

# Journal of Electronic Imaging

JElectronicImaging.org

## Detection of pavement cracks using tiled fuzzy Hough transform

Senthan Mathavan  
Kanapathippillai Vaheesan  
Akash Kumar  
Chanjief Chandrakumar  
Khurram Kamal  
Mujib Rahman  
Martyn Stonecliffe-Jones

**SPIE**•



Senthan Mathavan, Kanapathippillai Vaheesan, Akash Kumar, Chanjief Chandrakumar, Khurram Kamal, Mujib Rahman, Martyn Stonecliffe-Jones, "Detection of pavement cracks using tiled fuzzy Hough transform," *J. Electron. Imaging* **26**(5), 053008 (2017), doi: 10.1117/1.JEI.26.5.053008.

# Detection of pavement cracks using tiled fuzzy Hough transform

Senthan Mathavan,<sup>a,b,\*</sup> Kanapathippillai Vaheesan,<sup>c</sup> Akash Kumar,<sup>d</sup> Chanjief Chandrakumar,<sup>e</sup> Khurram Kamal,<sup>f</sup> Mujib Rahman,<sup>g</sup> and Martyn Stonecliffe-Jones<sup>h</sup>

<sup>a</sup>Nottingham Trent University, School of Architecture, Design and the Built Environment, Burton Street, Nottingham, United Kingdom

<sup>b</sup>Loughborough University, EPSRC National Centre for Innovative Manufacturing in Intelligent Automation, School of Mechanical and Manufacturing Engineering, Ashby Road, Loughborough, United Kingdom

<sup>c</sup>University of New South Wales, School of Electrical Engineering and Telecommunications, Sydney, New South Wales, Australia

<sup>d</sup>Iscorpion, GS Store, DHA, Karachi, Pakistan

<sup>e</sup>Massey University, School of Engineering and Advanced Technology, Palmerston North, New Zealand

<sup>f</sup>National University of Sciences and Technology, College of Electrical and Mechanical Engineering, Department of Mechatronics Engineering, Rawalpindi, Pakistan

<sup>g</sup>Brunel University, Department of Civil Engineering, Kingston Lane, Uxbridge, Middlesex, United Kingdom

<sup>h</sup>Dynatest International A/S, Søborg, Denmark

**Abstract.** Surface cracks can be the bellwether of the failure of a road. Hence, crack detection is indispensable for the condition monitoring and quality control of road surfaces. Pavement images have high levels of intensity variation and texture content; hence, the crack detection is generally difficult. Moreover, shallow cracks are very low contrast, making their detection difficult. Therefore, studies on pavement crack detection are active even after years of research. The fuzzy Hough transform is employed, for the first time, to detect cracks from pavement images. A careful consideration is given to the fact that cracks consist of near straight segments embedded in a surface of considerable texture. In this regard, the fuzzy part of the algorithm tackles the segments that are not perfectly straight. Moreover, tiled detection helps reduce the contribution of texture and noise pixels to the accumulator array. The proposed algorithm is compared against a state-of-the-art algorithm for a number of crack datasets, demonstrating its strengths. Precision and recall values of more than 75% are obtained, on different image sets of varying textures and other effects, captured by industrial pavement imagers. The paper also recommends numerical values for parameters used in the proposed method. © 2017 SPIE and IS&T [DOI: [10.1117/1.JEI.26.5.053008](https://doi.org/10.1117/1.JEI.26.5.053008)]

Keywords: condition monitoring; surface inspection; defect analysis; crack; Hough transform; fuzzy logic; pavement.

Paper 170143 received Feb. 23, 2017; accepted for publication Aug. 15, 2017; published online Sep. 9, 2017.

## 1 Introduction

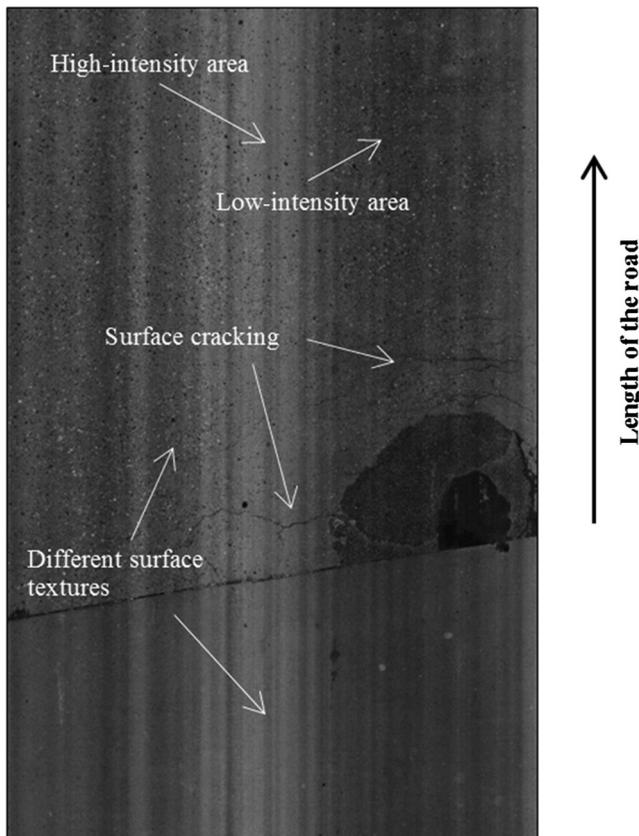
The planet had about 64 million kilometers of roads in 2013,<sup>1</sup> and about 65% of these were paved.<sup>2</sup> Maintaining the paved roads, of around 40 million kilometers, is a herculean task for road authorities all over the world. Condition monitoring is the key for road maintenance, and frequent health monitoring surveys are conducted to gather information about the state of pavement. Maintenance decisions are taken based on the evaluated severity, of paved roads, that are obtained via the surveys. The inspection surveys have been performed manually for a great deal of the 20th century. However, manual surveys are not feasible anymore due to the sheer volume of roads that need to be inspected. Furthermore, the difficulty is manifold these days, given that the capacity of every road in a network is used to the full. This utilization leads to an ever faster degradation, necessitating more frequent inspections. This higher inspection frequency is impossible to be tackled manually, and the picture is further complicated by the shortage of skilled inspectors. In addition, manual highway surveys are also prone to human subjectivity. On the other hand, an automated survey, when designed and validated aptly, can

be fast, accurate, and precise. In the coming decades, road inspections will be automated to a very high degree.

There are numerous types of pavement defects: potholes, raveling, spalling, and cracking, to name a few. For an exhaustive list of pavement distresses, please refer to the US Federal Highway Administration Distress Identification Manual.<sup>3</sup> Among the defects, potholes and cracks are the most common. Cracks are one of the earliest to occur on a newly paved road surface and are known to lead to potholes.<sup>4</sup> This fact makes the early detection of cracking in roads very critical for their maintenance and upkeep. Roads are mainly paved with either asphalt or concrete. Both types of paved surfaces undergo structural degradation during their usage. This degradation frequently leads to cracking. The cracks cause a loss in the load-spreading and water-resistant capacity of the roads. These detrimental effects then speed up the deterioration process of a pavement surface. If these cracks are untreated, the consequences become more severe and these cracks transform into potholes, deform the road, and sometimes even produce differential settlement of the road, making the ride bumpy.<sup>5</sup>

Crack detection in pavement images, such as the one shown in Fig. 1, has been an active area of research for almost the last two decades. The pavement image in

\*Address all correspondence to: Senthan Mathavan, E-mail: [s.mathavan@ieee.org](mailto:s.mathavan@ieee.org)



**Fig. 1** A pavement image obtained from a road monitoring equipment consisting of two different surface types (Courtesy: Dynatest UK, Ltd.).

Fig. 1 is obtained from the multifunctional vehicle (MFV), a mobile road inspection platform available from Dynatest.<sup>6</sup> The pavement image in Fig. 1 is a good example highlighting the difficulties involved in crack detection. In this figure, there are two different underlying surfaces with dissimilar textures, as the annotation shows. Additionally, the intensity of the image is varying across the width of the road, mainly due to tire wear and due to this wheel track areas tend to accumulate dark particulate rubber. Often, parts of a crack are found to be unconnected segments, as also see in Fig. 1.

Moreover, cracks are often very faint and thin in pavement images and often are embedded in a heavily textured and nonuniformly illuminated background. The problem is severe when the texture contrast is almost equal to that of usually dark cracks, making the latter's detection very difficult. Complicating the picture further, cracks are randomized structures, not fitting any analytical descriptions, hence detecting them is challenging. Due to these reasons, highway authorities exercise a high level of caution in adopting any of the proposed crack detection methodologies, and as such the research is very active to date.<sup>7,8</sup>

This paper presents a methodology for crack detection by the use of the fuzzy Hough transform. The work extends the earlier work of Vaheesan et al.<sup>9</sup> by a considerable amount. In this regard, the line detection capability of the Hough transform is utilized. Additional flexibility is provided by the fuzzified version of the Hough transform, which allows for nonstraight line sections to be picked out still. This aspect is essential for detection, because crack segments are

never perfectly straight. To reduce the influence of noise pixels, the Hough transform is performed over smaller image tiles, thereby increasing the robustness. As many crack detection algorithms concentrate on a specific type of image, a number of different image sets, with varying levels of surface texture, crack sizes, and crack contrasts, are used in this work. Some of the considered image sets are captured under controlled and uncontrolled conditions.

The paper is arranged as follows. Section 2 reviews the relevant literature. Section 3 presents the theory required for the work and the methodology adopted. In Sec. 4, the data that are used for the analysis are introduced. Results are presented in Sec. 4. Section 5 covers future developments. The paper ends with conclusions.

## 2 Related Works

The history of the automated detection of road cracks goes back as early as the 1990s.<sup>10</sup> In the 25 years since, there have been numerous publications outlining a variety of methods for detecting pavement cracks. Hence, it is difficult to provide an exhaustive overview of the methods that have been used hitherto. Here, only the important developments in this research area will be outlined together with literature relevant to this work. For a better overview of different methods used for road crack detection, refer to the review paper of Chambon and Moliard<sup>11</sup> or Koch et al.<sup>12</sup> Chambon and Moliard<sup>11</sup> classify the different detection methodologies, proposed so far, into the following categories: histogram-based, morphological, machine learning-aided, and filtering-based models. Here, a brief review about some of these techniques is provided followed by a detailed review of methods that have been used to detect cracks using the identification of line segments.

