MICHIGAN OHIO UNIVERSITY TRANSPORTATION CENTER
Alternate energy and system mobility to stimulate economic development.

Report No: MIOH UTC TS18p2 2010-Final

Transportation Informatics: Advanced Image Processing Techniques for Automated Pavement Distress Evaluation

FINAL REPORT

PROJECT TEAM

Dr. Ezzatollah Salari
Dr. Eddie Yein-Juin Chou
Dept. of Electrical Engineering & Computer Science
College of Engineering
The University of Toledo
Toledo, OH 43606

Dr. James Lynch
Dr. Utpal Dutta
Dept. of Civil, Architectural & Environmental Engineering
College of Engineering & Science
University of Detroit Mercy
Detroit, MI 48221

Report No: MIOH UTC TS18p2 2010-Final
TS18 Project 2 – the second of a three project series
FINAL REPORT

Developed By:

Dr. Ezzatollah Salari
Principal Investigator
Esalari@utnet.utoledo.edu

Dr. Eddie Yein Juin Chou, P.E.
Co-Principal Investigator
ychou@utnet.utoledo.edu

Dr. James Lynch
Co-Principal Investigator
lynchjj@udmercy.edu

Dr. Utpal Dutta
Co-Principal Investigator
duttau@udmercy.edu

SPONSORS

This is a Michigan Ohio University Transportation Center project supported by the U.S. Department of Transportation, The University of Toledo, and the University of Detroit Mercy.

DISCLAMER

Transportation Informatics: Advanced Image Processing Techniques for Automated Pavement Distress Evaluation

TS18 Project 2 – the second of a three project series

ABSTRACT

The current project, funded by MIOH-UTC for the period 1/1/2009- 4/30/2010, is concerned with the development of the framework for a transportation facility inspection system using advanced image processing techniques. The focus of this study is on the technical details of investigating and utilizing state-of-the-art image analysis techniques to further advance research in image processing based inspection systems in order to detect and classify the cracks in pavement. The detection of cracks and other degradations of pavement surfaces has traditionally been done by human experts conducting visual inspection while driving along the surveyed road.

This manual approach is not only time consuming but also costly and subjective. To overcome these limitations we developed two different approaches for automatic crack detection and classification to speed up the process and reduce subjectivity. In the first approach, after the pavement images are captured by a digital camera, regions corresponding to cracks are detected over the acquired images by local segmentation and then represented by a matrix of square tiles. Since the crack pattern can be represented by the distribution of the crack tiles, standard deviations of both vertical and horizontal histograms are calculated to map the cracks onto a 2D feature space, where four crack types can be identified as: longitudinal cracks, transversal cracks, block cracks and alligator cracks. This new technique provides a low-cost, near real time distress analysis option. In the second approach we explore the use of a more robust multi-resolution scheme based on the beamlet transform. This method uses a pavement distress image enhancement algorithm to correct the non-uniform background illumination by calculating the multiplicative factors that eliminate the background lighting variations.

To extract the linear features such as surface cracks from the pavement images, the image is partitioned into small windows and a beamlet transform based algorithm is applied. The crack segments are then linked together and classified into four types, vertical, horizontal, transversal, and block types. Simulation results show that the method is effective and robust in the extraction of cracks from a variety of pavement images. The experimental results, obtained by testing real pavement images over local asphalt roads, present the effectiveness of our algorithm for automating the process of identifying road distresses from images.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. Imaging Technologies in Pavement Distress Analysis

The need for a fast, objective, and relatively inexpensive automated road inspection system is highly desirable. Video technologies and image processing techniques used as tools for inspection, monitoring, and diagnosis have long been adopted by various fields, most notably in medicine and remote sensing. However, its use in transportation is still not widespread. The reason is that the amount of data to be processed could be very large and repetitive, while the accuracy requirements may not be as strict as in, say, medical diagnosis. Conventional visual and manual pavement distress analytical approaches in which the inspectors traverse the roads and stop and measure the distress objects are very costly, time-consuming, labor-intensive, and unstable. Therefore, automated analysis and pattern recognition is highly desirable for pavement inspection. In general, the desired approach is to capture pavement images using video cameras mounted on a moving vehicle and then to use a computer to recognize and quantify the pavement distresses from these video images.

There has been a significant amount of research during the past two decades in developing automated pavement inspection. Xu and Huang developed a customized image processing algorithm for pavement cracking inspection in which an image is divided into small cells and a cell is classified as either crack or non-crack seeds based on its local characteristics [1]. After verification, a cluster of seeds is identified as a real crack. Maser [2] used histogram equalization to improve the contrast of the images, and proposed a threshold-based segmentation. Li [3] used Sobel edge detectors and modified the automatic threshold determination method suggested by Kittler and Illingworth, in order to connect the crack segments to form a continuous cluster of object pixels. They relied on the assumption that noise clusters had a perimeter of less than a pre-specified number of pixels.

Koutsopoulos [4] proposed a lighting variations compensation method by subtracting an average of a few non-distress images from the same series. For segmentation, instead of using ordinary binary segmentation that assigns a value of one to object pixels and a zero value to background pixels, resulting in a binary image, a different approach is suggested, and it assigns values from 0 to 3 to each pixel, based on its probability of being an object pixel. Background pixels are drawn from the Gaussian distribution. Object pixels are drawn from a similar distribution with a lower mean and a higher variance. A threshold that meets various criteria can be obtained from these two distributions. Chou et al. [5] used moment invariants from different types of distress to obtain the features, and then used a back propagation neural network to classify the features. Georgopoulos et al. [6] proposed a method in which the distress can be represented by a set of vectors, approximating the cracks composing the distress. The direction vectors are then grouped into two categories, horizontal and vertical, leading to the classification of the cracks.

Cheng et al. [7] proposed an approach based on fuzzy set theory. First, the proposed method compares the darkness of the pixels and their neighbors by deciding the brightness membership function for gray levels in the difference image. The fuzzified image is mapped into the crack domain by finding the crack membership values of the pixels. Finally, the connectivity of the darker pixels is checked to eliminate the pixels lacking the connectivity property. An image projection algorithm is employed to classify the cracks.

The use of a discrete wavelet transform (DWT) has also been explored for crack image analysis. Using a fast wavelet transform, a pavement image can be decomposed into different frequency sub-bands. The magnitude of the wavelet coefficients represents the level of distress [8-10]. Javidi et al.[11] defined two wavelets which are, respectively, the partial derivatives along x and y of a two-dimensional smoothing cubic spline wavelet function. By measuring the evolution across scales of the wavelet transform maxima, the background noise can be separated from the crack pixels. The crack map images are then projected into the Hough transform domain to quantify the number of dominant cracks in a given image.

Though many attempts [12-14] have been made to automatically collect the pavement crack data, due to the non-uniform illumination effect and irregularities of the pavement surface, limited success was achieved in accurately detecting cracks and classifying the crack types. In addition, most existing systems require complex algorithm with high levels of computing power, thus better approaches are still needed to optimize the process.

