Weighted Neighborhood Pixels Segmentation Method for Automated Detection of Cracks on Pavement Surface Images

Lu Sun[1]; Mojtaba Kamaliardakani[2]; and Yongming Zhang[3]

Abstract: A new method is designed to detect and segment a crack on a pavement surface image from its background. Gray images of pavement surface have been collected from asphalt concrete pavement on an interstate highway in Maryland using charge-coupled device (CCD) digital cameras. These pavement surface images contain different types of pavement surface distresses and pavement markings. The first step of the new algorithm is preparing a uniform background by applying the new average brightness level of each column. The weighted neighborhood pixels method is proposed, which is based on the intensities of all pixels in three surrounding loops. Seven different patterns are studied and compared, which leads to the best performance eight-direction pattern in terms of accuracy and robustness for feature extraction of pavement images with cracking; then, a local threshold approach and shape filtering using eccentricity value parameters are applied to enhance the candidate cracks. Finally, crack fragments are connected by using a dilation operator. The performance of the new method is evaluated against the ground truth data using manual detection and segmentation. The results show that the developed automated detection and segmentation method is accurate, fast, robust, and suitable for online pavement condition assessment. **DOI: 10.1061/(ASCE)CP.1943-5487.0000488.**

Author keywords: Automated crack detection; Image processing; Pavement condition assessment; Road.

Introduction

When cracks appear in pavement but are untreated, they propagate out and become more severe with continued traffic loadings. Meanwhile, cracks allow water to penetrate the surface, leading to progressive upheaval of the pavement surface via the *frost heave* process. Keeping pavement in good repair will save motorists from additional operating costs. Pavement surface distress evaluation is an important part of a pavement management system for maintenance and rehabilitation. Pavements with initial deterioration can be identified early through periodic pavement inspections.

Traditionally, pavement surface distresses have been identified by human inspectors who walk along the road and produce report sheets through visual surveys. These manual inspections are time-consuming, costly, subjective, and unsafe (Cheng et al. 2003). Digital images are the most important source of information for qualitative evaluation of distresses. Advances in computer technology, digital image acquisition, and digital image processing techniques allow local agencies to use digital image processing for pavement distress analyses (Teomete et al. 2005). Automated pavement distress inspection is able to improve accuracy and subjectivity.

Collection and analysis of pavement images with distresses has received considerable attention due to its potential to improve the quality of pavement condition assessment. In the past two decades, efforts have been made to develop automated surface distress detection systems using image processing techniques. However, a fully automated system still remains a challenge in pavement surface distress detection and classification (Tsai et al. 2009).

In this study, a new feature extraction (segmentation) method based on the weighted neighborhood pixel is proposed and implemented for automated online detection and segmentation of cracks on pavement surface images recorded on video or photographic film. The advantage of this weighted neighborhood pixel segmentation method as compared to existing methods lies in its accuracy and efficiency. The contributions of the paper are as follows: the weighted neighborhood pixel method is used for the first time to automatically detect and segment cracks on pavement surface images. An algorithm is designed and implemented through optimal pattern selection of seven different patterns, achieving a high accuracy and a high processing efficiency. The new method is fast, automated, and produces objective and consistent assessment of pavement cracking.

Literature Review

Since the early 1990s, researchers have made considerable attempts to develop an automated pavement distress inspection system. Such a system consists of two parts: an automated survey for collecting pavement surface images, and image-based automated evaluation (rating) of pavement distress. While analog area scanners and line scanners have been used for acquisition of pavement surface images in the field for more than a decade, the automated evaluation of pavement distress based on pavement images is far from mature.

[1]Professor, School of Transportation, Southeast Univ., Nanjing, Jiangsu 210096, China; Professor and Chairman, Dept. of Civil Engineering, Catholic Univ. of America, Washington, DC 20064; and Adjunct Professor, School of Architecture and Civil Engineering, Xiameng Univ., Xiameng, China (corresponding author). E-mail: sunl@cua.edu

[2]Ph.D. Candidate, School of Transportation, Southeast Univ., Nanjing, Jiangsu 210096, China. E-mail: ardakani@seu.edu.cn

[3]Postdoctoral Researcher, School of Transportation, Southeast Univ., Nanjing, Jiangsu 210096, China; and Tianjin Bureau of Highway Quality and Safety Inspection, Nankai District, Tianjin 300384, China. E-mail: zym7281@163.com

Note. This manuscript was submitted on March 23, 2014; approved on February 5, 2015; published online on March 30, 2015. Discussion period open until August 30, 2015; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Computing in Civil Engineering***, © ASCE, ISSN 0887-3801/04015021(11)/$25.00.

Automated evaluation of pavement distress involves three steps: image preprocessing, image segmentation, and pavement distress classification and measurements. Image preprocessing is used to reduce noise and to enhance the region of interest. Common image preprocessing techniques are based on median filters (Li et al. 2010; Maode et al. 2007; Mustaffara et al. 2008), histogram equalization (Nejad and Zakeri 2011), and hybrid methods (Gavilán et al. 2011). Image segmentation is the process of partitioning an image into background and distresses. Pavement distress classification and measurements include the identification of distress type and quantification of distress extent and severity.

Image segmentation generally can be grouped into two categories: threshold-based methods and edge detection–based methods. Kirschke and Velinsky (1992) presented an initial effort toward a histogram-based machine vision technique for the automated detection of spontaneous cracks in highway pavement. Koutsopoulos et al. (1993) compared different threshold methods including regression, Otsu, relaxation, and Kittler's method, and showed that the regression-based method outperforms the others but does not perform so well when shadows exist on images. Koutsopoulos and Downey (1993) presented a statistical approach to distinguish crack segmentation. However, the method encounters some difficulty in classifying pavement distress types, mainly between alligator and block cracks. Cheng and Miyojim (1998) applied a neural network to select a threshold for distress segmentation. They used the standard deviation and mean as parameters in neural network training. Cheng et al. (1999) proposed a fuzzy set theory to detect and segment cracks. Hassani and Tehrani (2008) recommended a fuzzy logic-based system for automatic pavement distress detection using a general threshold for detection and segmentation of distresses. He and Qiu (2012) proposed an improved segmentation method based on the combination of multithreshold averaging and multidirectional mathematical morphology. They claimed their proposed algorithm can improve the effects of segmentation significantly and works against noise effectively. Wang and Tang (2012) presented a new method and compared it with the global threshold value algorithm and the local optimal threshold value algorithm. They claimed the new method was better than the global and optimal local threshold values. Kamaliardakani et al. (2014) developed an algorithm to automatically detect sealed cracks in pavement surface images based on a local minimum approach. The experimental results indicate that the developed algorithm has high accuracy and can consistently detect sealed cracks in different environments.