Crack pixels, as shown in Fig. 1, are darker than their background. Hence, a local histogram is expected to be bimodal, with crack pixels belonging to a mode. Any technique to demarcate the two modes effectively leads to crack segmentation. In this regard, Teomete et al.<sup>13</sup> use an extended technique, based on projected histograms in the vertical and horizontal directions, of  $4 \times 4$  pixel areas for crack detection in pavement images. Metrics obtained from the two histograms are used to calculate a shape measure that not only detects crack segments but also characterizes their directions. However, the test results are provided for images with a consistent, continuous crack width of 3 to 4 pixels only, and this continuity and consistency in width are not particularly challenging for many crack detection algorithms. The shown images<sup>3</sup> have good contrast crack segments as well, once again presenting the algorithm with the best-case scenario.

Based on the fact that cracks are very thin across their width, and that there can be a number of segments that are separated by small gaps (i.e., presented as segments), morphology is often employed for crack detection. Adhikari et al.<sup>14</sup> apply different morphological operators, such as dilation to close the gaps between different crack parts, for overall crack detection. The used images are devoid of background texture, removing a challenging aspect of real-world pavement images. In the presence of comparatively strong surface texture, which cannot be removed by preprocessing alone, the morphological operators are also bound to work on the texture, and this will lower the detection accuracy. In an example employing a learning-based

approach, Oliveira and Correia<sup>15</sup> use an unsupervised learning scheme to identify crack tiles from images of flexible pavements.

Out of these different groups of methods, as outlined in Chambon and Moliard,<sup>11</sup> the most effective method that appears to handle some serious amount of background surface texture uses a hybrid approach to combine histograms and morphology.<sup>16</sup> In this work, images are split into  $75 \times 75$  pixel tiles. The window, based on its mean intensity, is normalized. Then morphological erosion and dilation are applied successively to reduce any intensity variances. The morphologically operated images are then thresholded, dynamically, using a procedure based on Otsu's method. The obtained binary image is divided, once more, into  $75 \times 75$  pixel nonoverlapping tiles, and for each tile, entropy is calculated. Using the histogram of entropy for each tile in the image and using a thresholding procedure based on Otsu's method, each tile is identified as containing a crack segment or not. A point to note is that their images neither involve more than one type of background surface texture within a single image, as shown in Fig. 1, nor any lighting intensity changes across any given images.<sup>16</sup>

Cracks, being a byproduct of material fracture, are near straight-line segments, of various lengths, connected in a pseudorandom manner.<sup>17</sup> Hence, many line detection strategies have been experimented for crack detection. In this regard, Zalama et al.<sup>18</sup> use features generated by Gabor filters to detect cracks from images with considerable intensity variations and texture. The images are obtained from a custom-made vehicle for pavements imaging, and they outline a procedure to choose the filter parameters. In a similar work, Salman et al.<sup>19</sup> use a Gabor filter bank to segment cracks from images obtained from the laser road imaging system (LRIS) pavement imager. In another work, circular Radon transform is used for orientation analysis of cracks.<sup>20</sup> Two-dimensional (2-D) matched filters are employed to detect line segments representing cracks.<sup>21</sup> In this work, by a comparison with standard edge detection techniques, a claim of robustness is made on the use of predefined filter masks derived from the matched filter algorithm.

In a later work, Amhaz et al.<sup>22</sup> have applied a minimal path algorithm for the localization of pavement cracks. It is a cost minimization procedure. In this regard, the path is defined as a set of contiguous pixels, and the cost function is defined as the sum of the intensities of pixels in a path. Here, the idea is that crack pixels are usually dark in color; hence, the cost-minimized path between any two points in an image will run through crack pixels. The results presented in the paper, obtained on actual and synthetic crack images, definitely show good promise, especially for low-contrast cracks. However, one major drawback of the optimization procedure appears to be in the heavy times involved for the processing, which run in the order of 400 to 1000 s. In another recent research, adopting the current trend, a deep convolutional neural network has been employed by Zhang et al.<sup>23</sup> for crack detection. A lot of color images obtained by a smartphone have been used to train a deep neural network in a supervised manner. The method appears to perform well under changing illumination conditions, as there was no lighting control with a smartphone, for highly textured pavement images, with both precision and recall staying above 85%. However, low-contrast cracking has

not been tackled. In addition, no time performance of the algorithms is given, provided that deep neural networks can be computationally expensive. In another work, crack detection has been performed with a coarse-to-fine approach,<sup>24</sup> where pixels are initially grouped into clusters based on the intensities, followed by performing adaptive image segmentation. The so-called "region of belief" takes into its account some geometric properties of detected regions, such as its length and area properties, to determine the best crack hypotheses. Despite a great deal of challenges present in the analyzed images, such as heavy background texture and low crack contrast, the results are very promising with detection accuracy figures as high as 95%. Processing times are not provided, but briefly analyzing the methods and algorithms used, the times cannot be really high, making this a very promising method.

There have been a number of papers advocating the use of the Hough transform for detecting cracks. Sohn et al.<sup>25</sup> use a modified iterative Hough transform (MIHT) for image registration to track cracks, in concrete pavement, between successive frames of a video sequence. However, the actual crack detection itself, in the textureless images, is performed by a combination of techniques, such as image enhancement, noise removal, histogram thresholding, and thinning, and not by MIHT.<sup>25</sup> In another work, the genetic-based inverse voting Hough transform has been applied to detect cracks from the ultrasonic B-scan images of subsurface cracks.<sup>26</sup> However, these cracks appear as parabola in ultrasonic images, thus the main target of this work is parabola detection, and hence is not much related to the current research. A similar work uses the randomized Hough transform to look for arc-like patterns, representing cracks, in ultrasonic images.<sup>27</sup> Gavilan et al.<sup>28</sup> in their work on crack detection in pavement images, use the Hough transform to detect straight-lines, such as joints. Using this scheme, they reduce the false positives (FPs) in their detection. However, they do not use the Hough transform to directly detect cracks as such. In a similar manner, Nguyen et al.<sup>29</sup> use probabilistic Hough transform to detect lane markings in their crack detection algorithm; hence, the application of the Hough transform is not crack detection directly. Although Vivekanandreddy et al.<sup>30</sup> have considered the Hough transform for crack detection, their application is the direct use of the basic transform, and hence they could detect only straight line crack segments, with the detection accuracy on the very low side. Rababah<sup>31</sup> considers tiled Hough transform for crack detection, but only to reduce crack images to a representative form, from which they construct certain features and utilize these in conjunction with other features. All the generated features are used with a number of classifier types. Due to the composite nature of the features, the detection possibilities with the Hough transform-based detection only are not explored in this study.

In summary, the literature review shows that the Hough transform for straight lines has not been comprehensively studied for crack detection yet. In addition, this work exploits the capabilities of the Hough transform by combining two different extensions of the transform in an innovative manner. While the fuzzy Hough transform has been used to detect near-straight crack segments, the effects of surface texture and noise on the detection are reduced by the tiled version of the Hough transform.

### 3 Proposed Methodology

The Hough transform is one of the methods used to find shapes from a set of points in a given space.<sup>32</sup> Hence, this is frequently used to find arbitrary shapes in an image. The detection is performed by a voting procedure. A binary image (the Hough concept is also applicable to grayscale images) will be the result of some processing operation, such as thresholding, where the foreground and the background will be of contrasting values in the original image. Then, the aforesaid voting procedure will be applied to the binary image leading to the detection, or nondetection, of an object of specific shape. In this work, edge detection is used to find candidate pixels for cracks. The resulting binary images from the edge detection are then subject to the Hough procedure to detect cracks differentiating them from other artifacts, such as texture and noise.

#### 3.1 Edge Detection Using Second Derivatives

Grayscale images of pavements are to be used in this research. If raw images are in color, they must be converted to grayscale to start with. The Hough transform and its variants, as explained later in this section, are proposed to be applied on binary images that consist of crack segments and any other artifacts, such as texture signatures. As the algorithm is expected to handle images of varying intensity, thresholding for usually darker than background cracks is not preferred. Edge detection is chosen to detect potential crack segments in an image. Especially, the second derivative in the gradient direction (SDGD) filter is found to be very effective. The mathematical basis of the filter is as follows. The 2-D Gaussian function is

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (1)$$

where  $\sigma$  is the standard deviation. The first derivatives of the Gaussian, in the  $x$ - and  $y$ -directions, respectively, are

$$\begin{aligned} dG_x(x, y) &= -\frac{x}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \\ dG_y(x, y) &= -\frac{y}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}. \end{aligned} \quad (2)$$

These continuous filters are implemented in the discrete form as square matrices. The second derivatives are defined in the following manner:

$$\begin{aligned} dG_{xx} &= dG_x \otimes dG_x \\ dG_{yy} &= dG_y \otimes dG_y \\ dG_{xy} &= dG_{yx} = dG_x \otimes dG_y, \end{aligned} \quad (3)$$

where  $\otimes$  stands for the convolution operation of the discrete matrices. In Eq. (3), the pixel indices  $x$  and  $y$  are dropped for the sake of simplicity. The image,  $I(x, y)$ , is then convolved with the first and second derivatives. To define the notation, any convolved image  $I_{i,j}$  is obtained from

$$\begin{aligned} I_i &= dG_i \otimes I \\ I_{ij} &= dG_{ij} \otimes I. \end{aligned} \quad (4)$$

Once the first and second derivatives of the image are found, the SDGD operation is defined as<sup>33</sup>

$$\text{SDGD} = \{[I_{xx}(I_x)^2 + 2I_{xy}I_xI_y + I_{yy}(I_y)^2]/[(I_x)^2 + (I_y)^2]\}. \quad (5)$$

The final output (i.e., binary image) is obtained by thresholding the SDGD, in Eq. (5), with a suitable value.