2. Methodologies

Pavement distress refers to visible imperfections on the surface of pavements due to overloading, environmental conditions, and normal wear. Often the distresses are present in the form of surface cracking. Four common pavement crack types are illustrated in Figure 1 which includes (a) transversal crack, (b) longitudinal crack, (c) block crack and (d) alligator crack, forming the target of our classification system.

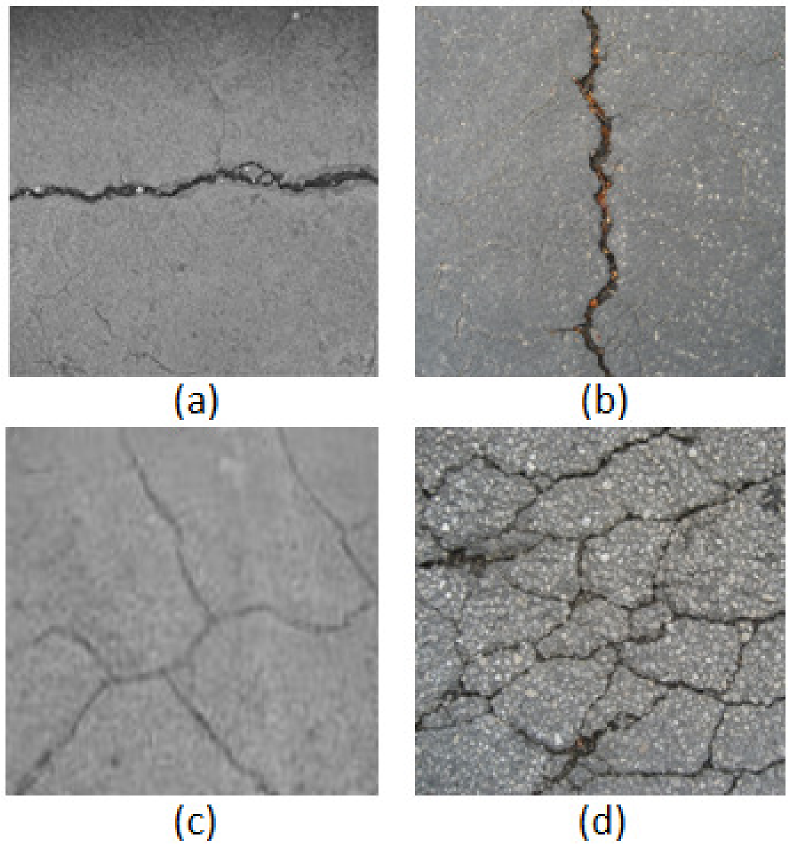


Figure 1. Types of pavement cracks

In this report, we present a reliable automated pavement distress evaluation system capable of detecting cracks as well as classifying crack types from digital pavement images. The proposed model consists of two major parts: crack detection and crack classification.

For the crack detection function, a non-illumination effect removal filter followed by a local segmentation method with multiple threshold values were applied to the acquired image to provide a more accurate binary cracking contour. The binary image is then divided into sub-images called "tiles", each of which stands for either crack or background. Based on the distribution of these crack tiles, the crack type is then identified after mapping to a 2D feature space. This tile-based system could significantly reduce the computational complexity relative to pixel-based approaches. In addition, it is less sensitive to background noise since a few noise pixels alone will not be sufficient for a tile to be classified as a crack tile. As a result, the proposed algorithm is able to provide more reliable classification results within a reasonable period of time. An overall flow chart of the proposed algorithm is shown in Figure 2.

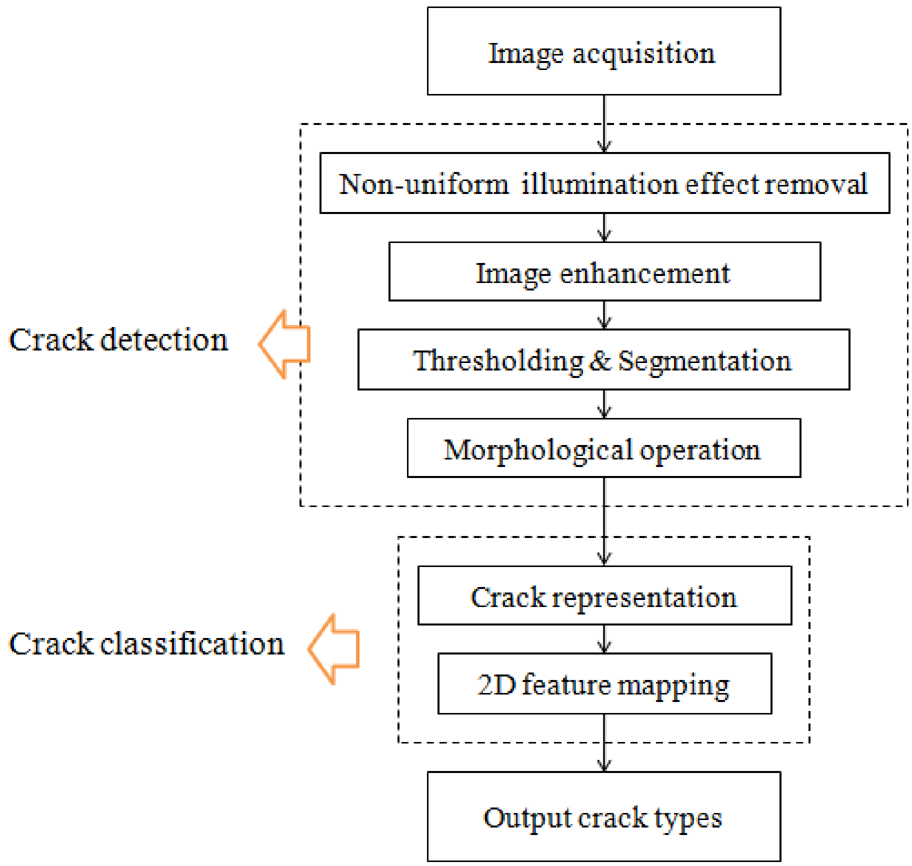


Figure 2. The Overall Follow Chart of the Proposed Model

2.1. Image Analysis Techniques for Crack Detection

2.1.1. Removal of Non-uniform Illumination Effects

In order to provide a more uniform background from an observed pavement image, a background subtraction method based on morphological operations is introduced in this section, ensuring the same background lighting condition for all the pavement images.

First a morphological opening is applied by performing a local minimum filter (erosion) followed by a local maximum filter (dilatation) with the same structuring element. This allows the elimination of small objects (cracks in the foreground) while preserving the large ones (the background). In this way the non-uniform illumination of the background can be successfully extracted. Then, by subtracting the background from the original gray image, the non-uniform illumination effect can be fully removed. Following this step, we add the average intensity value of the background back to every pixel of the image to obtain a pavement image with more uniformly distributed background. Figure 3 shows the details of the above procedure.