Edge detection based on segmentation methods has the advantage of reducing image size while keeping critical information. A number of edge detectors have been developed over the past three decades, such as Prewitt, Sobel, Roberts, and LOG edge detectors (Davies 2004). Recently, multiresolution-based edge detection at multiple scales has become popular. Subirats et al. (2006) and Zhou (2004) used continuous wavelet transform for pavement crack surface detection. Wang et al. (2008) and Zhou et al. (2006) used a wavelet edge detection procedure for pavement distress segmentation. Chen et al. (2012) and Zuo et al. (2008) developed a segmentation approach using a fractal dimension. These methods are efficient, but very expensive and slow. Hong et al. (2010) used wavelet transform and pseudocoloring to detect the cracks and apply a Radon transform on the binary image to classify and evaluate the cracks. (Wei et al. 2010; Ying and Salari 2010) presented a novel asphalt pavement crack detection algorithm based on beamlet analysis. The length, orientation, and location of cracks can be detected. Moghadas Nejad and Zakeri (2011) used multiresolution texture analysis for the fast isolation cracks and potholes distress from pavement images. It offers a comprehensive analysis of distress isolation and detection algorithms using three sets of wavelet, ridgelet, and curvelet-based texture vectors. Their result indicates that using curvelet-based texture features can improve the classification of potholes, and ridgelet-based texture features can improve the classification of cracking distress. Wu and Liu (2012) presented a method based on wavelet analysis to remove noise. Their proposed method is suitable for real-time detection. Zhibiao and Yanqing (2013) developed an algorithm based on the contourlet domain. Simulated results demonstrate that the algorithm has a good robustness. Tsai et al. (2009) used a scoring measure to compare the performance of six different pavement distress segmentation algorithms: statistical/relaxation threshold, Canny edge detection, multiscale wavelets, crack seed verification, iterative clipping, and a dynamic optimization–based method. It is concluded that the dynamic optimization–based method is the best among these six methods, but it does require a long computation time.

In summary, several segmentation methods have been proposed over the last decade for detection and segmentation of pavement surface images with cracking. Among existing methods, some have suffered from unstable performance in terms of accuracy, while others require a long computation time that is not suitable for online processing.

The remainder of the paper is organized as follows. "New Crack Detection Algorithm" section introduces the proposed algorithm for pavement cracking segmentation. "Pattern Selection" section presents pattern selection procedure through experimental study using the ground truth pavement images with cracking. "Performance Evaluation of Proposed Method" section evaluated the performance of the proposed method against artificial detection of the ground truth pavement images with cracking. Conclusions are presented in the last section.

New Crack Detection Algorithm

Fig. 1 outlines the proposed new algorithm for pavement cracking detection. The method is built upon two assumptions: (1) crack pixels are darker than the surrounding background; and (2) cracks are long linear or thin rectangular regions (Chambon and Moliard 2011).

Input Image

The gray pavement surface images collected from the interstate highway in Maryland with a frame size of 3,480 × 2,048 are considered as the input.

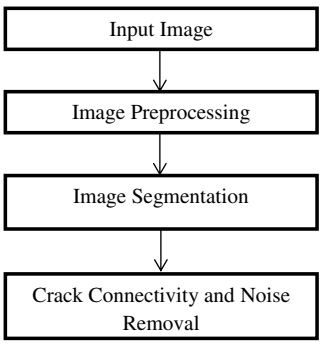

Fig. 1. Flowchart of the proposed algorithm

Image Preprocessing

A pavement image is composed of background, noise, and probably distresses. Cracking is part of the distress. An obstacle to automatic pavement distress detection and segmentation is that the road pavement images are often obtained under nonuniform distributed lighting conditions. In order to overcome this obstacle, preprocessing is an essential step of image analysis and pattern recognition. It is necessary to convert all source images to a uniform lighting condition; preprocessing corrects the average brightness level of each image column. At the beginning of the process, the gain and the exposure time of camera is adjusted individually to a average pixel value of 128 (the middle of 0 and 255). Nevertheless, brightness measured along a given line is not constant due to the fact that the lighting and viewing conditions are not exactly the same at every point (Zalama et al. 2013). In order to correct existing differences, the following process has been suggested

$$I'_{ij} = \frac{128}{A_i} I_{ij} \quad i = 1, 2, \ldots, M \quad j = 1, 2, \ldots, N \tag{1}$$

where I_{ij} = pixel value of the original image at column i and row j; I'_{ij} = transformed image pixel value; M = number of columns of the image; and N = number of rows. Moreover, A_i = average pixel value of column i. Then, a 5×5 median filter is used to reduce salt-and-pepper noise. Finally, a 3×3 min filter is used twice to extend dark objects such as cracks.

Figs. 2 and 3 depict two examples of the preprocessed images. Fig. 2(a) is a pavement image which includes a horizontal crack with stripe noise. Fig. 2(b) shows average pixel value in the X-direction that changes at the different locations, which indicates that the background illumination is not constant. In order to extract the crack information from the original image, it is crucial to convert the background component into a constant base intensity. Fig. 2(c) is the image after brightness level correction with uniform background in the X-direction, which can be seen in terms of histogram in Fig. 2(d). From comparing Figs. 2(b and d), it can be seen that brightness level correction provides a uniform distribution for illumination of the image. Finally, Figs. 2(e and f) show the results of the preprocessed image and correspondent histogram after applying the median filter and min filter. Fig. 3 shows the magnified image of the preprocessing procedure, and it can be seen that crack features are enhanced and the appearance of the noncrack area is smoothed.

Image Segmentation

Image segmentation is a technique for partitioning an object of interest (i.e., cracks) from the background. The intensity thresholding approach is widely used for the detection process. The performance of three image segmentation algorithms for noisy pavement surface images is evaluated. The first method is iterative clipping (Oh et al. 1998; Tsai et al. 2009); the second method is weighted mean–based thresholding (Lokeshwor et al. 2013). The third method is weighted neighborhood pixel thresholding.

Iterative Clipping

Tsai et al. (2009) evaluated the performance of this algorithm for the segmentation of noisy road surface images. In this algorithm, images are first divided into eight subimages. Then, clipping values are determined for each subimage by calculating mean values. All pixel values above the clipping value are set to a clipping value and the rest values are unchanged. After that, the mean and standard deviation of each clipped tiles are calculated to determine a corresponding new clipping value

$$C^n = \mu^{n-1} - (1.2 \times S^{n-1}) \tag{2}$$

where C^n = clipping value at the nth iteration; μ^{n-1} = mean value at $(n-1)$th iteration; and S^{n-1} = standard deviation value at $(n-1)$th iteration. The iteration is stopped when the difference between current clipping computed by Eq. (2) and the mean after the clipping is less than 0.3. Finally, threshold values are defined by the mean value minus five.

Weighted Mean-Based Thresholding

Lokeshwor et al. (2013) presented an adaptive thresholding method for segmentation of noisy road surface images. In these methods, the threshold level is defined on a pixel-by-pixel basis by computing an equally weighted mean of a neighborhood area around each pixel location minus a constant value to be subtracted from the mean. The key steps of the applied adaptive thresholding technique include the subsequent:

1. Compute equally weighted mean (m) of pixel values over a local window size, i.e., neighborhood area around the current pixel in image.
2. Set a threshold level for the current pixel using $T(i, j) = m(i, j) - C$, where (i, j) are the coordinates at the current pixel and C is a constant value.
3. Convert the current pixel value (I) into binary using the following condition:

$$I(i, j) = \begin{cases} 255 & \text{if } I(i, j) \le T(i, j) \\ 0 & \text{if } I(i, j) > T(i, j) \end{cases} \tag{3}$$

4. Repeat until the last pixel in the image undergoes steps 2–4.
5. End.

Feature Extraction and Using Low Threshold

The proposed weighted neighborhood pixels method for feature extraction of pavement image with cracks is illustrated in Fig. 4, in which the letter C means central pixel. A pixel can have 8, 16, and 24 neighbors in the first, the second, and the third surrounding loops, which are represented by the star, dash, and cross symbols in this figure, respectively. The surrounding loops can go on to a large number. A weight assigned to a pixel is calculated based on the pixel values of its surrounding pixels according to Eq. (4)

$$w = \frac{G_{\text{total}}}{G_{\max}} \tag{4}$$

where w = neighborhood pixels weight; G_{total} = sum of all pixel values in three surrounding loops; and $G_{\max}$ = maximum achievable pixel value of all pixels in three surrounding loops.

