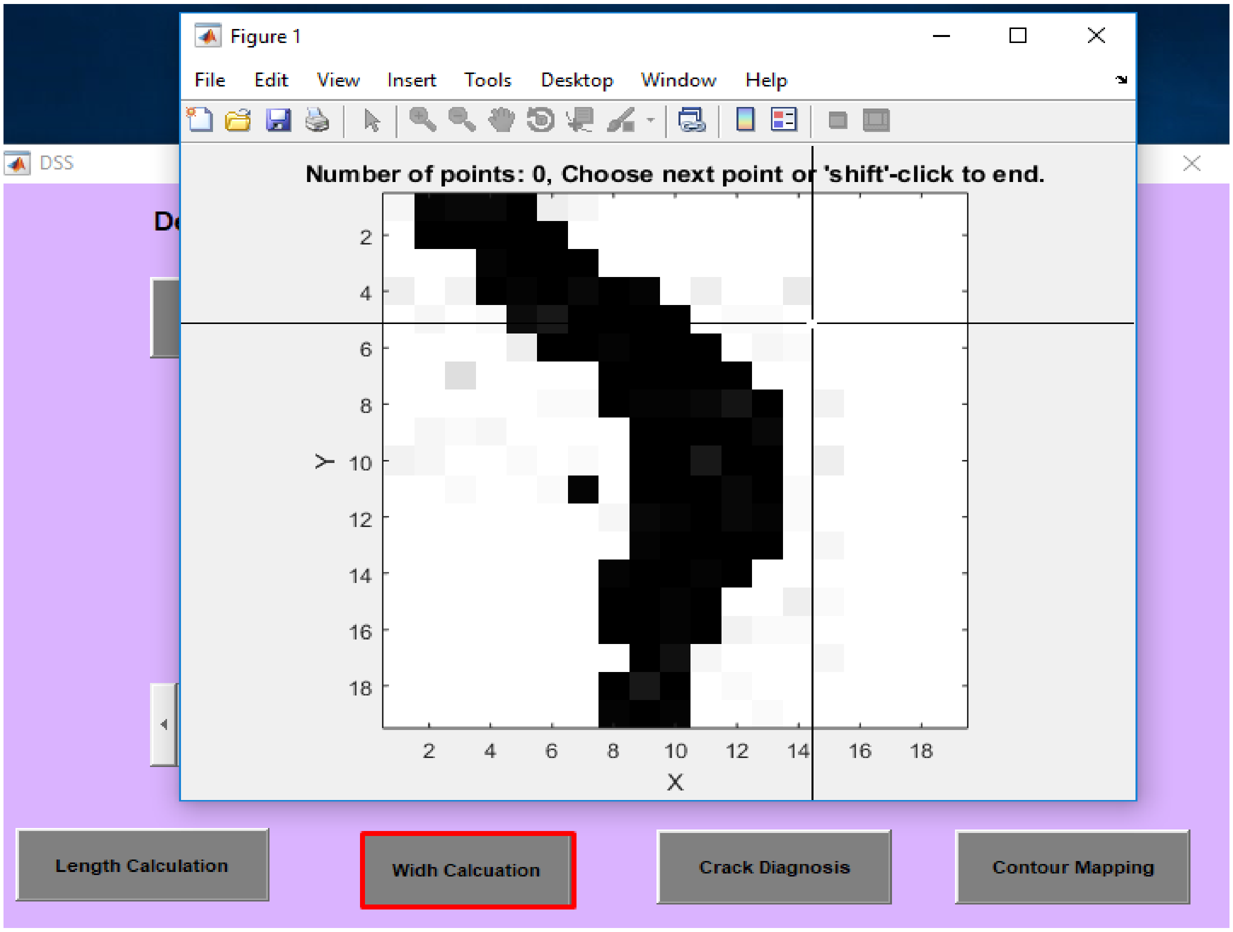
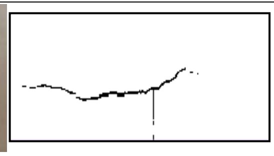


Fig. 9. Validation for crack detection and width measurement

Validation for Crack Length Detection by the Algorithm

For validating the crack length measurement capacity of the proposed algorithm, the working of the program is tested with the data available in the literature work of Khalili and Vahidnia (2014). The main idea of crack measurement using MATLAB has been inspired from their work. However, the objective is different. The crack sample they have studied has an original crack length of 15.1 cm. They have used Image Processing Toolbox of MATLAB for crack measurement. They have used a fixed threshold value of '127' for binarization of the image since their objective was to calculate the crack length of that particular image. After binarization, the morphological operations like Erode, Opening, Filtering and Pruning are done to get the center line of the crack (i.e. skeletonizing of the binary image) of 1-pixel width. The crack length is measured with the command 'regionprop', which finds the area of the black pixels (crack). The details of verification of crack length are tabulated in Table 5.