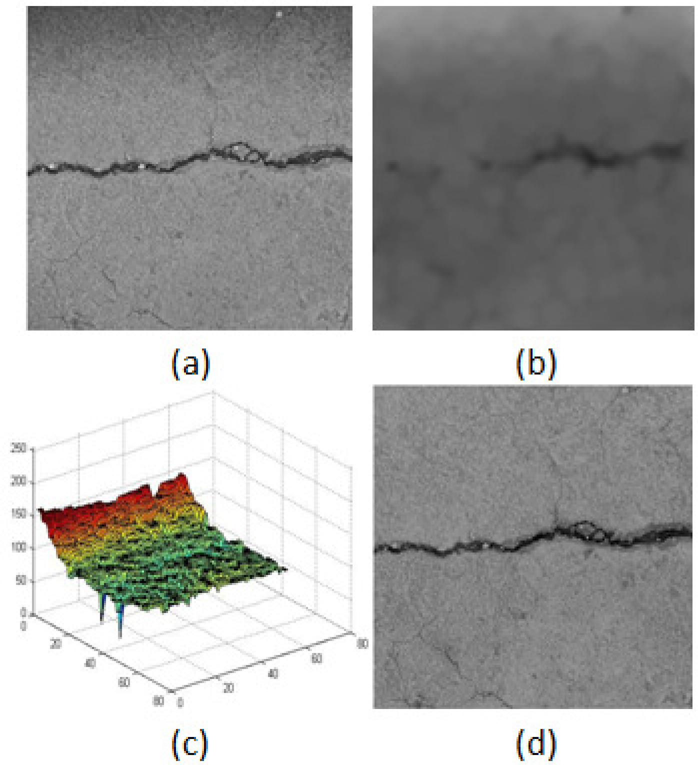


Figure 3. (a) Original pavement image with non-uniform illumination of background
(b) Non-uniform illuminated background extracted by morphological opening
(c) Intensity surface of the background in 3D space
(d) Pavement image after removal of non-uniform illumination effect

2.1.2. Image Enhancement by Wavelet Denoising

Wavelet denoising attempts to remove the noise present in the gray image while preserving the edge sharpness. The common approach to wavelet based denoising is to transform the signal into the wavelet domain, shrink the detail coefficients and transform back to the image domain. A more precise explanation of the wavelet denoising procedure can be given as follows. Assume that the observed data is:

$$X(t) = S(t) + N(t) \tag{1}$$

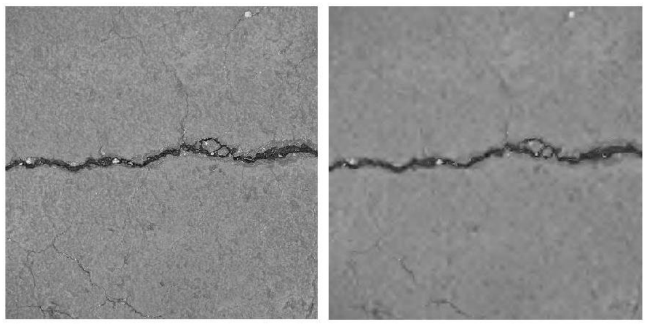
where *S(t)* is the uncorrupted signal with additive noise *N(t)*. Let W and W^{-1} denote the forward and inverse wavelet transform operator, $D(\sim,\lambda)$ denote the denoising operator with threshold λ. We intend to denoise *X (t)* to recover $\hat{S}(t)$ as an estimate of *S (t)*. The practice of thresholding method for denoising consists of the following three steps:

$$\begin{aligned} Y &= W(X) \\ Z &= D(Y,\lambda) \\ \hat{S} &= W^{-1}(Z) \end{aligned} \tag{2}$$

The soft thresholding operator $D(Y,\lambda)$ is defined as:

$$D(Y,\lambda)=\begin{cases} sign(Y)(|Y|-\lambda) & if \quad |Y|>\lambda \\ 0 & otherwise \end{cases} \tag{3}$$

An appropriate choice of the threshold value λ is fundamental to the effectiveness of the described wavelet de-noising procedure. A large threshold might remove important parts of the underlying signal whereas too small threshold retains noise in the reconstruction. Figure 4 shows the result of image denoising in the wavelet domain.



(a) **(b)**

Figure 4. (a) Image before wavelet denoising

(b) Image after wavelet denoising

In addition, with an increasing level of noise in pavement images, we have developed a multi-stage median filter which contains 4 masks (as shown below in Figure 5) as an alternative filtering scheme with limited success.

1	1	1

	1	
	1	
	1	

		1
	1	
1		

1		
	1	
		1

Figure 5. Four Masks Used for Median Filtering

2.1.3. Thresholding & Segmentation

The selection of an appropriate threshold value plays a very important role in the entire process since it defines the mapping of the distress features in the binary image. In most cases, cracks only take a small portion of the entire image. Thus according to the histogram distribution, some dark noisy pixels, such as shadow, fallen leaves, and oil stains etc. are easily misrecognized as cracks. Therefore a simple global segmentation method is not practical for this type of application.

Instead of processing the pavement image as a whole, each tested image is divided into a set of non-overlapping blocks (shown in Figure 6 (a)) for the purpose of local segmentation. The block size is a very important parameter for this type of application. A single window should be large enough to contain both the crack and background regions while representing only the local information.

Since image blocks containing cracks (called “crack blocks” hereafter) tend to present a lower average intensity but higher standard deviation than those non-crack blocks, we can roughly distinguish the crack-blocks from the non-crack blocks. Then a common thresholding method based on Otsu’s algorithm [15] is applied to these crack blocks to obtain a set of local threshold values. Otsu’s method is one of the most successful methods for image thresholding based on the statistics of gray level histogram. It selects the global optimal threshold by maximizing the between-class variance. The average of these local threshold values is selected to be the global threshold value for the segmentation. The process leads to a lower threshold value which could effectively reduce the misclassification caused by low gray value noises.