#### 3.2 Hough Transform for Straight Lines

The Hough transform uses the fact that through any given pixel an infinite number of lines can pass to detect straight lines in images. So, there is a chance that any of these lines can be common between a given number of pixels, forming the basis for line detection. Referring to Fig. 2, for an arbitrary pixel with coordinates  $(x_0, y_0)$  in the image plane  $x$ - $y$ , a line that goes through it is shown in green color. The equation of any straight line, which is generally written in terms of the gradient of the line with the  $x$ -axis and its intercept on the  $y$ -axis, can also be written as a function of the  $(r, \theta)$  plane as

$$y_0 = \left(\frac{-\cos \theta}{\sin \theta}\right)x_0 + \left(\frac{r}{\sin \theta}\right). \quad (6)$$

Rearranging the equation, it can be shown that if  $\theta \in [0, 2\pi)$

$$y_0 = x_0 \cos \theta + y_0 \sin \theta, \quad (7)$$

where  $r \in [0, D]$  and  $D$  is the diagonal length of the image in number of pixels.

Equation (7) represents all the lines that go through a given point,  $(x_0, y_0)$ , in the image. Moreover, it can be noted that the expression for  $r$  in Eq. (7) represents a sinusoid in the  $(r, \theta)$  plane. So, for every pixel detected in the image plane  $x$ - $y$ , there will be a sinusoid in the  $(r, \theta)$  plane representing all the lines that pass through it. For implementation reasons, both the  $r$  and  $\theta$  axes have to be discretized, along the axes, forming bins. The arrangement of the bins in the  $(r, \theta)$  plane itself gives the appearance of an image and is called the accumulator array.

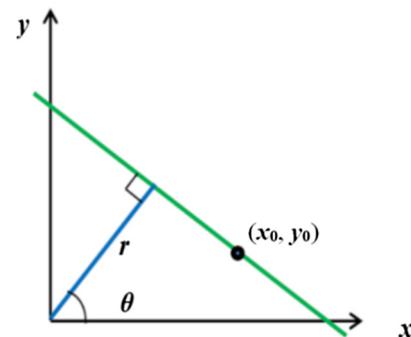
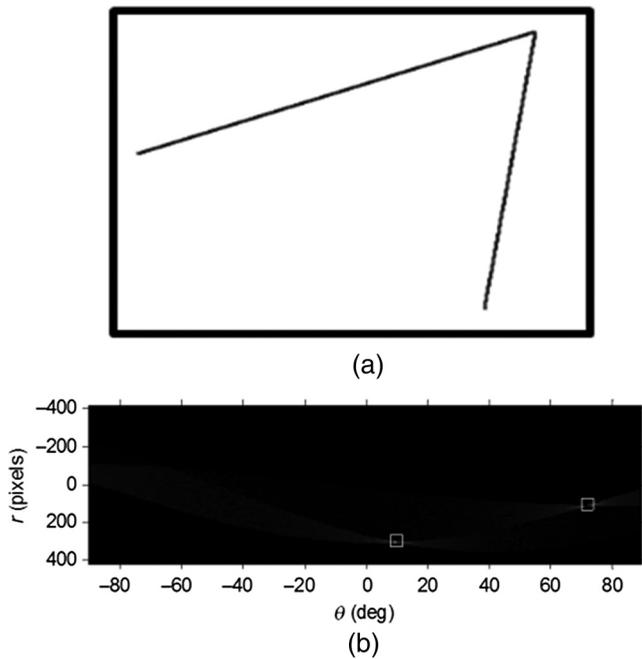


Fig. 2 Definition of the  $(r, \theta)$  pair for a straight line through point  $(x_0, y_0)$ .



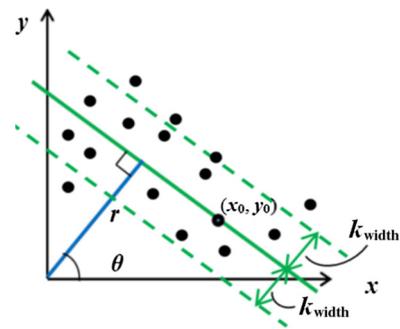
**Fig. 3** Hough transform: (a) a binary image consisting of two lines and (b) its accumulator array with two peaks representing the lines.

Then for each point in the image, a value of 1 will be added to the corresponding bins, which form the shape of a sinusoid. The intersections of the sinusoids of a number of detected pixels are detected as the straight line through those points. By finding out the local peaks in the accumulator array, the potential lines will be detected. Usually a thresholding is performed on the accumulator array to detect lines that are greater than a given length. An example is shown in Fig. 3(a), where two lines in an image result in two distinct peaks in the accumulator array, shown in Fig. 3(b), highlighted within a bounding box each. It must be noted here that the standard Hough transform will fail if points do not lie on a perfect straight line.

### 3.3 Fuzzy Hough Transform

If the Hough transform for straight-line detection is applied to an image, the straight line segments in it can be detected. However, crack segments do not always belong to straight lines, but most of the times are only nearly straight. Hence, the limitation of the Hough transform for straight lines must be overcome before it can be used to detect cracks. This deficiency is removed by the fuzzy Hough transform.

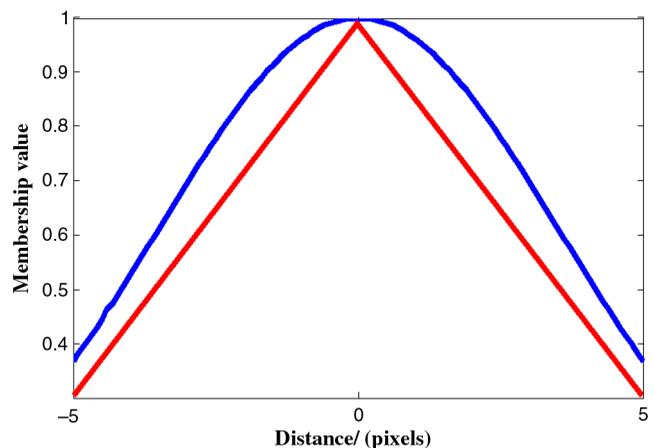
The fuzzy Hough transform allows for the fact that there can be pixels slightly away from a given straight line (i.e., a bin in the  $r - \theta$  accumulator array) due to sensor noise or due to any discrepancies in preprocessing operations, such as intensity thresholding or edge detection.<sup>34</sup> The closer the pixel to any line, hence to the corresponding accumulator array bin, the more the contribution of it goes to that specific bin. This capability of the fuzzy Hough transform is utilized here to detect crack lines that are inherently nonstraight. To explain further, assume there are a number of noncollinear crack points detected in an image. An example could be the black points in Fig. 4. Although not all of them are on a straight line, using a fuzzy membership function, they all can claim “right” to the solid line shown in



**Fig. 4** The influence area of fuzzy Hough transform, shown in dotted lines, also incorporating points away from a solid, straight line through  $(x_0, y_0)$ .

green, at various degrees of closeness. In this regard, referring to Fig. 4, the fuzzy version of the Hough transform also takes into account the points at up to a distance of  $k_{width}$  distance away from the straight line, on both sides. The contribution of the points in the area of interest, defined by the dotted lines in Fig. 4, is determined by fuzzy membership functions centered around the middle, solid line. Figure 5 shows two such membership functions, triangular and Gaussian, where a distance of zero from the straight line under consideration means a membership value of 1. In addition, the membership value reduces as the distance increases. The Gaussian membership function is chosen to be used in this work. In the fuzzy Hough transform, the membership value of any pixel in the  $k_{width}$  neighborhood is used as its contribution to the accumulator array. Hence, a pixel that is far away from the line considered makes a smaller contribution to the accumulator array, and the points that are close to a straight line lead to a large contribution to the accumulator array.

For a given set of points, several candidate lines will be tested before the line with the largest accumulated membership of neighboring points is chosen as the best candidate. It has been shown<sup>35</sup> that the easiest way to get the fuzzy contribution of all pixels is to form the accumulator array with the usual Hough transform and then convolve it with the following one-dimensional kernel in the  $r$  direction of the accumulator array



**Fig. 5** Two fuzzy membership functions for a kernel width of 5 pixels, triangular (red) and Gaussian (blue), and negative and positive distances are on either side of the straight line considered.

$$g(d) = \begin{cases} e^{-\frac{r^2}{k_{\text{width}}^2}} & \text{if } r < k_{\text{width}}, \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where  $r = k_{\text{width}}$  defines the width of the kernel. The larger the width of the kernel, the more nonstraightness in the points the algorithm can tolerate. However, on the downside, more noise pixels can also get detected as crack segments. For smaller kernel widths, noise detections will be reduced at the expense of detected crack segments. As a general rule, the fuzzy Hough transform is preferred for noisy images over the Hough transform.

Dividing the image into a number of tiles reduces the contribution of accumulated noise/texture pixels from the whole length and breadth of the image to the accumulator array bins. This way of dividing the image, and making an accumulator array for each of the tiles, is known as the tiled Hough transform.<sup>36</sup> A graphical explanation of the effectiveness of the tiled Hough transform is provided in Fig. 6. In Fig. 6, a number of point detections, in black, belonging to potential straight lines are shown in both the figures, representing an image. Figure 6(a) shows the standard Hough transform case where the image is used as a whole to generate a single accumulator array. As shown in Fig. 6(a), six noise pixels, aligned along a straight line, will give rise to the peak detection in the accumulator array. Hence, the noise response results in a FP, whereas the real line in the image, consisting of four detected points, does not get detected. Figure 6(b) shows the same set of detections, but now in an array of nonoverlapping square tiles, where an accumulator array each is formed for every tile. As shown in Fig. 6(b), the division of the image into a number of tiles increases the chance of detecting the correct crack segment over the noisy data. In Fig. 6(b), the actual line

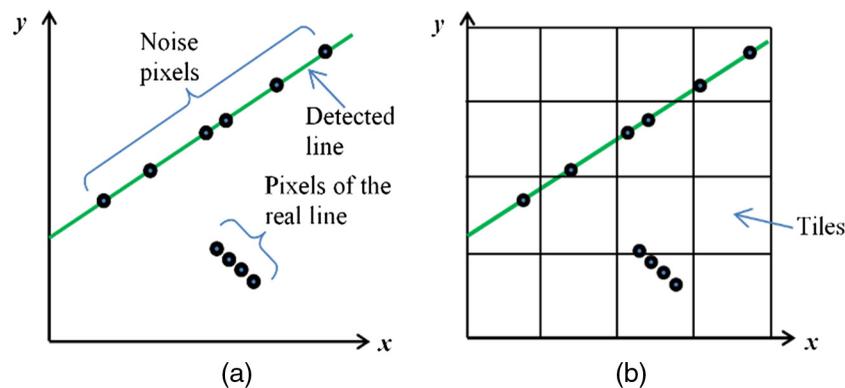
segment, consisting of three detected points, stands above the noise detections that have become separated from each other by the tiled configurations. Hence, the tiled accumulator array building, on the right, reduces the contribution of noise on line detections. Therefore, this scheme is used in the paper to reduce the influence of noise. In a similar manner, the misleading effects of surface texture, which are present heavily in some pavement images, on crack detection will also be kept under control. The overall flow of the algorithm is provided in Fig. 7.