Different distresses show different patterns in pavement surface images. Cracking has its own characteristics as well. For this reason, a good feature-extraction segmentation method should be calibrated to a specific type of pavement distress (i.e., cracking in this case) to achieve the best performance. To investigate which neighborhood pixels weight works the best for feature extraction, seven different three-loop patterns shown in Fig. 5 are investigated, in which Fig. 5(g) was used by Kumar (2010). The quantitative evaluation shown in the eight-direction pattern is the best three-loop pattern among all patterns that will be selected as the final pattern of the weighted-neighborhood-pixel feature extraction.

Fig. 6(a) shows a three-loops window of Fig. 5(f). The value of intensity of all first-loop pixels is $255 + 255 + 76 + 77 + 81 +$

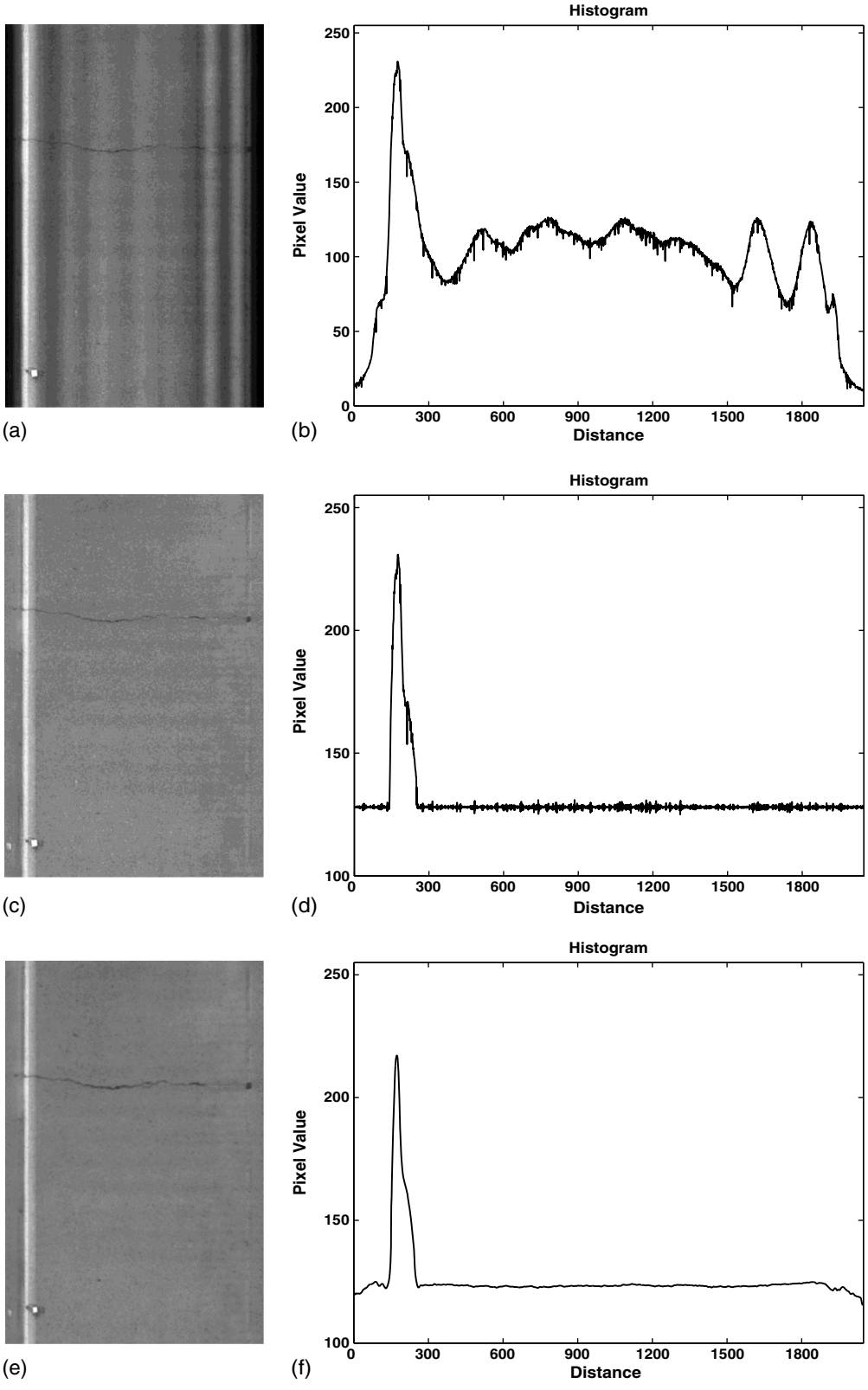

Fig. 2. Enhancement of nonuniform pavement image and corresponding histogram: (a) original image A with cracks; (b) average pixel value plot in X-direction of the original image A; (c) measured brightness levels correction A; (d) average pixel value plot in X-direction of the brightness level correction A; (e) result of preprocessed image; (f) average pixel value plot in X-direction of the preprocessed image A

$72 + 255 + 255 = 1{,}326$. This value for the second and third loop pixels is 1,501 and 1,307, respectively. As a result, the nominator in Eq. (4) is $1{,}326 + 1{,}501 + 1{,}307 = 4{,}134$. Since there are 24 pixels that construct the pattern in Fig. 5(f), the maximum pixel value of all pixels that can be achieved in all three surrounding loops is $24 \times 255 = 6{,}120$. Thus, the weight for this three-loop window is given by $4{,}134/6{,}120 = 0.68$, which is summarized in Fig. 6(b).