Assuming an image is represented in L gray levels [0, 1…L-1]. The number of pixels at level *i* is denoted by n_i , and the total number of pixels is denoted by $N = n_1 + n_2 + ... + n_L$. The probability of gray level *i* is denoted by

$$P_i = n_i / N,\ P_i \geq 0,\ \sum_{0}^{L-1} P_i = 1 \tag{4}$$

In the bi-level thresholding method, the pixels of image are divided into two classes with gray levels C_1 and with gray levels [0, 1…t] and C_2 by the threshold [t+1, t+2…L-1] by the threshold t. The gray level probability distributions for the two classes are:

$$w_1 = \Pr(C_1) = \sum_{i=0}^{t} P_i \tag{5}$$

$$w_2 = \Pr(C_2) = \sum_{i=t+1}^{L-1} P_i \tag{6}$$

The means of class C_1 and C_2 are

$$m_1 = \sum_{i=0}^{t} iP_i / w_1$$

(7)

$$m_2 = \sum_{i=t+1}^{L-1} iP_i / w_2 \tag{8}$$

The total mean of gray levels is denoted by,

$$m_T = w_1 m_1 + w_2 m_2 \tag{9}$$

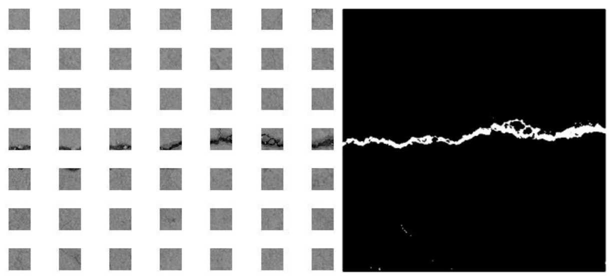
The between-class variance is,

$$\sigma_B{}^2 = w_1(m_1 - m_T)^2 + w_2(m_2 - m_T)^2 \tag{10}$$

Otsu method chooses the optimal threshold t by maximizing the between-class variance,

$$t = \arg\{\max\{\sigma_B{}^2(t)\}\} \tag{11}$$

Once an appropriate threshold value is determined, the pixels with gray level below the threshold are classified as distress pixels and pixels whose gray level value exceeds the threshold are assigned to background. Figure 6 (b) shows the binary image obtained from the proposed segmentation algorithm.

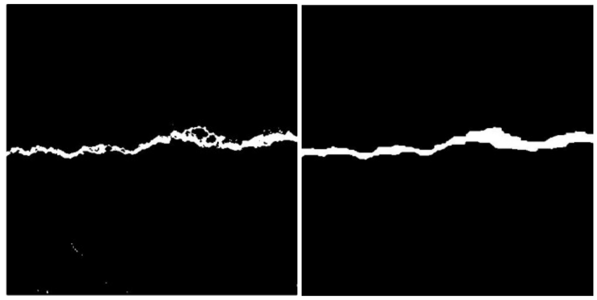


(a) (b)

Figure 6. (a) Subdivide the pavement image into non-overlapping blocks
(b) Binary image obtained from proposed segmentation algorithm

2.1.4. Morphological Operation

In this step, morphological closing is first applied in order to link the disconnected crack pixels by filling small holes and bridging the thin gaps appeared in the binary image. A line structuring element (SE) of size 15 pixels was used for the closing operation. After linking the crack pixels, a binary noise removal approach based on the size of the connected components are applied. Any connected components with pixel numbers less than a predefined threshold value is considered to be noise, therefore, will be removed. The result of this process is shown in Figure 7, where it can be seen that most of the noises have been removed and the small gaps between cracks being filled with object pixels.



(a) (b)

Figure 7. (a)Binary image obtained from local segmentation

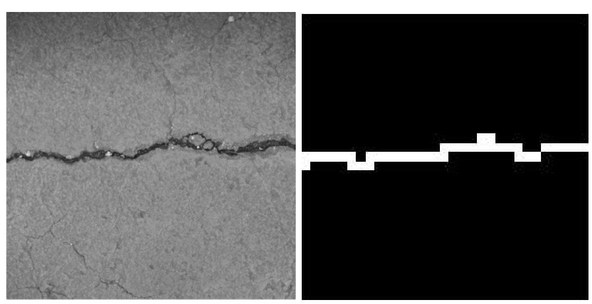
(b)Binary image after morphological operation

2.1.5. Crack Classification

Cracks extracted through the process as explained in the previous section can be classified into different types. In our research, a projection-based pavement distress classification system has been developed, taking into account four distress types: longitudinal cracks, transverse cracks, diagonal cracks and alligator cracks.

a) Crack Representation

In order to a save the processing time and reduce memory storage requirements, the resulting binary image is then subdivided into square tiles, each of which stands for either cracked tile or non-cracked tile. The decision to classify a tile as a crack tile is based on the percentage of crack pixels present in a tile. Any tile with more than 10% of its pixels being crack pixels is considered as a crack tile, therefore labeled '1', otherwise labeled with '0'. The size of the tile depends on the width of the crack. The result obtained at this stage is shown in Figure 8.



(a) (b)

Figure 8. (a) Original pavement image

(b) Pavement image represented by cracked tiles

b) 2-D Feature Mapping

To represent the distribution of crack tiles, two types of histograms, vertical histogram and horizontal histogram as shown in Figure (9), are recorded by accumulating the number of crack tiles presented in each column and row as followed:

$$H[i] = \sum_{j=1}^{M} crack\_tiles[i, j] \text{ , } i = 1, 2\ldots\mathrm{N} \tag{12}$$

$$V[j] = \sum_{i=1}^{N} crack\_tiles[i, j] \text{ , j=1, 2}\ldots\mathrm{M} \tag{13}$$

where V and H represent vertical and horizontal histograms, M and N denote the number of rows and columns, respectively.

These two histograms could demonstrate a clear pattern of the crack. If the crack is developed in a longitude direction, there would be a peak in the vertical histogram. On the other hand, if the crack is developed in the transversal direction, there would be a peak in the horizontal histogram. If the crack is an alligator type, the peaks could be found in both vertical and horizontal directions. For a block crack, peaks could also be found in both histograms with a smaller magnitude than those of an alligator crack.

Based on the above observation, the 2D feature space used for crack classification is composed of the standard deviations of the vertical (feature one) and horizontal (feature two) histograms.

$$\sigma_V = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(V[i]-\mu)^2} \tag{14}$$

$$\sigma_H = \sqrt{\frac{1}{M}\sum_{j=1}^{M}(H[j]-\mu)^2} \tag{15}$$

Where μ represent the mean value of the histogram

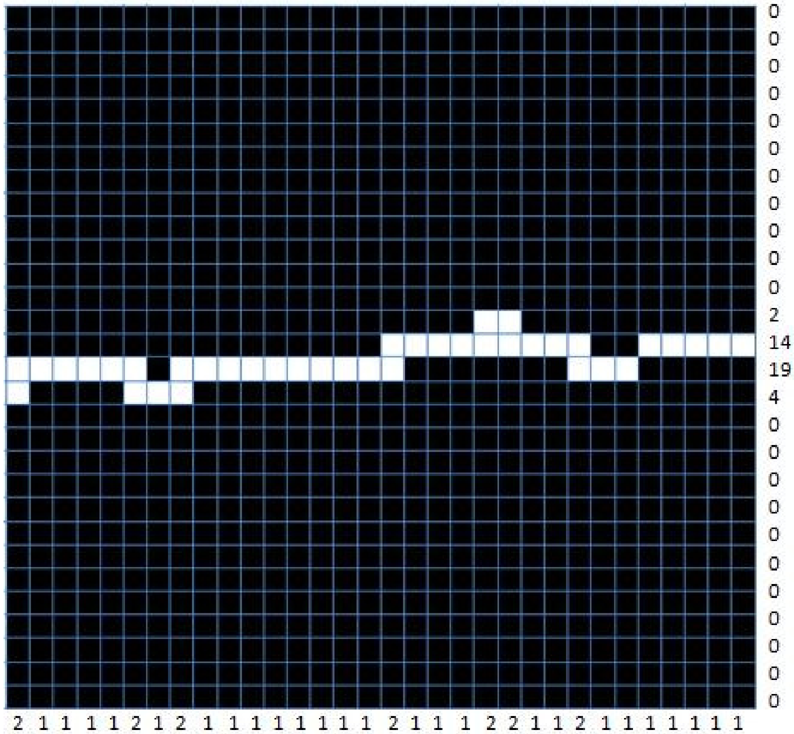


Figure 9. Crack Tiles Distribution

The result of the 2D feature space is depicted in Figure 10. As we can see from graph, the lines L1 and L2 partition the 2D feature space into three regions, region1, rigion2 and region 3, representing different types of cracks. Once the mapping point positioned in this space, the crack is classified according to the following rules:

- If the feature point maps into region 1 of the 2D features space, the crack is classified as a transversal crack.

- If the feature point maps into region 2 of the 2D features space, the crack is classified as a longitude crack.

- If the feature point maps into region 3 of the 2D features space and the percentage of the crack tiles is more than 20% of the entire image, the crack is classified as an alligator crack.

- If the feature point maps into region 3 of the 2D features space and the percentage of the crack tiles is less than 20% of the entire image, the crack is classified as a block crack.

  The result of the example is given in Figure 10. Since the feature point falls into region 1, the crack is classified as a transverse crack.

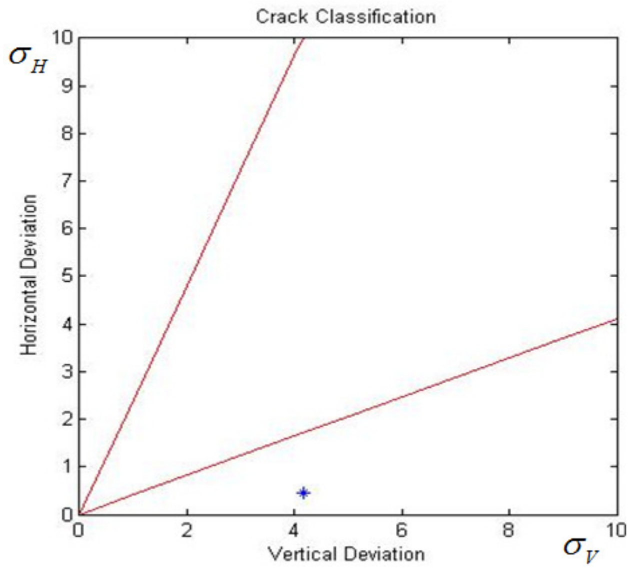


Figure 10. 2D Feature Space for Classification

2.1.6. Experimental Results

In this section, 200 actual digital images taken from the road surface are tested. Figure 11 shows sample crack classification results extracted from the image database.

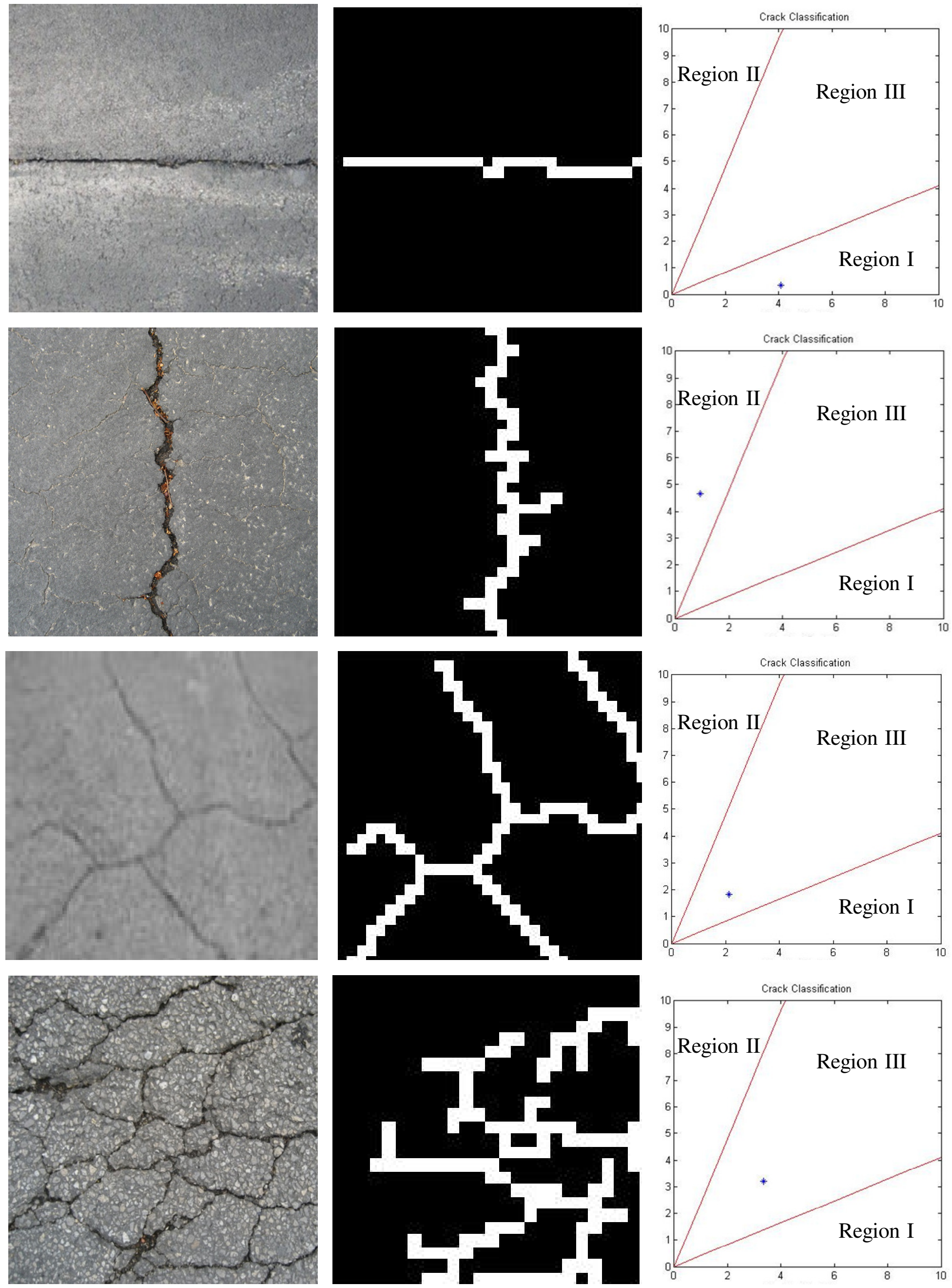


Figure 11. Original images (left column); crack detection results (center column); crack classification feature space (right column).