In this work, a combination of the fuzzy and tiled variants of the Hough transform is employed to detect crack segments in pavement images. By doing this, the chance of detecting crack segments, which are essentially nonstraight line segments, embedded in a noisy, highly textured environment is vastly increased as shown below.

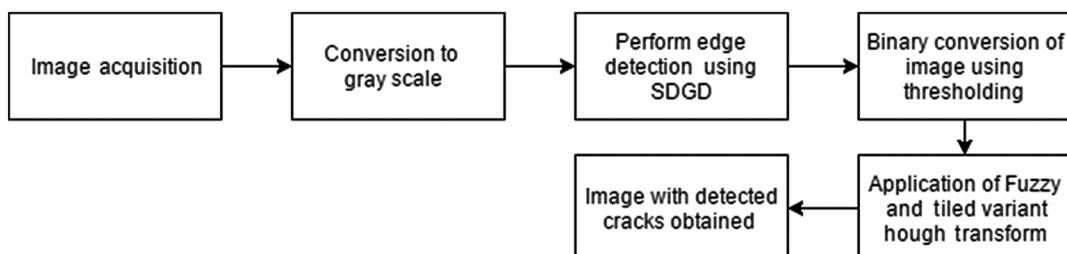
## 4 Results and Discussion

### 4.1 Images and Processing

Four different image datasets are used to test the proposed method. The different datasets are chosen to test the viability of the proposed crack detector in different imaging contexts encountered in real-life roads. Out of these four datasets, two are provided by Dynatest UK Ltd. They are obtained by the MFV of Dynatest. The MFV is equipped with intensity cameras and laser range imagers.<sup>7</sup> One intensity camera on the vehicle images the road surface perpendicularly capturing images similar to the one shown in Fig. 1. The image in Fig. 1 belongs to the dataset named database 1 (DB1). Nine hundred images make up this set. The physical resolution of the road captured by the image is about 4 mm/pixel. The grayscale images are 1040 × 1250 pixels in size.



**Fig. 6** Hough transform (a) considers image as a whole and (b) tiled version performs better in avoiding the influence of noise.



**Fig. 7** The flowchart of the crack detection scheme.

The MFC of Dynatest is also mounted with a triangulation-based laser scanner at the rear of the vehicle. Range (i.e., three-dimensional) images obtained from the scanner are also made available to the research. One such range image, rescaled to the intensity range of 0 to 255, i.e., 8-bit, is shown in Fig. 8(a). Dataset DB2 is formed by 900 such images. The lateral (i.e., on the road surface) resolution of the image, as with DB1, is 4 mm/pixel. DB2 images are also  $1040 \times 1250$  pixels in resolution. Figure 8(b) has a crack segment, of a vertical crack of good contrast and a horizontal crack of poor disparity from the background. This monochrome image set is obtained by the LRIS system of Pavemetrics Inc., Canada. LRIS images of a similar nature are included in dataset DB3 of 25 images. Its resolution is also 4 mm/pixel, with  $900 \times 1080$  pixels in size. DB3 images are also grayscale images. The last image set, DB4, consists of images taken by a digital camera. Images are captured using a Sony Cyber-shot DSC-W180 camera in ambient lighting. The resolution of these images is 0.33 mm/pixel. The image canvass itself is  $3264 \times 2448$  pixels. The road section is flat, and the camera is held perpendicular to the road surface, and at a constant height, while the images were captured. This image database incorporates images that have a strong presence of surface texture, mainly coming from large aggregates a lot greater than the first five datasets, in conjunction with cracks of varying contrasts. The images are taken in direct sunlight, hence shadows are shown and the images also have very poorly contrasted crack segments, due to the directional effects of the sun. Two such images from DB4 are shown in Figs. 9(a) and 9(b). There are 150 images in this dataset.

For all the images, side borders, etc., consisting of pavement markings are manually cropped out as the cracks that are found within the lane markings of a road are not the

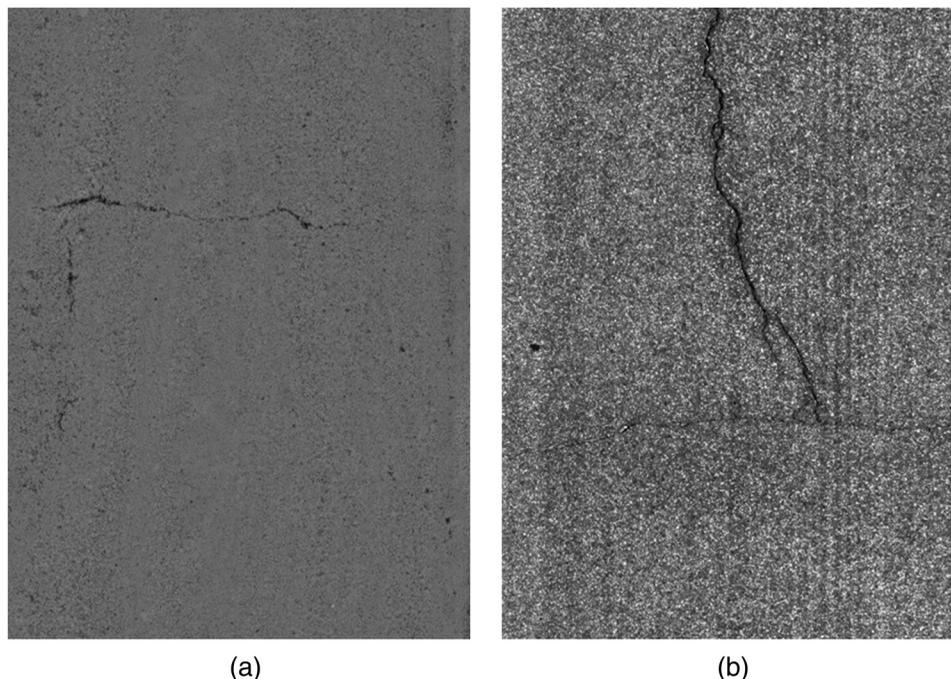
consideration of the current study. However, the automation of this cropping itself is not considered, as there are other standard techniques to detect lane markings, hence their removal is assumed a straightforward task.

The algorithm is coded in MATLAB<sup>®</sup>. The different datasets necessitated different values for the tile sizes and the other parameters in the fuzzy Hough transform. These details are set out in Sec. 4.2.

## 4.2 Results

For the images of DB1, the edge detection is performed with a Gaussian kernel size of  $6 \times 6$ . In an image that is processed with the SDGD filter, the dark edges are usually large in value, and any brighter areas will have low values. The resulting matrix from the SDGD operation is thresholded with a value of  $-0.05$ . The original image and its edge detection by SDGD are shown in Fig. 10. The edge image is a binary image. The detection of texture, especially in the darker vertical bands toward the left and right edges of the image, must be noted. Once the edges are detected, the tiled Hough transform is operated on the binary edge image. A tile size of  $25 \times 25$  pixels is found to be the best for this purpose.

A fuzzy kernel width of 1 pixel is used, as at this scale many edge segments of cracks take a near straight-line form, reducing the need to incorporate large excursions from the center-line, which represents the final detection (see Fig. 4). For the discrete accumulator array, an  $r$  resolution of 1 pixel and a  $\theta$  resolution of 3 deg are used. The accumulator array of every tile is thresholded with a value of 17. For each tile, the accumulator bins, i.e., the discrete  $r$  and  $\theta$  values, which consist of bin values greater than the threshold, are identified. Then, these  $r$  and  $\theta$  values are used, in conjunction with Eq. (6), to identify the line segment within that tile. The identified lines are plotted back on the



**Fig. 8** Images from two databases: (a) a DB2 range image containing cracks obtained by the MFV and (b) a DB3 image from the LRIS of Pavemetrics.

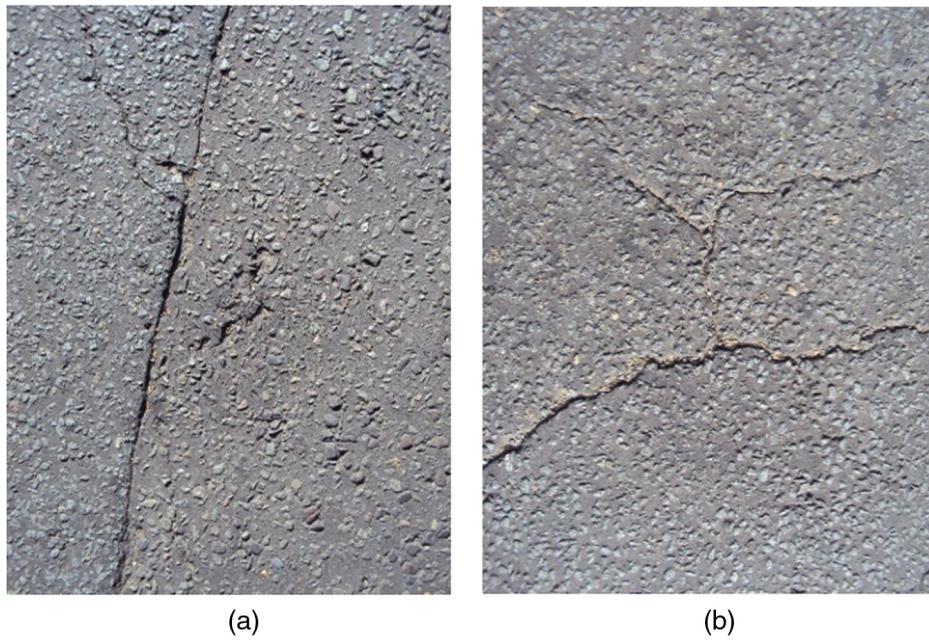


Fig. 9 DB4 dataset: (a and b) two images.