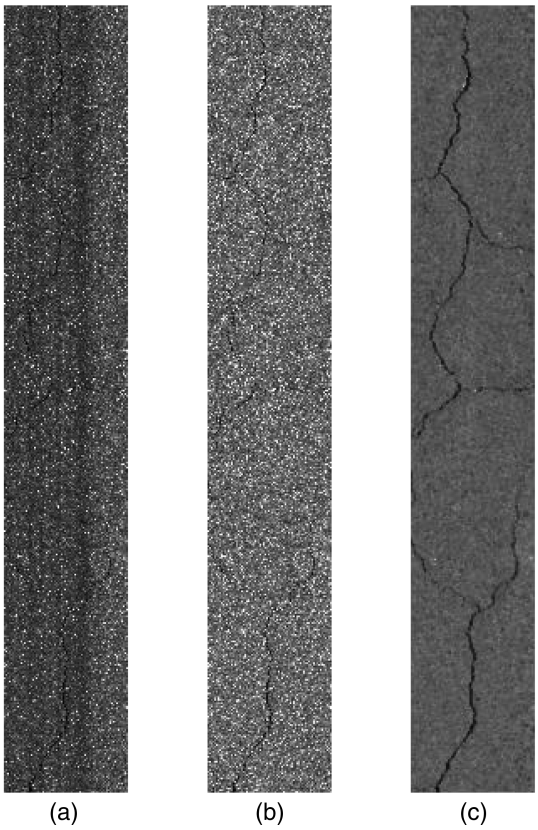

Fig. 3. Illustration of magnified preprocessed image: (a) original image B with cracks; (b) measured brightness levels correction B; (c) result of preprocessed image B

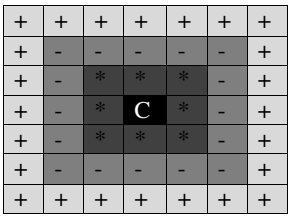

Fig. 4. The neighborhood pixels of a three-loop window

The image obtained at the end of the feature extraction is converted to a binary image. All pixels have values of one, except object or noise pixels with zero values. In this approach, the binary candidate cracks image can be obtained by selecting the lower threshold. A local sliding window is used at all the pixel locations of the preprocessed image. The average pixel value (I_{ave}) of the local window is compared with the pixel value of the weighted pixel (w_c). The condition for threshold is

$$I_{\text{avg}} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} I(i,j) \tag{5}$$

$$w_c \leq 0.85 \times I_{\text{avg}} \tag{6}$$

where $I(i,j)$ = pixel value of the preprocessed image at row i and column j; M = number of rows; and N = number of columns regarding the sliding window. If the weighted pixel can satisfy the condition in Eq. (6), the value of 1 will be assigned, as a candidate crack pixel. Otherwise, the pixel will be assigned 0 as a background pixel.

Image Segmentation Using a Higher Threshold

After applying the low threshold to the pavement images, the binary candidate crack images are obtained. Due to image noise or pixel brightness, some of the pixels detected by the low threshold may not represent actual cracks. Detected pixels may correspond to the background or other dark objects pixels on the pavement. To overcome this difficulty, a higher threshold is taken into account. The proposed image segmentation process includes the following three steps. Segmentation process steps are demonstrated in Fig. 7.

1. Noise removal: According to assumption (2), cracks often appear as a long line or thin rectangle regions, whereas noncrack regions appear as irregular groups of pixels. Therefore, these irregular noise regions can be filter out by examining the eccentricity and area values. The range of the eccentricity parameters is from 0 to 1, where a value close to one indicates a line and a value close to zero indicates a circle. Therefore, the fragments with an eccentricity value less than 0.9 can be considered as noncrack objects or noise and removed.
2. Block labeling: In this step, the binary candidate crack image is divided into rectangular blocks (here the image is divided into 8 by 8 blocks). Then, each block containing cracks will be assigned a specific label.
3. Applying higher threshold to the pavement images block: After finding the contour of the cracks, the original image will be divided into same-sized rectangular blocks. Then, the labeled block image and its neighborhood are considered, and Eq. (6) is applied with new coefficient (here it is 0.9). The candidate cracks with a higher threshold are separated from the background.

Crack Connectivity and Noise Removal

The actual cracks with higher intensity than the specified threshold may filter out in the segmentation process. Subsequently, segmented images are fragmented and disconnected. Also, the background with lower intensity than the specified threshold demonstrates the small single points that represent image noise. Segmented images are enhanced in the following two-step process: gaps between two close fragments are connected by means of dilation operation, and noises are then removed to prevent failing detection in the next steps.

Crack Connectivity

Connecting gaps is essential to connect the crack fragments that are often observed after the segmentation process. These connecting gaps can be obtained using a morphological dilation operation. The dilation operation grows objects in a binary image. With A and B as sets in $\mathbb{Z}^2$, the dilation operation is defined as

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\} \tag{7}$$

where $A \oplus B$ is dilation of image A by structure element (SE) B. This equation is based on reflecting B about its origin and shifting this reflection by z (Gonzalez and Woods 2008). Selecting the shape and size of the SE depends on the former information about the characteristics of the image features that are required to be extended. If the SE size is chosen very large, it may cause close single noise to be merged, and if it is too small the fragments of cracks may not connect properly. The result from investigating several

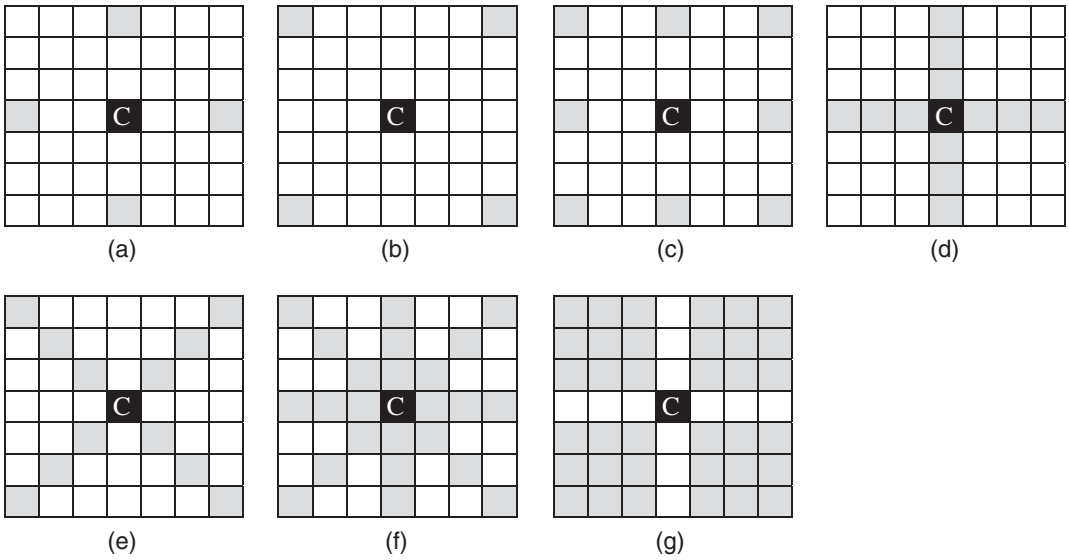

Fig. 5. Seven different patterns used for assigning neighborhood pixel weights: (a) four main far pixels; (b) four partial far pixels; (c) eight-direction far point pixels; (d) four main-direction pixels; (e) four partial-direction pixels; (f) eight-direction pixels; (g) Kumar pattern

+65	77	255	255+	74	255	255+
255	255-	255	255-	80	67-	255
255	255	255*	255*	76*	79	255
+74	-78	77*	255	81*	81-	255+
255	69	72*	255*	255*	255	255
255	-255	255	255-	255	255-	255
+255	64	255	73+	255	75	75+

(a)

Loop	Sum of all pixel values	Weight
First loop	1326*	0.65
Second loop	1501-	0.74
Third loop	1307+	0.64
Total	4134	0.68

(b)

Fig. 6. Pixel value *C* and its surrounding pixels of three loops: (a) pixel values of all three-loop pixels; (b) weight assignment using eight-directions pattern of Fig. 5(f)

images in the database shows that cracks often appear as a line or as a thin rectangle, and gaps typically are at distance of less than approximately 10 pixels; consequently, a line SE with length of 10 pixels is used for the dilation operator. The use of a dilation operator with a 10-pixel line as SE oriented to a specific angle would allow growing of crack segments along this angle. The orientations are assumed to range between −90° and 90° in 22.5° increments. Fig. 8(a) shows the shape and orientation of the structure element for the dilation operation.