Two kinds of criteria are considered to assess the system performance. One is computation time and the other is the accuracy of the system. Table 1 shows the detailed test results of the proposed system.

Table 1. Test Results of the System

Crack type	Number of test images	Computation time	Classification accuracy
Transverse	50	2.2 s	100%
Longitude	50	1.9 s	100%
Block	50	2.4 s	97%
Alligator	50	2.5 s	98%

From Table 1 we can see that the system takes only 2.3 seconds on average on a computer equipped with Intel(R) core(TM) 2Duo CPU to determine a crack type for a pavement image of size 512*512 pixels. This is nearly real time and produces an average accuracy of 98.75% for all classes of the detected cracks.

2.2. Beamlet Transform Based Technique for Pavement Crack Detection

We have also investigated a method based on Beamlet transform to extract and classify the crack features from the images of the pavement. Initially, a filtering technique is used to enhance the image with non-uniform illumination. A Beamlet transform, which is a robust method for curvilinear feature extraction, is then applied to extract various types of the pavement surface cracks. The extracted pavement cracks are then classified into four types: vertical, horizontal, transversal, and block types.

2.2.1. An Image Processing Technique for Non-uniform Background Removal

The intensity matrix from an image with a pavement crack contains three types of variations: (1) non-uniform background illumination, a very low-frequency signal; (2) pavement distress or non-distress irregularities (stains or dark materials on surface) having high-frequency components on the edges; and (3) noise, caused by heterogeneous materials and granularity, a random, high-frequency, and low- to medium-amplitude signal [3]. In addition, the healthy pavement surface presents texture variations due to the nature of its ingredients and especially due to the presence of the mineral aggregates, which results in strong variations in color.

In general, the product of illumination and surface reflectance determines the pixel intensity values in an image. For the automatic pavement crack detection system, due to non-uniform lighting conditions, and the pavement's reflectance, the background may have different intensities in different areas. The brightness information is the key to detecting cracks, and thus a non-uniform background will cause classification errors. The non-uniform background intensity effects must be eliminated so that the background has a uniform average intensity, while cracks will still have lower intensities.

Consider a pixel P in a pavement image, and its intensity $B(p)$ which is composed of three components $B_b(p)$, $B_c(p)$, and $B_n(p)$ representing the background illumination signal, the signal due to the occurrence of a crack, and the noise component, respectively. The pixel intensity value can be represented by the following equation:

$$B(p) = B_b(p) + B_c(p) + B_n(p) \tag{16}$$

To extract the crack features, the pavement image needs to be thresholded. However, the pavement image usually has a high-amplitude of the $B_b(p)$ component, and thus the non-uniform background will fail the thresholding operation. In order to extract crack information from the image, it is necessary to convert the background component $B_b(p)$ into a constant intensity value B, i.e., $B_b'(p) = B$, where B is an arbitrarily chosen non-zero gray-level. The process starts with partitioning the image into small windows and the idea is to use a factor to adjust the mean value of each window to the target value B. However, some image windows will contain cracks or other objects. The crack gray level is always much lower leading to a lower mean value of the gray level in the block. Thus, it is necessary to remove the effects of the cracks and other undesired elements. The proposed method for the removal of the non-uniform background can be summarized in the following steps:

1) Partition the image into rectangular windows. The size of the window can vary with the size and type of the input images. However, the window size is 16x16 pixels in our implementation for an input image of size 256x256 pixels.

2) Calculate the mean (G_{mean}), minimum ($G_{\min}$), and the maximum ($G_{\max}$) gray level of each window.

3) Set an upper limit (r_h) and a lower limit (r_l) for which the points with gray levels outside the limits are considered as suspicious points for noise, crack pixels, or other objects on the road. We set the limiting factor to be 30%. Therefore, The range [r_l, r_h] is determined by the following equations:

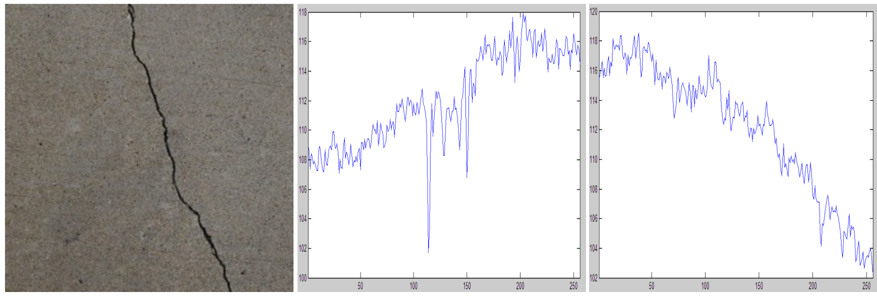
$$r_h = G_{mean} + (G_{\max} - G_{mean}) * 30\% \tag{17}$$

$$r_l = G_{mean} - (G_{mean} - G_{\min}) * 30\% \tag{18}$$

4) With the exemption of the suspicious points, recalculate the mean value of the gray level G'_{mean}. Note that G'_{mean} is the updated mean value without considering the noise and crack pixels.

5) The amplitude correction factor is calculated as, $f = B/G_{mean}$, thus the modified picture intensity value is defined as, $I' = I * f$.

Figure. 12(a) shows a pavement image with non-uniform background. The intensity profiles of a row and a column scan are shown in Figure 12 (b) and (c), respectively. The results of non-uniform background removal are shown in corresponding Figures 13 (a, b, c).

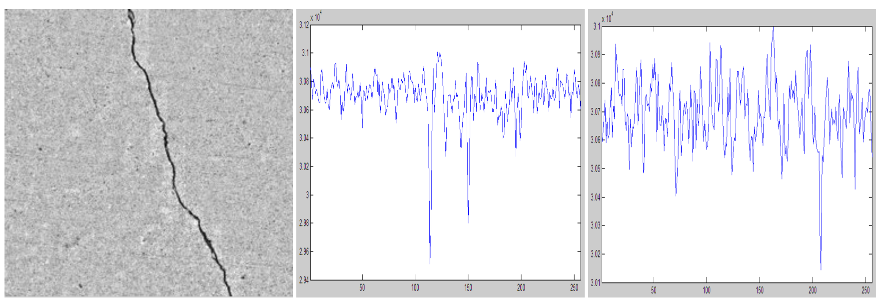


(a) (b) (c)

Figure 12. (a)Pavement image with non-uniform background

(b)Average Gray level plot in x direction of the original image

(c)Average gray level plot in y direction of the original image



(a) (b) (c)

Figure 13. (a) Improved pavement image

(b)Average gray level plot in x direction after non-uniform background removal

(c) Average gray level plot in y direction after non-uniform background removal

2.2.2. Beamlet Transform

The traditional signal detection algorithms for pavement crack detection are generally based on pixel-level processing, and most of them have very poor SNR ratios. Beamlet transforms have been proved to be insensitive to noise, computationally efficient, and able to detect features with high accuracy. Beamlets are a simple dyadically organized collection of all line segments at different locations, orientations, and scales. The Beamlet transform is the collection of line integrals along the set of all Beamlets. This method allows for the extraction of linear features such as edges in noisy pictures, where traditional methods may fail.