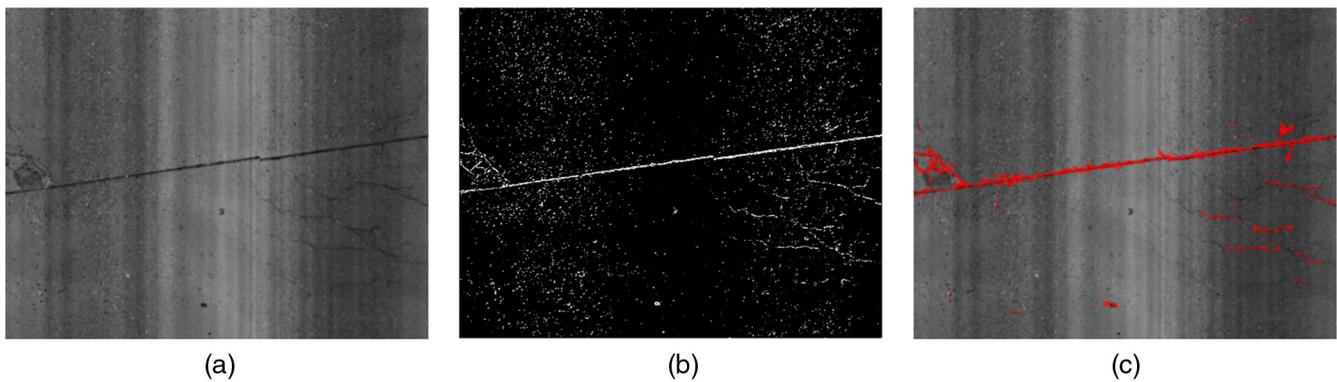


Fig. 10 Images of DB1 at various processing stages: (a) original image, (b) edges detected by SDGD, and (c) crack segments detected.

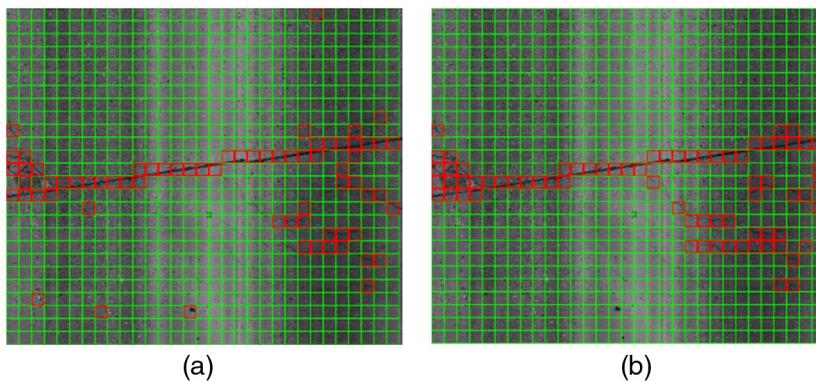


Fig. 11 A DB1 image: (a) tiled-based detections and (b) the ground truth.

image using red pixels. The image in Fig. 11(a) shows the detected lines. For the same image, Fig. 11(b) shows all the tiles analyzed (in green borders), and red tiles are drawn wherever crack segments are detected. For the pavement image, tiles of size  $25 \times 25$  pixels are manually analyzed,

and crack tiles are identified. This manual crack tile identification is assumed as the ground truth, as shown in Fig. 11 (a), and based on this, three quality measures, namely precision, recall [a.k.a. true positive (TP) rate], and FP rate are estimated for detections in the following manner:

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}}, \quad (9)$$

$$\text{Recall} = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}}, \quad (10)$$

$$\text{False positive rate} = \frac{\text{False positives}}{\text{False positives} + \text{True negatives}}. \quad (11)$$

The detection shown at the top, in Fig. 11, when compared against its ground truth, has 57 tiles that are true positives (TP), 13 FP tiles, and 16 tiles detected as false negatives (FNs). There are 806 (i.e.,  $26 \times 31$ ) tiles in total. This means that true negatives are 720 tiles. Using Eqs. (9) and (10), this detection corresponds to a precision and recall of 81.4% and 78.1%, respectively. From Eq. (11), the FP rate for the detection is 0.0177.

On a closer inspection of the detected image, the unusually large texture blobs (probably due to aggregates sticking out due to top layer wear) and a few more regular texture blobs lie close to each other, hence triggering a high response in the accumulator array, resulting in FPs. The FNs are mainly from the extremely low-contrast crack segments, which can be guessed by the global crack network pattern for the naked eye, but get a very low Hough transform response based on the localized tiles.

Now all images in DB1 are manually processed for ground truth. Then they are processed with the proposed algorithm, but with varying levels of the accumulator array threshold between 1 and 55, in the increment of 1. For each threshold value, the FP and TP rates are calculated, using Eqs. (10) and (11), and the values are plotted against each other. The resulting receiver-operator characteristic (ROC) curve is shown in Fig. 12. Based on the performance, a threshold value of 13 is selected. The resulting precision and recall values for database DB1 are 44.4% and 93.2%,

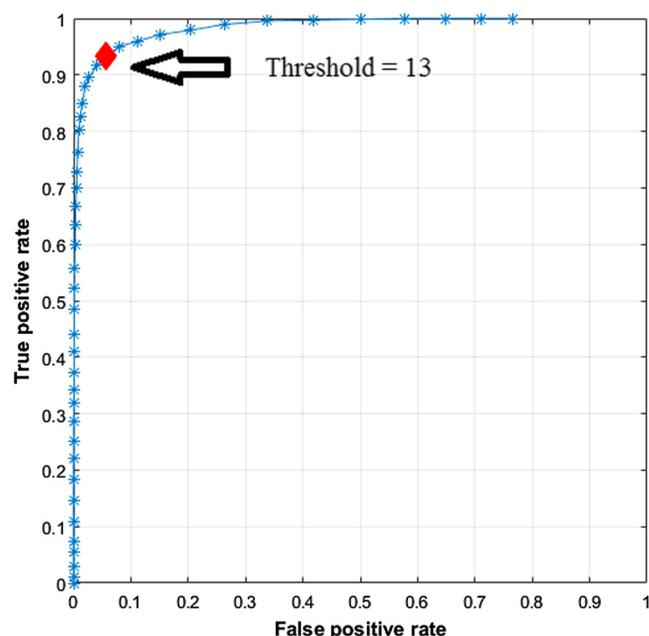


Fig. 12 ROC curve for DB1 for changes in threshold.

Table 1 Processing parameters used for different datasets.

Data	SDGD threshold	Square tile size (pix)	Kernel size (pix)	Accumulator array threshold (pix)
DB1	-0.05	25	1	13
DB2	-0.05	25	2	24
DB3	0.05	25	2	37
DB4	-0.1	150	17	1925

respectively. A similar process is followed for the other databases, and the chosen values are given in Table 1. The ROC curves themselves are left out for space considerations here.

Although the parameter values reported in Table 1 appear to be random, a consideration of the image resolutions for different databases offers a better view from a general parameter selection perspective. In this regard, it is said earlier that DB1, DB2, and DB3 are at 4-mm/pixel resolution, whereas DB4 is 0.33 mm/pixel. Based on this, the following values can be recommended as good starting points for the parameters for crack detection based on the method proposed here. Hence, to be applicable for any pavement image resolution, parameters are specified here at the pavement level instead of in pixels (these can also be easily worked out from Table 1 and image resolutions). The recommended values are all referring to the actual sizes at road level, a tile size in the range  $50 \times 50$  to  $100 \times 100$  mm<sup>2</sup>, a kernel size between 4 and 8 mm, and a threshold value of pixels representing an area of 200 to 600 mm<sup>2</sup>.

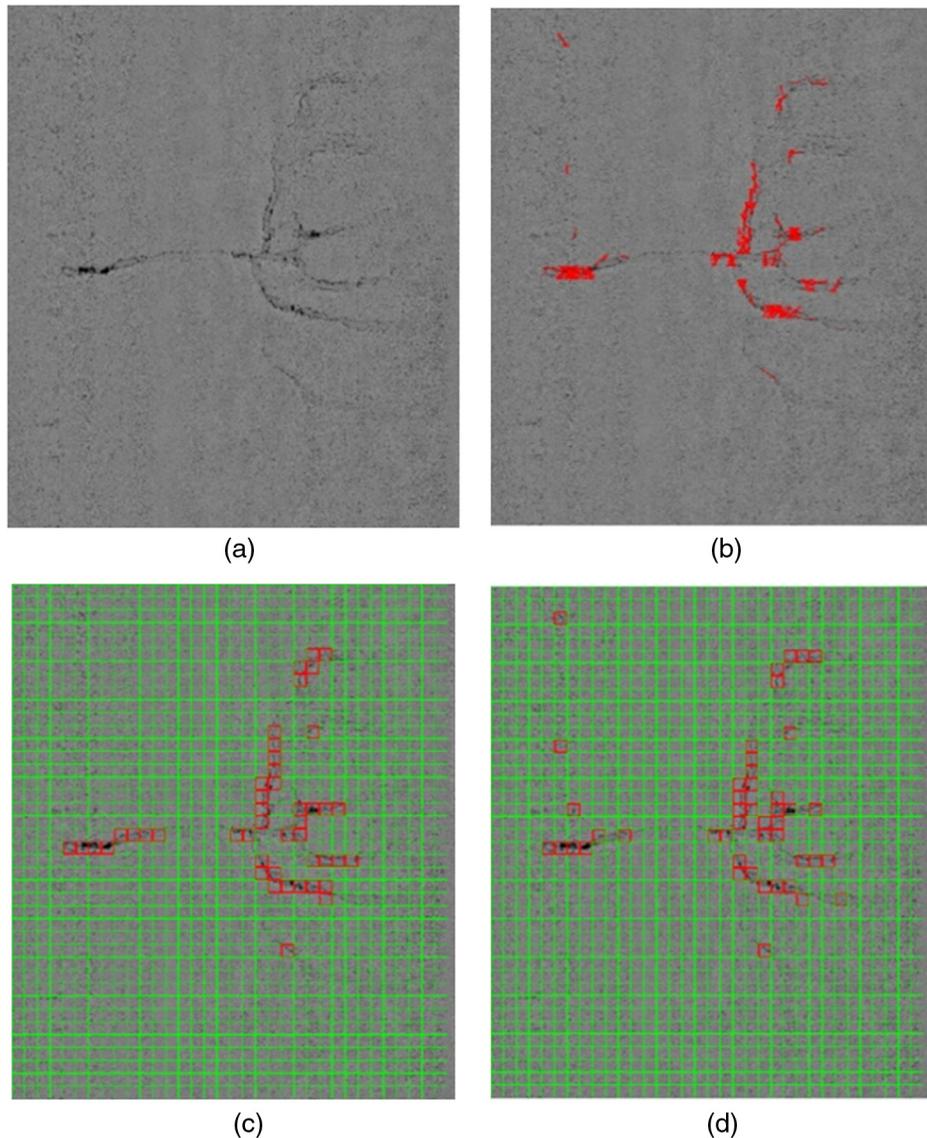
Images in DB2 are processed with the parameters shown in Table 1. A sample image is shown in Figs. 13(a)–13(d) together with processed images and the ground truth of cracks, identified manually. When tile-based detections are compared against the ground truth, there are 38 TP tiles together with 8 FPs and 8 FNs. This results in 82.6% precision and a recall value of 82.6%.