The dilation operation is implemented to the segmented images by covering the total range of the directions with an increment of 22.5°. Eight binary images are created by repeating the dilation operator for each orientation. Each binary image presents a particular direction. Finally, eight binary images are added together to form the output of the connected candidate crack. Fig. 8(b) demonstrates the result of connecting gap for the example images by applying dilation operation. Either dilation can be used as crack connectivity in this case. However, the use of dilation also provides unwanted excessive merging of noises that are close to each other.

Noise Removal

In pavement surface images, an area is described by the number of pixels in binary image and pixel size (mm^2). The AASHTO Cracking Protocol PP44-01 (AASHTO 2005) states that areas less than 75 mm^2 (equivalent to an area of 3 mm in width by 25 mm in length) are not treated as a crack. Therefore, small areas less than 75 mm^2 are treated as isolated objects and small clusters of pixels are indicated for removal. They are then removed using the forward-scan two-pass connected-component labeling (CCL) algorithm employing the union-find data structure (Hernandez-Belmonte et al. 2011; Wu et al. 2008, 2009) with a set minimum cut off of 75 mm^2.

CCL algorithms include multipass, two-pass, one-pass, parallel, and tracing-type algorithms. These algorithms try to relabel component pixels according to an equivalence relation among temporary labels via different connected masks (Fig. 9). In this paper, a two-pass CCL algorithm consisting of three phases (i.e., scanning, analyzing, and labeling) is used over a binary image. The aim of the

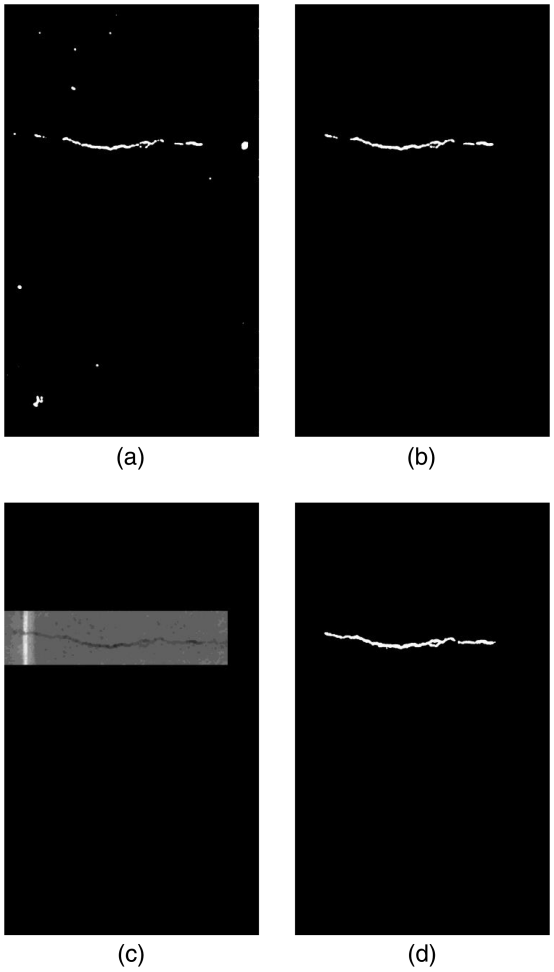

Fig. 7. Step-by-step image segmentation process: (a) after applying a low threshold; (b) output of the shape filtering; (c) block labeling which includes cracks; (d) segmented image after applying a higher threshold

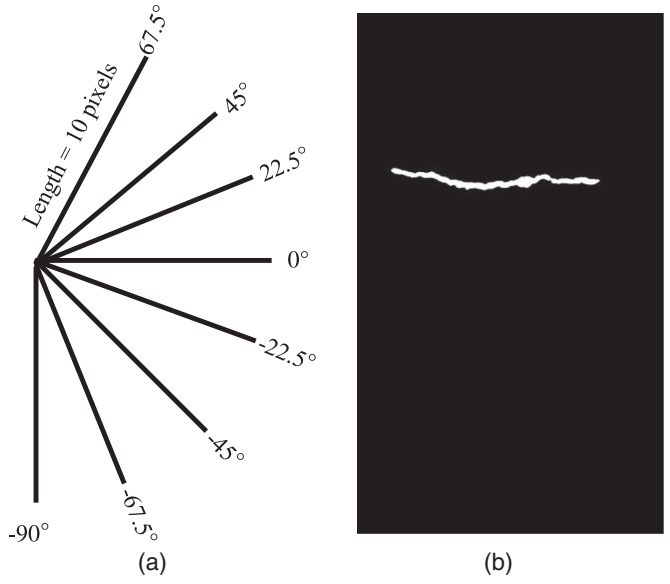

Fig. 8. Dilation operation: (a) orientation of the line structural element with length of 10 pixels for the dilation operation in the connectivity process; (b) result of gap-filling and connecting

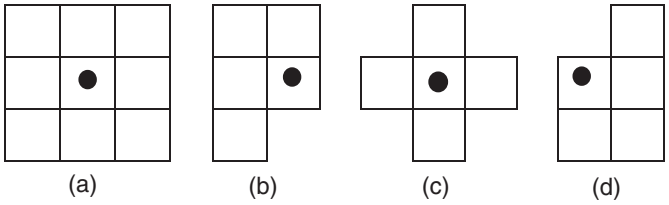

Fig. 9. Shapes of neighborhood masks used for connectivity analysis in CCL algorithms: (a) eight-connected mask; (b) forward-scan mask; (c) four-neighbor mask; (d) backward-scan mask

scanning phase is to assign temporary labels to all object pixels and store the equivalence information about the temporary labels based on the values of neighbors. The analyzing phase analyzes the label equivalence information to determine the final label of each temporary label. The labeling phase assigns the final labels to the object pixels by doing the second pass through the image.

The union-find data structure is one of the most efficient data structures for representing the equivalence information. A union-find data structure conceptually represents rooted trees, where each node of a tree is a temporary label, and each edge represents the equivalence between two labels. Only three operations are needed on a union-find labeling structure: combining two trees (union operation), finding the root node linked to another node (find operation), and creation of a new tree with a single node. The union operation adds an edge from the root of one tree to the root of another one when there is more than one root. The find operation starts from a node and traces back until it reaches the root node. This operation modifies all the nodes in the path to the root label.

In short, a two-pass labeling algorithm employing a union-find data structure generally starts with a scanning phase by using one of the scanning masks demonstrated in Fig. 9 to examine the image and assign the provisional labels to object pixels. During the scanning phase, it also builds up the union-find data structure to record the equivalence information among the provisional labels. After the scanning phase, it analyzes the union-find data structure to determine the final label for each provisional label, a stage called the labeling phase.

Pattern Selection

Performance and computational time depend upon image size, image resolution, the number of neighborhood pixels, and the number of loops involved in the feature extraction. Performance and CPU time of these seven different patterns are evaluated and compared in this section. All these aforementioned analyses are conducted using *MATLAB* in a ThinkPad T530 laptop with an Intel Core i5 CPU, and 4 GB of RAM.