The concept of Beamlet transform was first introduced by Donoho and Huo as a tool for multi-scale image analysis [16]. Consider an image as a function residing on a $[0,1]\times[0,1]$ unit square with "pixels" of size 1/n by 1/n, the following definitions would be helpful in obtaining the Beamlet transform of an image.

Definition 1. A dyadic square **S** is the collection of points $\{(x1,x2): [k_1/2^j, (k_1+1)/2^j] \times [k_2/2^j, (k_2+1)/2^j]\}$, where $0 \le k_1, k_2 < 2^j$ for an integer $j \ge 0$.

Definition 2. Consider two vertices $v_1, v_2 \in [0,1]^2$ within a dyadic square, the line segment $b = \overline{v_1 v_2}$ is called a beam. There are $O(n^4)$ such beams if only beams connecting vertices $(k_1/n, k_2/n)$ are considered.

Definition 3. Take the collection of all dyadic squares at scales $0 \le j \le J$ and fix resolution δ, the set of Beamlets is the collection of all beams connecting vertices on the boundary of each dyadic square. There are $O(n^2 \log_2 n)$ Beamlets [16]. Figure 14 shows Beamlets at different scales.

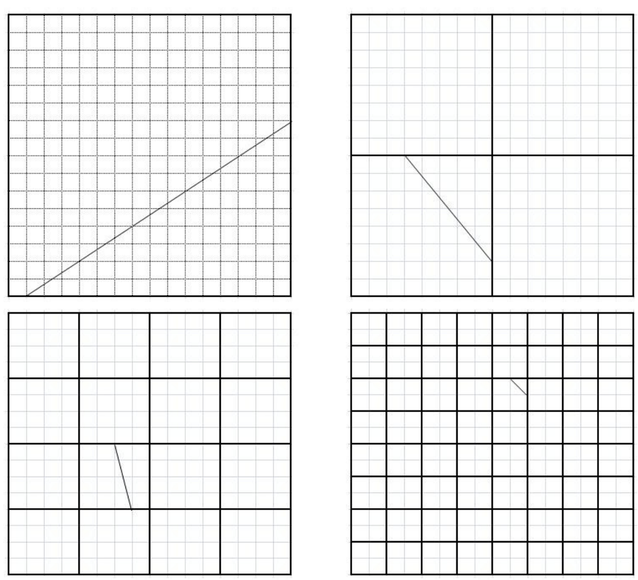


Figure 14. Beamlets at Different Scales

The Beamlet transform is the collection of line integrals along the set of all Beamlets. Let $f(x_1, x_2)$ be a continuous function on $[0, 1]^2$, then the Beamlet transform of function f is defined as follows,

$$T_f(b) = \int_b f(x(l))dl \,, \; b \in B_{n,\delta} \tag{19}$$

where the collection of Beamlets, $B_{n,\delta}$, is a multiscale collection of line segments occurring at a full range of orientations, positions, and scales.

However, for a digital image f_{i_1,i_2}, the Beamlet transform for all the points along the Beamlet b is defined as,

$$T(b) = \sum_b f_{i_1,i_2} \phi_{i_1,i_2} \tag{20}$$

where ϕ_{i_1,i_2} is considered to be the weight function for each pixel. In this paper, we use the following equation:

$$\phi_{i_1,i_2} = \frac{l_n}{\sqrt{L}} \tag{21}$$

where L is the total length of the beam , and l_n is the length of a segment in each square pixel on the beam. Obviously,

$$L = \sum_n l_n \tag{22}$$

Figure 15 shows the Beamlet transform as a weighted sum of pixel values along the shaded line that the Beamlet traverses.

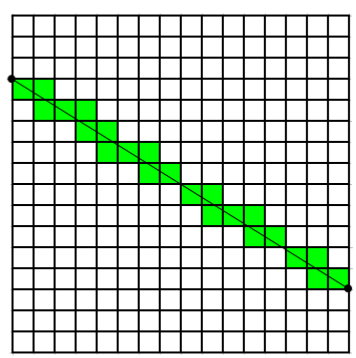


Figure 15. Beamlet Transform as a Weighted Sum of Pixel Values Along the Shaded Line

After removing the non-uniform background using the technique explained in the previous section, a thresholding operation is applied to extract the binary crack image. In order to further reduce the calculation, only fix sized dyadic squares are considered, where there are $O(n^2)$ such Beamlets. To proceed, the binary image is then divided into smaller windows, and a Beamlet transform is applied in each window.

For each window, the Beamlet transform is applied and the beam which provides the max value will be selected if its value exceeds a threshold value. The length of the beam will determine the length of the cracks in each window.

2.2.3. Crack Detection and Extension Check

To obtain an extended crack for further analysis, the neighboring blocks are checked for further extensions. To proceed, the following three data structures are required.

a. Status Matrix: Record the status of each block. Valued with “unchecked”, “no crack”, or crack id, which specifies the crack number.

b. Length Table: Record the length of each branch.

c. Branch Candidate Table: Record the starting point of each branch candidate.

The crack connectivity check will then follow the following steps:

1) Scan status matrix, block by block from left to right, top to bottom, and find the first block with a crack feature with “unchecked” status. Modify the status to the corresponding id number, add the crack length in the block to the length table of the current branch, and then proceed to check all its eight neighboring blocks with “unchecked” status.

2) If one and only one neighbor is a crack block, then add the crack length to the corresponding item in the length table, move to the neighboring block, and continue the process.

3) If there is more than one block detected with cracks, then select one as the current branch extension direction and continue the extension check. Save all the others into the branch candidate table.

4) If there is no unchecked crack block remaining in the neighboring blocks, it means the branch extension has reached its end. If the branch length is shorter than a threshold, then it is not a real branch and will be ignored.

5) Find the next branch candidate from the branch candidate table, and continue the extension check until the table is empty.

6) The length of a crack is the sum of the length of all the branches contributing to that crack. Finally, if the length of the crack is shorter than a threshold, it is not considered to be a real crack.

2.2.4. Crack Classification

Generally, cracks in the pavement images possess linear features, embedded in noise, and are discontinuous. Additionally, the pavement images have specific patterns which make crack detection more difficult using traditional pixel based methods. The Beamlet transform will be a suitable algorithm for crack detection due to its robustness to line segment detection.

Following the above crack extension procedure, cracks are extracted and their projection in horizontal and vertical directions can be measured. This will in turn provide the information necessary for crack classification. Cracks are classified into four types: vertical, horizontal, transverse, and block types. The type of a crack is determined by its angle with the horizontal axis (Ω) and the number of branches in the crack, as summarized in Table 2. Note that the angle Ω is calculated according to the start and end points of the crack. For each window, the maximum Beamlet transform value is defined as the crack length in the block. The total crack length is defined as the sum of all the blocks along the crack.