A sample detection of a DB3 image is shown in Figs. 14(a)–14(d), together with the original and the tile-based ground truth. There are 36 TPs, 10 FPs, and 7 FNs (all in number of tiles). This count results in a 78.2% and 83.7% precision and recall, respectively. Figure 15 shows an example image from DB4 showing the detections and the ground truth, respectively.

As the images in DB4 (Fig. 15) are much higher in resolution than DB2 (Fig. 13) and DB3 (Fig. 14), dedicating more pixels to a given area, a larger tile size of  $150 \times 150$  pixels is used. A SDGD threshold value of -0.1 is used. In conjunction with large tile sizes, a kernel width of 17 pixels is used with a threshold value of 1925 for the accumulator array. On the final detection, there are 19 TP, 0 FP, and 12 FN tiles. The precision and recall values for this detection are 100% and 61.3%, respectively. The final detection accuracies for all datasets are provided in Table 2.

### 4.3 Benchmarking

To benchmark the detection performance of the proposed method, it is decided to compare it against a well-known crack detection method in the literature. In this regard, the method proposed by Oliveira and Correia<sup>16</sup> is chosen, due



**Fig. 13** A DB2 image: (a) raw image, (b) its detection, (c) ground truth identified by tiles, and (d) the detections in tiles.

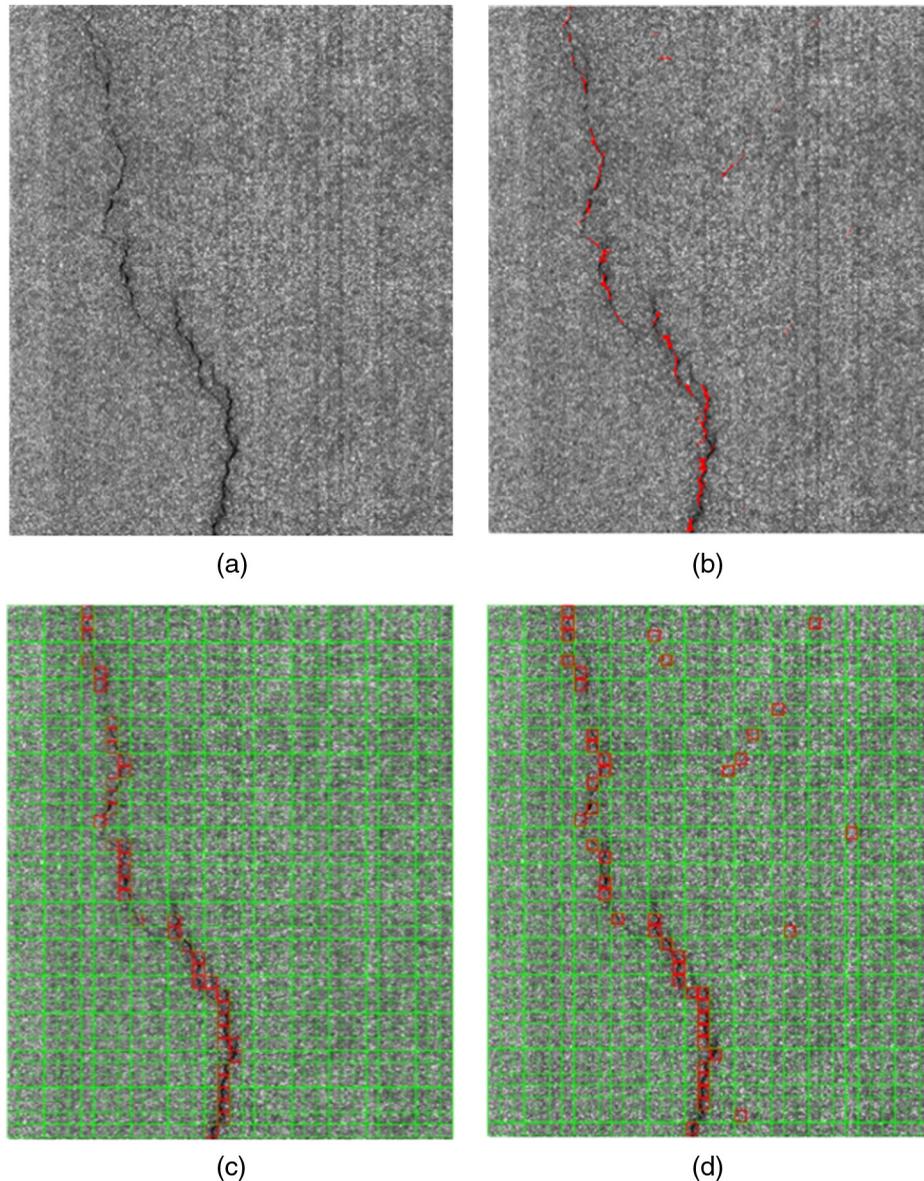
to the following reasons: its highly sophisticated nature and very high values of precision and recall reported (84% to 95% and 95%, respectively), and the clear documentation that allows an easy reproduction of their algorithm. The tile-based crack detection algorithm of Oliveira and Correia<sup>16</sup> consists of extensive preprocessing, which is reported, in detail, in Oliveira and Correia.<sup>37</sup> The modified-Otsu method used in the algorithm is clearly detailed in Ref. 37. The method has been implemented as a MATLAB<sup>®</sup> toolbox for road crack detection.<sup>38</sup> This tool, named CrackIT, has been used here to benchmark the results obtained for DB1 to 4.

In CrackIT detection output image files, green crosses show tiles that are TP. Similarly, yellow crosses represent FPs and red ones show FNs in Figs. 16 and 17. Figure 16(a) shows the detections obtained from CrackIT for an image in DB1. It can be seen that, unlike the detections by the proposed method, shown in Fig. 11(a), the thin, low-contrast cracks are not detected by CrackIT at all. The intensity

changes of the background seen in Fig. 16(a) could also have contributed to the lower detection by CrackIT.

However, the FPs of five tiles are smaller than the 7 obtained by the current method for that image. Although when the whole of DB1 is considered, the proposed method has a lower percentage of FPs. This is reflected in Table 2, where the summaries of results, for the two algorithms, are given for all four image sets. In this regard, the precision value for the proposed method is 44.4% against 77.9% from CrackIT. However, when it comes to FNs, the proposed method outperforms by far. Fig. 16(a) shows that most of the low-contrast crack segments are not detected, whereas in Figs. 11(a) and 11(b), a considerable portion is detected. This is reflected in the considerable performance difference in recall: 93.2% by the fuzzy Hough method versus 56.1% by CrackIT.

Figure 16(b) also shows an image from DB2 processed by CrackIT. Many crack segments in the image are missed (red crosses), and one tile is detected falsely as crack. Comparing



**Fig. 14** DB3 images: (a) raw image, (b) its detection, (c) ground truth identified by tiles, and (d) the detections in tiles.

this against Fig. 13(d) image, it is quite easy to identify that the proposed method has performed much better than CrackIT. The overall precision values for the two methods for DB2, in Table 2, also reinforce the superiority of the current method. Moreover, the recall value of 76.0% versus 53.3% shows the advantage gained by using the fuzzy Hough transform for crack detection.

On DB3, the method of Oliveira and Correia<sup>6</sup> performs slightly better in both the precision and recall aspects. Figure 17(a) shows that the low-contrast segments are missed by CrackIT. On these segments, the fuzzy Hough transform also has not performed very well either, although it fared better than CrackIT. However, CrackIT's performance on good contrast segments is better than the proposed algorithm (although detecting high-contrast crack segments is not usually a challenge). DB4 has the poorest of performances obtained via CrackIT. As seen in Fig. 17(b), many noncracked areas are detected as cracks by CrackIT. Since

this is an over detection, there are no FNs in Fig. 17(b). Owing to this characteristic also on other images from DB4, the recall value is very high via CrackIT (about 88.9%). Due to many FPs, the precision is very low at 5.9%. In comparison, the fuzzy Hough transform performs better, when both the aspects are considered. The random surface texture, of DB4, in combination with uncontrolled illumination increases FPs, bringing down the overall precision values for both the algorithms, and much higher for CrackIT.

In summary, the proposed algorithm has performed better than the state of the art, in general. Even in one dataset, i.e., DB3, where it performed worse than CrackIT, it detected low-contrast crack segments better. The low-contrast crack segments from the other datasets are detected better by the fuzzy Hough method. In addition, the quality of detection on highly textured pavements and that under varying illumination is found to be better with the proposed method.