Fifty pavement surface images with cracks are used in comparison of seven feature extraction patterns. For each image, three different image sizes (2,048 × 3,480, 1,740 × 1,024 and 1,160 × 683) are studied, corresponding to high resolution images, medium resolution images and low resolution images, respectively, indicating the amount of image shrinkage. Proposed patterns were applied to 50 images, and the results detected were compared to the ground truth.

A team of three pavement engineers first manually identified and marked (in red) cracks on these 50 pavement images. One of these pavement images is shown in Fig. 10(b). Such a marked image is defined as a ground truth image for comparison.

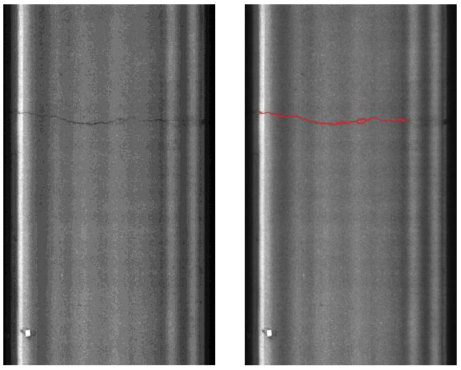

Fig. 10. The definition of original image versus ground truth: (a) original image; (b) ground truth image marked by a team of three pavement engineers

For the quantitative evaluation of crack detection, a type I error is defined as follows; it is used to measure the effectiveness of patterns:

$$\alpha_I = \frac{\text{Number of incorrectly identified pixels of cracks}}{\text{Total number of pixcels of cracks in ground truth image}} \quad (8)$$

where α_I is the type I error, referring to the percentage of pixels not considered as a crack that are in fact a crack in the ground truth image. For the image illustrated in Fig. 10, a type I error is $773/27{,}549 = 2.8\%$, where 773 is the number of pixels of cracks identified incorrectly and 27,549 indicates the total number of pixels of cracks in the ground truth image.

Fig. 11 compares Type I error against seven patterns for three different image sizes. Among all patterns, the eight-direction pattern in Fig. 5(f) has a minimum error. This can be justified due to the fact that most cracks are either in vertical, horizontal, or diagonal directions, and the eight-direction pattern can cover all of them. Therefore, the eight-direction pattern is selected in this study as the calibrated neighborhood pixel weight feature extraction for image segmentation of cracks.

Another important aspect that should be considered is CPU time. It is found that these different patterns do not result in considerably different CPU times. With the eight-direction pattern, CPU times for high-resolution, medium-resolution, and low-resolution images using the eight-direction pattern are 0.813, 0.144, and 0.049 s, respectively. CPU time for low-resolution images is approximately $1/16$ and $1/3$ of CPU time for high-resolution and medium-resolution images, respectively. Since accuracy is more critical in this study, pavement images with high resolution are used for further investigation.

Performance Comparison

To test the robustness of the proposed method, a number of pavement surface images were selected from the collected pavement surface images database. Three different image segmentation methods described in a previous section are applied to the selected pavement surface images. An example of a processed road image using the different image segmentation techniques described in aforementioned sections is illustrated in Fig. 12. The original image, iterative clipping method, weighted mean–based adaptive thresholding, and proposed method are shown in Figs. 12(a–d), respectively.

The results show that the adaptive thresholding and iterative clipping methods were not able to perform well on a noisy road image as compared with the proposed method. This was mostly due to the presence of nonuniform illumination, and also because those methods do not take into account the geometric characteristics of cracks. In such situations, these techniques were not found to be useful for automated segmentation of road surface distress.

Performance Evaluation of Proposed Method

A sample of 1,669 gray images of pavement surface has been manually collected from two different sections of asphalt concrete pavement (i.e., Sections 1 and 2) on an interstate highway in Maryland.

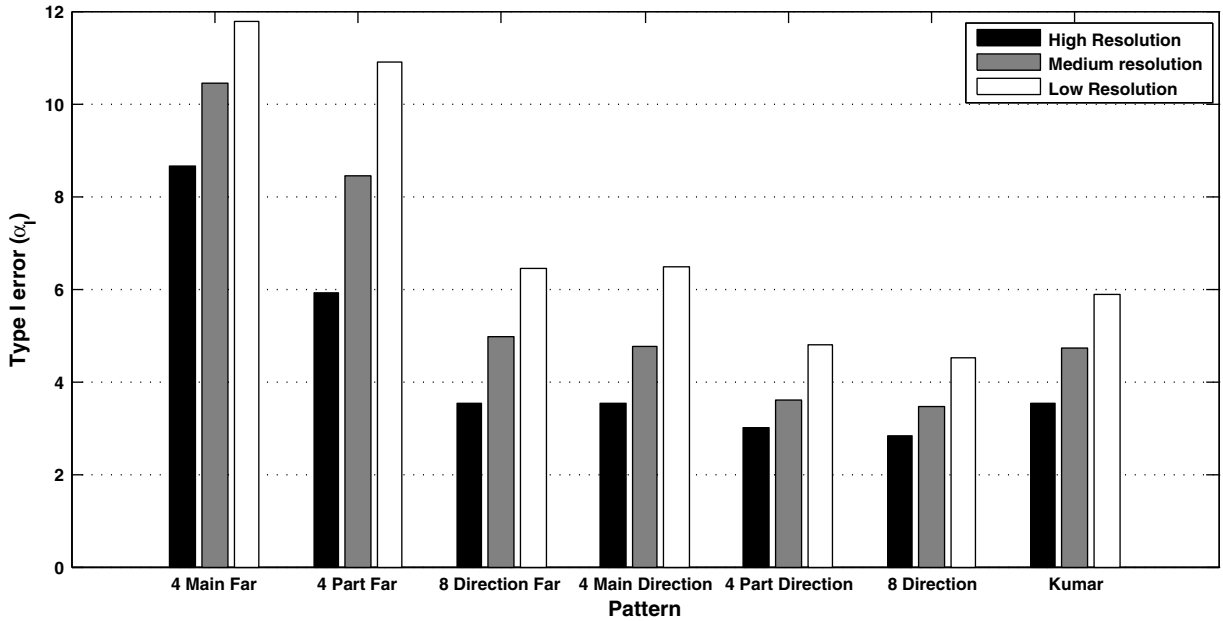

Fig. 11. Type I error of different patterns for high-resolution, medium-resolution, and low-resolution images

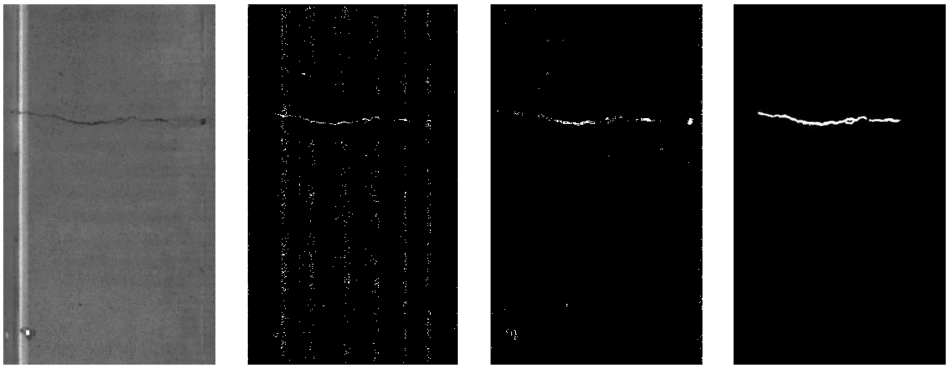

Fig. 12. Comparison performance of image segmentation techniques in road image: (a) pavement surface image; (b) iterative clipping; (c) weighted mean thresholding; (d) proposed method

Out of these images, Section 1 has 48 cracks and Section 2 has 36 crack images. Each image covers approximately a 3.5-m width by 6-m length of pavement surface with a frame resolution of $2{,}048 \times 3{,}480$ pixels. Each pixel represents about 3 mm^2 of pavement area. Pavement surface images with and without cracking are manually labeled by a team of three pavement engineers to obtain the ground truth images for the evaluation of the proposed approach.