Table 5. Crack length measurement of an image taken from literature work of Khali and Vahidinia (2014)

Original image of crack	**Binary image of crack**	**Manual measurement of crack length (cm)**	**Crack length given by Khalili and Vahidnia (2014) (cm)**	**Crack length provided by the proposed algorithm (cm)**
		15.1	13.8	14.5
	Error (%)		8.6	3.9

The proposed algorithm is working well then the algorithm developed by Khalili and Vahidinia (2014).

UNCERTAINTIES HANDLED BY THE ALGORITHM

The algorithm will show the uncertainties in the form of warning message so that the user can rectify it. They are discussed in Table 6.

Table 6. Uncertainties handled by the algorithm

Uncertainties	**Warning messages**
If the image size [Sy,Sx] is lesser than 10	'Image is too small – try with another image.'
If the user marked the path away from crack pixels	'Point lies outside the crack pixel.'
If the user marked the path outside the image	'Point lies outside the image.'

CONCLUSIONS

In this article, for measuring crack properties viz. crack length and crack width, the algorithm developed using MATLAB was presented. This algorithm after applying on the image of crack was successfully adopted. The results of the practical experiment show that a sample surface concrete cracks with a 95 mm crack length and 0.45 mm average crack width were recognized and measured properly and by means of provided algorithm the accuracy of the calculation can be highly improvised in order to decrease the error of measurement to a minimum. The widths of cracks change from place to place; hence a non-contact algorithm won't give an accurate result. Thus the research has proposed a methodology of seed point fixing along the crack line. In addition, the algorithm has analyzed totally four crack images for measurement of crack length and width. The crack width, lesser than 0.1 mm is visible to naked eye, but it was unable to detect and measure with proposed algorithm. The precision of the proposed algorithm for the irregular shape of cracks with a lot of inequality and variable thickness is sufficiently satisfactory. The image with two cracks is also handled satisfactorily.

REFERENCES

Abdel-Qader, I., Abudayyeh, O. and Kelly, M.E. (2003). "Analysis of edge-detection techniques for crack identification in bridges", *Journal of Computing in Civil Engineering*, 17(4), 255-263.

ACI 224R-90. (1990). "Control of Cracking in Concrete Structures", America Concrete Institute: Farmington Hills, MI, USA.

Firdousi, R. and Parveen, S. (2014). "Local thresholding techniques in image binarization", *International Journal of Engineering and Computer Science*, 3(3), 4062-4065.

Kabir, S. and Rivard, P. (2007). "Damage classification of concrete structures based on grey level co-occurrence matrix using Haar\'s discrete wavelet transform", *Computers and Concrete*, 4(3), 243-257.

Khalili, K. and Vahidnia, M. (2014). "Improving the accuracy of crack length measurement using machine vision", *8th International Conference Inter-Disciplinarily in Engineering., Procedia Technology*, 19, 48-55.

Koch, C., Georgieva, K., Kasireddy, V., Akinci, B. and Fieguth, P. (2015). "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure", *Advanced Engineering Informatics*, 29(2), 196-210.

Laefer, D.F., Gannon, J. and Deely, E. (2010). "Reliability of crack detection methods for baseline condition assessments", *Journal of Infrastructure Systems*, 16(2), 129-137.

Lattanzi, D. and Miller, G.R. (2014). "Robust automated concrete damage detection algorithms for field applications", *Journal of Computing in Civil Engineering*, 28(2), 253-262.

Li, S. and An, X. (2014). "Method for estimating workability of self-compacting concrete using mixing process images", *Computers and Concrete*, 13(6), 781-798.

Martins, A.P., Pizolato Junior, J.C. and Belini, V.L. (2013). "Image-based method for monitoring of crack opening on masonry and concrete using mobile platform", *Revista IBRACON de Estruturas e Materiais*, 6(3), 414-435.

Mohammadizadeh, M. (2018). "Identification of Structural Defects Using Computer Algorithms", *Civil Engineering Infrastructures Journal*, 51(1), 55-86.

Önal, O., Özden, G. and Felekoglu, B. (2008). "A methodology for spatial distribution of grain and voids in self-compacting concrete using digital image processing methods", *Computers and Concrete*, 5(1), 61-74.

Prasanna, P., Dana, K., Gucunski, N. and Basily, B. (2012). "Computer-vision based crack detection and analysis, in SPIE smart structures and materials + nondestructive evaluation and health monitoring", In Sensors and Smart Structures Technologies for Civil, Mechanical and Aerospace Systems, Vol. 8345, pp. 834542, International Society for Optics and Photonics.

Yamaguchi, T. and Hashimoto, S. (2010). "Fast crack detection method for large-size concrete surface images using percolation-based image processing", *Machine Vision and Applications*, 21(5), 797-809.

Yu, S.N., Jang, J.H. and Han, C.S. (2007). "Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel", *Automation in Construction*, 16(3), 255-261.