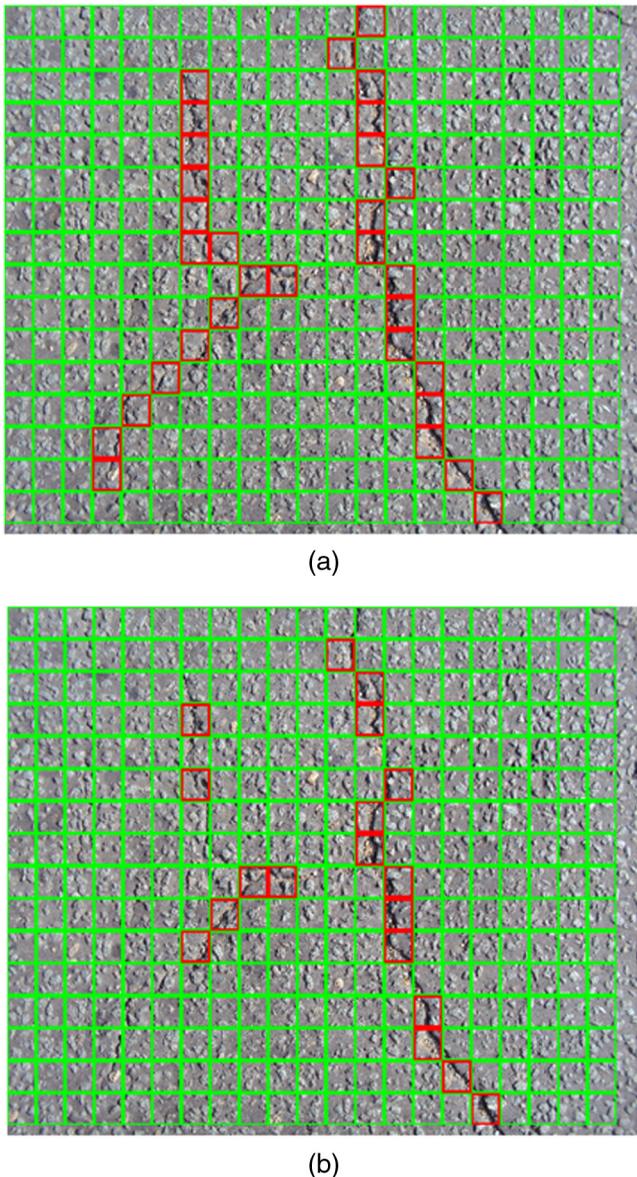


Fig. 15 A DB4 image: (a) detection and (b) the ground truth.

Table 2 A performance comparison of the algorithms.

Dataset	Proposed algorithm		Algorithm of Oliveira and Correia	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)
DB1	44.4	93.2	77.9	56.1
DB2	76.8	76.0	49.0	53.3
DB3	66.0	60.5	71.7	63.7
DB4	63.2	57.8	5.9	88.9

#### 4.4 Issues in Detection

It has been shown that the proposed algorithm consistently outperforms the one that is proposed by Oliveira and Correia.<sup>16</sup> Cracks, as small as 8 mm in width, are detected

from the databases DB1 and DB2, which consist of images having a resolution of 4 mm/pixels. This means that cracks as small as 2 pixels wide are detected by the proposed algorithm. However, the performance of 44% to 75% precision and 75% to 95% recall on the Dynatest datasets DB1 and DB2 still needs improvement. The FPs come from the fact that the background texture is comparatively similar in magnitude to the cracks present. In this regard, as local, tile-based detection is performed, the impact of surface texture, although small, is still felt. On the other hand, very low-contrast cracks are still found to be hard to detect. If the accumulator array threshold is lowered to pick out this low-contrast cracking, the strong surface texture on other parts of the image results in increased FPs. A manual selection of the threshold has been difficult and this is where the ROC analysis (e.g., Fig. 12) is found to be useful.

Dataset DB3 presents even more problems. Especially, the low-contrast horizontal crack segments, which are lateral to the length of road, have gone largely undetected, giving rise to large FNs; the consequence is a lower recall value. However, the consistent low-contrast of lateral cracks from LRIS appears to be an issue with its lighting. Many pavement images have high powered lasers pointing in the lateral direction of roads, and this lighting will create a good contrast to longitudinal cracks, whereas with lateral cracking, the illumination gets to all areas of the trench-like structure created by these cracks, resulting in low-contrast. The effect is clearly seen in the image in Fig. 17(a) from DB3. A similar problem is also present with DB4 images where the direction of the sun poorly contrasts the segments of cracks oriented toward it. The extreme levels of texture, found in the local roads from where the images are captured, bring down the accuracy figures even below that for DB3.

#### 4.5 Processing Times

The images are processed on a PC with an Intel Core-i7 processor (2.10 GHz) having an 8-GB RAM. The algorithm is coded in MATLAB<sup>®</sup>. The average times consumed to process each image in the databases are shown in Table 3.

#### 5 Future Developments

Although the proposed algorithm performs much better than the state of the art, background texture in DB1 and low-contrast cracking in DB3 have been identified to pose fundamental difficulties in the detection. In this regard, the authors have recently been experimenting with probabilistic detection methods for pavement cracks and have had reasonable success in detecting low-contrast cracking (the results of which are yet to be published). The idea now is to integrate the strengths of both fuzzy Hough and probabilistic detections to advance the detection capabilities further. It also becomes necessary to test a lot more images with the proposed algorithm to test its robustness. In this regard, images that also consist of other defects, together with cracking, will also be tested to evaluate the sensitivity of the proposed method further.

As seen in Table 3, the times taken for the algorithm, within MATLAB<sup>®</sup>, are in the order of tens of seconds. Even though MATLAB<sup>®</sup> is not a real-time development environment, but far from it, the processing times will have to be drastically reduced. The main algorithmic element that consumes much time is found to be, unsurprisingly, the

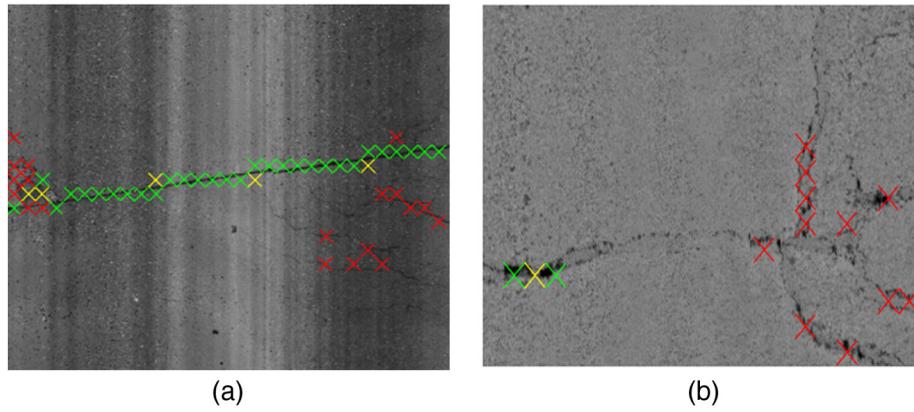


Fig. 16 Detections by the tool of Oliveira and Correia:<sup>38</sup> (a) DB1 and (b) DB2 images.

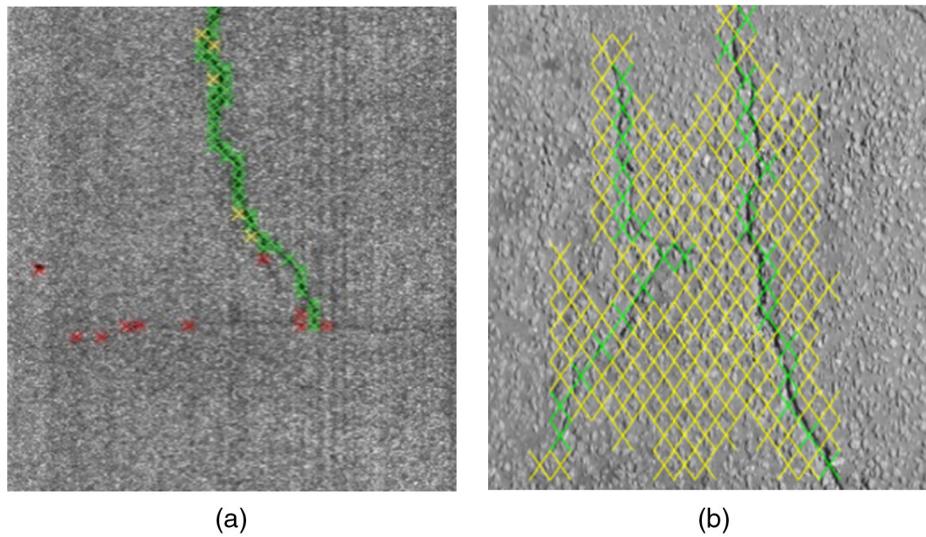


Fig. 17 Detections by the tool of Oliveira and Correia:<sup>6</sup> (a) DB3 and (b) DB4 images.

Table 3 Average processing times for per image.

Dataset	Time taken (s)
DB1	32.0
DB2	31.5
DB3	80.9
DB4	69.7

tiled Hough transform. As a number of accumulator arrays are formed for each of the tiles, the sequential processing within MATLAB<sup>®</sup> does not help. In this regard, the algorithm is ported to C++ with OpenCV library functions, which is better suited for real-time implementations than MATLAB<sup>®</sup>, and being tested. Most importantly, the strengths of parallel computing are being made use of via graphical processing unit (GPU) usage. The accumulator array of a tile does not have any dependency on the arrays of any other tile in the image, making the formulation of these arrays to be parallelized without any complexities.

An Intel Core-i7 processor has four cores, so a four-way parallelism is immediately possible, at least for intermediate image processing operations, such as accumulators array formulation. In addition, dedicated GPU units, such as nvidia TEGRA X1, are also considered to massively parallelize the accumulator array formulation, as they easily consist of 10s of cores. This is expected to bring the run-time of the algorithm to real-time requirements.

## 6 Conclusions

This paper presents a method whereby fuzzy Hough transform is proposed as a technique to detect cracks from pavement images. The proposed algorithm makes use of the superior capabilities of the Hough transform to detect cracks from highly textured images. Two added features of the Hough transform, in terms of fuzzy membership and tiled detections, are found to reduce the effect of texture pixels and noise pixels. The resulting method provides accuracies and precision in the range of 44% to 75% for images captured with an actual road inspection imager, the MFV of Dynatest. The recall values are between 75% and 95%. It has been shown that the proposed algorithm is highly effective when it comes to detecting low-contrast cracks, especially among highly textured and nonuniformly

illuminated backgrounds. A comparison with a state-of-the-art algorithm shows that there is a consistent advantage in using the proposed algorithm, also for other image sets, including the one captured under totally uncontrolled lighting.

### Acknowledgments

The permission of Pavemetrics Systems Inc., Canada, to use their images in this paper is greatly acknowledged.