The eight-direction pattern of the weighted neighborhood pixel segmentation method is evaluated using 1,669 real pavement surface images with $3{,}480 \times 2{,}048$ size. Precision, recall and performance criterion are used to measure the overall performance of the proposed segmentation method. These quantities are defined below, in which P, R, and PC stand for precision, recall, and performance criterion, respectively

$$P = \frac{\text{Number of pixels correctly classified as cracks}}{\text{Total number of pixcels of detected cracks}} \tag{9}$$

$$R = \frac{\text{Number of pixels correctly classified as cracks}}{\text{Total number of pixcels of detected cracks in ground truth image}} \tag{10}$$

$$PC = \frac{2 \times p \times r}{p + r} \tag{11}$$

In order to have a quantitative assessment of the proposed algorithm for frame identification, metrics such as accuracy, precision, and recall are used. Table 1 shows the confusion matrix for the test. The matrix elements include true positive (TP), true negative (TN), false negative (FN), and false positive (FP). The element TP presents correct detection of crack frames, TN represents correct detection of noncrack frames, FN represent incorrect detection of crack frames (i.e., prediction no crack detected for actual crack instances), and FP represents incorrect detection of noncrack frames.

Accuracy indicates how many images with distress and without distress are identified correctly. Precision indicates how many of the total positives (both true positives and false negatives) are correctly identified as frames with distress. Recall indicates how many frames with distress are classified correctly. Average time per frame is obtained by total time over total frames

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{12}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{13}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{14}$$

$$\text{Average time} = \frac{\text{Total time}}{\text{Total frames}} \tag{15}$$

$\text{TP} = 45$, $\text{TN} = 754$, $\text{FP} = 7$, and $\text{FN} = 3$ with total frames of 809 and $\text{TP} = 35$, $\text{TN} = 819$, $\text{FP} = 5$, and $\text{FN} = 1$ with total frames of 860 for Sections 1 and 2, respectively. Performance metrics calculated using Eqs. (12)–(14) for Section 1 were found to be 98%, 86%, and 94%, and for Section 2 resulted in 99%, 88%, and 97%, respectively.

Table 2 lists the result of crack detection using the selected pattern, including accuracy and average CPU time of all images for each pavement section. Overall, it achieves 87% precision for pavement Section 1 and 92% for pavement Section 2. In terms of recall, it achieves 92% and 97% for pavement Sections 1 and 2, respectively. The speed and accuracy are analyzed for the manual and automated detection frame identification and the results are presented. Accuracy and average time per image for each section are also displayed. Note the processing times vary for each image. For instance, in Section 1 image 24 takes 0.57 s processing time, while it takes 360 s for the manual system. The last two columns show precision and recall criteria for frame identification. It is evident that the proposed method is able to detect cracks with accuracy up to 98% in a much shorter time than manual detection.

In Table 2, average time per image in seconds is calculated by this formula: Time per image (s) = Total time (minute) $\times$ 60 s/ Number of image with crack. For example, in Section 1 total manual time for 48 images with cracks is 260 min, which according to the formula for average time per image is 260 min $\times$ 60 s/48 = 325 s.

Table 1. Confusion Matrix Showing Relationships between Frame with or without Distress and Frame Identification

Events	Frame with distress (positive)	Frame without distress (negative)
Correctly frame identified (true)	True positive (TP)	False positive (FP)
Incorrectly frame identified (false)	False negative (FN)	True negative (TN)

Table 2. Performance Evaluation

Sample information			Crack detection			Frame identification					
						Accuracy		Average time per image (s)			
Section number	Number of images with no crack	Number of images with cracks	Precision (%)	Recall (%)	PC (%)	Manual	Automated	Manual	Automated	Precision (%)	Recall (%)
1	761	48	87	92	89	100	98	325	5.18	86	94
2	824	36	92	97	95	100	99	250	4.17	88	97
Total	1,585	84	89	94	92	100	99	293	4.66	87	95

Conclusion

A new image processing method is designed for online automated detection and segmentation of pavement cracking. For image preprocessing, new brightness level correction method is applied to remove nonuniform background. For image segmentation, seven different patterns are studied and compared for their performances. The eight-direction pattern is finally chosen due to its best performance in terms of accuracy and robustness to noise. A corresponding binary image is then defined by applying threshold. Noise regions from the binary image are removed by examining the eccentricity and area values by applying the region properties and the forward-scan two-pass-CCL algorithm using a union-find data structure.

The proposed method was tested on different crack types, and the detection accuracy has been evaluated by comparing the automated detection and segmentation to the ground truth data obtained through manual detection and segmentation. Using the test dataset, the new method identified pavement cracking with 98% accuracy in a fast and robust fashion, allowing for online automated detection and segmentation of pavement surface image with cracking. When affiliated with the geographical location information of pavement surface images, it provides a powerful tool for online automated pavement condition assessment of a highway network.

Acknowledgments

This study is sponsored by Shanghai PPD Transportation Science & Technology, Inc., sponsored in part by the National Science Foundation CAREER Award CMMI-0644552, by the National Science Foundation of China under grants Nos. 51250110075 and U1134206, to which the authors are very grateful. The authors are also very thankful to the anonymous reviewers for their helpful comments and constructive suggestions, which helped us improve the content and presentation of the original manuscript.

References

AASHTO. (2005). "Standard practice for quantifying cracks in asphalt pavement surface." *PP 44-01*, Washington, DC.

Chambon, S., and Moliard, J.-M. (2011). "Automatic road pavement assessment with image processing: Review and comparison." *Int. J. Geophys.*, 2011, 20.

Chen, B., Cao, W., and He, Y. (2012). "Fractal dimension applied in highway surface crack detection." *Proc., 4th Int. Conf. on Digital Image Processing (ICDIP 2012)*, International Society for Optics and Photonics, Bellingham, WA.

Cheng, H., Chen, J. R., Glazier, C., and Hu, Y. (1999). "Novel approach to pavement cracking detection based on fuzzy set theory." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)0887-3801(1999)13:4(270), 270–280.

Cheng, H., and Miyojim, M. (1998). "Novel system for automatic pavement distress detection." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)0887-3801(1998)12:3(145), 145–152.

Cheng, H., Shi, X., and Glazier, C. (2003). "Real-time image thresholding based on sample space reduction and interpolation approach." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)0887-3801(2003)17:4(264), 264–272.

Davies, E. R. (2004). *Machine vision: Theory, algorithms, practicalities*, Morgan Kaufmann Publishers, CA.

Gavilán, M., et al. (2011). "Adaptive road crack detection system by pavement classification." *Sensors*, 11(12), 9628–9657.

Gonzalez, C. R., and Woods, E. R. (2008). *Digital image processing*, Prentice Hall, Upper Saddle River, NJ.

Hassani, A., and Tehrani, H. G. (2008). "Crack detection and classification in asphalt pavement using image processing." *Proc., Rilem 6th Int. Conf. on Cracking in Pavements*, CRC Press, Boca Raton, FL.

He, Y., and Qiu, H. (2012). "A method of cracks image segmentation based on the means of multiple thresholds." *J. Commun. Comput.*, 9, 1147–1151.

Hernandez-Belmonte, U. H., Ayala-Ramirez, V., and Sanchez-Yanez, R. E. (2011). "A comparative review of two-pass connected component labeling algorithms." *Adv. Soft Comput.*, 7095, 452–462.

Hong, L., Salari, E., and Chou, E. (2010). "Pavement information system: Detection, classification and evaluation." *Proc., Electro/Information Technology (EIT), 2010 IEEE Int. Conf.*, IEEE, New York, 1–5.

Kamaliardakani, M., Sun, L., and Ardakani, M. (2014). "Sealed-crack detection algorithm using heuristic thresholding approach." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000447, 04014110.

Kirschke, K., and Velinsky, S. (1992). "Histogram-based approach for automated pavement-crack sensing." *J. Transp. Eng.*, 10.1061/(ASCE)0733-947X(1992)118:5(700), 700–710.

Koutsopoulos, H., and Downey, A. (1993). "Primitive-based classification of pavement cracking images." *J. Transp. Eng.*, 10.1061/(ASCE)0733-947X(1993)119:3(402), 402–418.

Koutsopoulos, H. N., El Sanhouri, I., and Downey, A. B. (1993). "Analysis of segmentation algorithms for pavement distress images." *J. Transp. Eng.*, 10.1061/(ASCE)0733-947X(1993)119:6(868), 868–888.

Kumar, S. (2010). "Neighborhood pixels weights—A new feature extractor." *Int. J. Comput. Theor. Eng.*, 2(1), 69–77.

Li, Q., Yao, M., Yao, X., and Xu, B. (2010). "A real-time 3D scanning system for pavement distortion inspection." *Meas. Sci. Technol.*, 21(1), 015702.

Lokeshwor, H., Das, L. K., and Goel, S. (2013). "A robust method for automated segmentation of frames with/without distress from road surface video clips." *J. Transp. Eng.*, 140(1), 31–41.

Maode, Y., Shaobo, B., Kun, X., and Yuyao, H. (2007). "Pavement crack detection and analysis for high-grade highway." *Proc., Electronic Measurement and Instruments, 2007 ICEMI'07 8th Int. Conf.*, IEEE, New York.

MATLAB [Computer software]. Natick, MA, Mathworks.

Moghadas Nejad, F., and Zakeri, H. (2011). "A comparison of multi-resolution methods for detection and isolation of pavement distress." *Expert Syst. Appl.*, 38(3), 2857–2872.

Mustaffara, M., Lingb, T., and Puanb, O. (2008). "Automated pavement imaging program (APIP) for pavement cracks classification and quantification—A photogrammetric approach." *ISPRS J. Photogramm. Remote Sens.*, 37(B4), 362–372.

Nejad, F. M., and Zakeri, H. (2011). "An optimum feature extraction method based on wavelet-radon transform and dynamic neural network for pavement distress classification." *Expert Syst. Appl.*, 38(8), 9442–9460.

Oh, H., Garrick, N. W., and Achenie, L. E. (1998). "Segmentation algorithm using iterative clipping for processing noisy pavement images." *Proc., 2nd Int. Conf. Imaging Technologies: Techniques and Applications in Civil Engineering*, ASCE, Reston, VA.

Subirats, P., Dumoulin, J., Legeay, V., and Barba, D. (2006). "Automation of pavement surface crack detection using the continuous wavelet transform." *Proc., Image Processing, 2006 IEEE Int. Conf.*, IEEE, New York, 3037–3040.

Teomete, E., Amin, V. R., Ceylan, H., and Smadi, O. (2005). "Digital image processing for pavement distress analyses." *Proc., 2005 Mid-Continent Transportation Research Symp.*, Iowa Dept. of Transportation, Iowa State Univ., Ames, IA.

Tsai, Y.-C., Kaul, V., and Mersereau, R. M. (2009). "Critical assessment of pavement distress segmentation methods." *J. Transp. Eng.*, 10.1061/(ASCE)TE.1943-5436.0000051, 11–19.

Wang, K. C., Li, Q., and Gong, W. (2008). "Wavelet-based pavement distress image edge detection with a trous algorithm." *Transportation Research Record 2024*, Transportation Research Board, Washington, DC, 73–81.

Wang, S., and Tang, W. (2012). "Pavement crack segmentation algorithm based on local optimal threshold of cracks density distribution." *Adv. Intell. Comput.*, 6838, 298–302.

Wei, N., Zhao, X., Dou, X. Y., Song, H., and Wang, T. (2010). "Beamlet transform based pavement image crack detection." *Proc., Intelligent Computation Technology and Automation (ICICTA), 2010 Int. Conf.*, IEEE, New York, 881–883.

Wu, K., Otoo, E., and Suzuki, K. (2008). "Two strategies to speed up connected component labeling algorithms." Lawrence Berkeley National Laboratory, Berkeley, CA.

Wu, K., Otoo, E., and Suzuki, K. (2009). "Optimizing two-pass connected-component labeling algorithms." *Pattern Anal. Appl.*, 12(2), 117–135.

Wu, S., and Liu, Y. (2012). "A segment algorithm for crack dection." *Proc., Electrical and Electronics Engineering (EEESYM), 2012 IEEE Symp.*IEEE, New York, 674–677.

Ying, L., and Salari, E. (2010). "Beamlet transform-based technique for pavement crack detection and classification." *Comput.-Aided Civ. Infrastruct. Eng.*, 25(8), 572–580.

Zalama, E., Gómez-García-Bermejo, J., Medina, R., and Llamas, J. (2013). "Road crack detection using visual features extracted by Gabor filters." *Comput.-Aided Civ. Infrastruct. Eng.*, 29(5), 342–358.

Zhibiao, S., and Yanqing, G. (2013). "Algorithm on contourlet domain in detection of road cracks for pavement images." *J. Algorithms Comput. Technol.*, 7(1), 15–26.

Zhou, J. (2004). *Automated pavement inspection based on wavelet analysis*, State Univ. of New York, Stony Brook, NY.

Zhou, J., Huang, P. S., and Chiang, F.-P. (2006). "Wavelet-based pavement distress detection and evaluation." *Opt. Eng.*, 45(2), 027007.

Zuo, Y., Wang, G., and Zuo, C. (2008). "A novel image segmentation method of pavement surface cracks based on fractal theory." *Proc., Int. Conf. Computational Intelligence and Security 2008*, IEEE, New York, 485–488.