### References

- Central Intelligence Agency, "The world factbook," <https://www.cia.gov/Library/publications/the-world-factbook/fields/2085.html> (1 February 2017).
- Trading Economics, "World-roads, paved (% of total roads)," <http://www.tradingeconomics.com/world/roads-paved-percent-of-total-roads-wb-data.html> (1 February 2017).
- J. S. Miller and W. Y. Bellinger, "Distress identification manual for the long-term pavement performance program," 4th revised ed., Rep. FHWA-RD-03-031, Federal Highway Administration, 2003, <http://www.fhrc.gov/pavement/ltp/reports/03031/03031.pdf> (July 2017).
- K. S. Jassal, "Development of potholes from cracks in flexible pavements," MASC Dissertation, Concordia University (1998).
- Ontario Hot Mix Producers Association, "The ABCs of potholes," <http://www.onasphalt.org/files/Publications/ABCs%20of%20Potholes.pdf> (28 August 2017).
- Dynatest, "Dynatest multi-functional vehicle," <http://www.dynatest.com/equipment/functional/imaging.aspx> (1 February 2017).
- S. Chambon, "Detection of points of interest for geodesic contours application on road images for crack detection," in *7th Int. Conf. on Computer Vision Theory and Applications*, Vilamoura, Algarve, Portugal, pp. 210–213 (2011).
- C. Zou et al., "Crack tree: automatic crack detection from pavement images" *Pattern Recognit. Lett.* **33**(3) 227–238 (2012).
- K. Vaheesan et al., "Tiled fuzzy Hough transform for crack detection," *Proc. SPIE* **9534**, 953411 (2015).
- H. Lee, "Accuracy, precision, repeatability and compatibility of the Pavedex PAS 1 automated distress measuring device," *Transp. Res. Rec.* **1311**, 136–143 (1991).
- S. Chambon and J. Moliard, "Automatic road pavement assessment with image processing: review and comparison," *Int. J. Geophys.* **2011**, 989354 (2011).
- C. Koch et al., "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Adv. Eng. Inf.* **29**(2), 196–210 (2015).
- E. Teomete et al., "Digital image processing for pavement distress analyses," in *Proc. of the Mid-Continent Transportation Research Symp.*, p. 13 (2005).
- R. Adhikari, O. Moselhi, and A. Bagchi, "Image-based retrieval of concrete crack properties for bridge inspection," *Autom. Constr.* **39**, 180–194 (2014).
- H. Oliveira and P. L. Correia, "Automatic road crack detection and characterization," *IEEE Trans. Intell. Transp. Syst.* **14**(1), 155–168 (2013).
- H. Oliveira and P. L. Correia, "Automatic road crack segmentation using entropy and image dynamic thresholding," in *Proc. of 17th European Signal Processing Conf.*, Glasgow, pp. 622–626 (2009).
- P. P. Milella, "Morphological aspects of fatigue crack formation and growth," in *Fatigue and Corrosion in Metals*, pp. 73–108, Springer-Verlag, Italy (2013).
- E. Zalama et al., "Road crack detection using visual features extracted by Gabor filters," *Comput.-Aided Civ. Infrastruct. Eng.* **29**(5), 342–358 (2014).
- M. Salman et al., "Pavement crack detection using the Gabor filter," in *16th Int. IEEE Conf. on Intelligent Transportation Systems (ITSC '13)*, pp. 2039–2044 (2013).
- Y. O. Ouma and M. Hahn, "Wavelet-morphology based detection of incipient linear cracks in asphalt pavements from RGB camera imagery and classification using circular Radon transform," *Adv. Eng. Inf.* **30**(3), 481–499 (2016).
- A. Zhang et al., "Matched filtering algorithm for pavement cracking detection," *Transp. Res. Rec.* **2367**(1), 30–42 (2013).
- R. Amhaz et al., "Automatic crack detection on two-dimensional pavement images: an algorithm based on minimal path selection" *IEEE Trans. Intell. Transp. Syst.* **17**(10), 2718–2729 (2016).
- L. Zhang et al., "Road crack detection using deep convolutional neural network," in *IEEE Int. Conf. on Image Processing (ICIP '16)* (2016).
- D. Zhang et al., "An efficient and reliable coarse-to-fine approach for asphalt pavement crack detection," *Image Vision Comput.* **57**, 130–146 (2017).
- H. Sohn et al., "Monitoring crack changes in concrete structures," *Comput.-Aided Civ. Infrastruct. Eng.* **20**(1), 52–61 (2005).
- K. Maalmi et al., "Crack defect detection and localization using genetic-based inverse voting Hough transform," in *Proc. of the 16th Int. Conf. on Pattern Recognition (ICPR '02)*, Vol. **3**, (2002).
- T. M. Meksen, M. Boudraa, and R. Draï, "Detection of cracks in materials using the randomized Hough transform on ultrasonic images," in *Proc. of the 6th WSEAS Int. Conf. Signal Processing, Computational Geometry and Artificial Vision*, pp. 202–206 (2006).
- M. Gavilan et al., "Adaptive road crack detection system by pavement classification," *Sensors* **11**(10), 9628–9657 (2011).
- T. S. Nguyen, M. Avila, and S. Begot, "Automatic detection and classification of defect on road pavement using anisotropy measure," in *European Signal Processing Conf.*, pp. 617–621 (2009).
- D. N. Vivekanandreddy, A. Kammar, and B. Sowmyashree, "Hough transforms to detect and classify road cracks," *Int. J. Eng. Res. Technol.* **3**(6), 1500–1505 (2014).
- H. Rababah, "Asphalt pavement crack classification: a comparative study of three AI approaches: multilayer perceptron, genetic algorithms, and self-organizing maps," MSc Thesis, Indiana University, South Bend (2005).
- M. Sonka, V. Hlavac, and R. Boyle, *Image Processing Analysis, and Machine Vision*, 2nd ed., PWS Publishing, California (1999).
- F. Y. A. Rahman et al., "Enhancement of background subtraction technique using a second derivative in gradient direction filter," *J. Electr. Comput. Eng.* **2013**, 21 (2013).
- S. Jaime-Castillo, J. M. Medina, and A. Garrido, "A method for locating the iliac crests based on the fuzzy Hough transform," in *IFSA/EUSFLAT Conf.*, pp. 1063–1067 (2009).
- H. H. Joon, L. T. Kóczy, and T. Poston, "Fuzzy Hough transform," *Pattern Recognit. Lett.* **15**(7), 649–658 (1994).
- C. R. Jung and R. Schramm, "Parallelogram detection using the tiled Hough transform," in *Proc. of Int. Conf. on Systems, Signals and Image Processing*, pp. 177–180 (2006).
- H. Oliveira and P. L. Correia, "Supervised strategies for crack detection in images of road pavement flexible surfaces," in *Proc. 16th European Signal Processing Conf. (EUSIPCO)*, Lausanne, Switzerland, pp. 25–29 (2008).
- CrackIT, "A MATLAB toolbox for road crack detection and characterization," <http://amalia.img.lx.it.pt/CrackIT/> (1 February 2017).

**Senthan Mathavan** is an engineer and researcher working in the areas of machine vision and mechatronics. He received his BSc degree in mechanical engineering from the University of Peradeniya, Sri Lanka, in 2005. He then obtained his PhD in mechatronics from Loughborough University, UK, in 2009. He held a Rolls-Royce funded postdoctoral research position at Loughborough University from 2010 to 2011. From 2011 to 2016, he was a mechatronics engineer with ASML, the Netherlands. Currently, he is a system architect at Nobleo Technology, The Netherlands. He is also a visiting fellow at Nottingham Trent University, UK, since 2012. He has carried out research and development in the aerospace, semiconductor, and transportation industries.

**Kanapathippillai Vaheesan** was a first-year PhD student at the University of New South Wales, Sydney, at the time of his untimely death in 2015. Earlier, he was with Huawei Technologies in Colombo, Sri Lanka. He has a bachelor's degree in electrical and electronics engineering from the University of Peradeniya, Sri Lanka.

**Akash Kumar** received his bachelor's degree in mechatronics engineering from the College of Electrical and Mechanical Engineering, National University of Sciences and Technology (NUST), Islamabad, Pakistan, and followed a nondegree program at Mississippi State University for one semester. He has worked as a design and simulation engineer at Hubstaff, Karachi, Pakistan. His research interests include planetary robotics, medical imaging, sensor systems, unmanned aerial vehicles, and intelligent agricultural systems.

**Chanjief Chandrakumar** is a doctoral research scholar at the School of Engineering and Advanced Technology, Massey University, and also affiliated to the New Zealand Life Cycle Management Centre. His PhD aims at developing an absolute sustainability assessment method by integrating three popular frameworks: sustainable development goals, planetary boundaries, and life cycle analysis. Prior to commencing his PhD at Massey, he worked at the University of Peradeniya and the University of Jaffna, Sri Lanka, as an assistant lecturer. Moreover, for the 2016 academic year, he was a T.I.M.E. European Summer School Scholar in Sustainability and the Global Economy at KTH Royal Institute, Sweden. He graduated from the University of Peradeniya in 2016 with first-class honors after studying production engineering.

**Khurram Kamal** is serving as an assistant professor in the Department of Mechatronics Engineering, NUST, Islamabad, Pakistan. Prior to his PhD research at Loughborough University, UK, he has also worked as an assistant manager of production planning and control at the Press Shop Department of Pakistan, Suzuki Motor Company Ltd., from 2003 to 2004. His areas of interest are artificial intelligence, image processing, embedded systems, intelligent machines, condition-based maintenance, unmanned aerial vehicles, and sustainability.

**Mujib Rahman** is presently working as a senior lecturer of civil engineering at the School of Engineering and Design, Brunel University.

He received his BS degree in civil engineering from Bangladesh Engineering University and Technology, Bangladesh, his MSc degree in structural engineering from City University London, UK, and his PhD from the University of Nottingham, UK. He has been actively involved in pavement engineering-related research, consultancy, and teaching for more than 12 years. He was a recipient of an Institution of Civil Engineers East Midland Merit Award in 2013 for his research on the performance of asphalt patch repair.

**Martyn Stonecliffe-Jones** has been the head of consultancy at Dynatest International A/S since 2011. Before this, he was the technical director at Jacobs Engineering for 10 years.