Table 2. Features for Different Types of Cracks

Direction	Ω	Branches?
Vertical	$\Omega >= 60^o$	No
Horizontal	$\Omega <= 30^o$	No
Transverse	$60^o > \Omega > 30^o$	No
Block	-	Yes

2.2.5. Experiment Results

The proposed algorithm has been implemented and its performance and simulation results are presented. Figure 16 shows different types of pavement cracks and the corresponding results from the Beamlet transform. Table 3 shows the length and the type of each crack from the pavement images in Figure 16. Note that, Figure 16 (I) is an image containing two separate cracks, a horizontal crack which is longer than the shorter vertical one. Figure 16 (II) has only one vertical crack which runs across the whole image. Figure 16 (III) is composed of two branches of a crack, so it is considered a block type. In Figure 16 (IV), there are four branches of cracks and is also considered a block type.

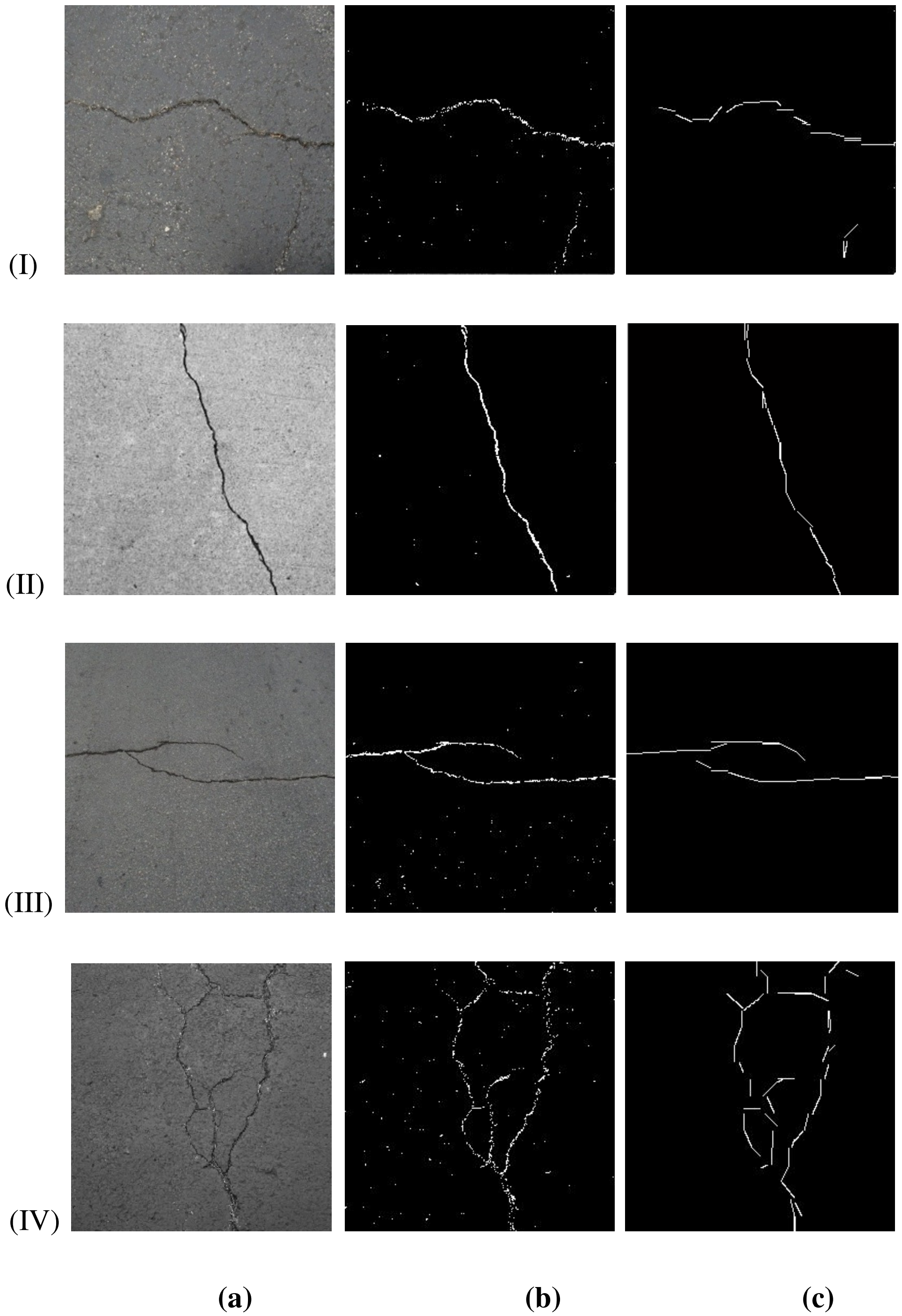


Figure 16: Pavement image with longitude crack and crack detection with a beamlet transform

(a) Original pavement image; (b) Enhanced and thresholded. Image; (c) Result after the beamlet transform and extension check

Table 3. Results of Crack Classification

Image	Crack id.	Ω	Branches?	Crack Length	Crack Type
(I)	1	9^o	No	308.07	Horizontal
	2	67^o	No	68.8	Vertical
(II)	1	71^o	No	341.15	Vertical
(III)	1	-	Yes	425.73	Block
(IV)	1	-	Yes	993	Block

3. Conclusions

We have presented a low-cost, user-friendly, fast pavement distress detection and classification method using advanced image processing techniques. It has been shown that the proposed pavement analysis system allows complete automation with near real-time evaluation of pavement distresses. The accuracy of this system in identifying pavement distress meets the standards set out by the road authority for pavement management. The experimental results indicate that our proposed system produced highly reliable and accurate results from the 200 tested samples.

In addition, a beamlet transform based technique to extract the linear crack features from pavement images is introduced. Beamlet transform provides an effective method for the extraction of curvilinear features such as cracks in pavement images. An enhancement method is applied to reduce the effects of non-uniform background and undesired objects to facilitate the application of the beamlet transform. Experimental results demonstrate that the proposed beamlet transform based method is very effective with the presence of noise in pavement images and could be a viable alternative to common pixel based approaches to crack extraction.

The applications developed in this research were mainly focused towards distress detection and classification. Future developments will target the analysis of the crack properties, such as width, length and severity of the cracks.

4. References

[1] Xu B. and Huang Y. “Development of an Automatic Pavement Surface Distress Inspection System”, Tech report FHWA/ TX-05/7-4975-1, October 2003, University of Texas at Austin.

[2] Maser, K. R., "Computational Techniques for Automating Visual Inspection." Working Paper, 1987. Department of Civil Engineering, MIT, Cambridge, Mass.

[3] Li L, Chan P, Rao A, Lytton R L., “Flexible Pavement Distress Evaluation Using Image Analysis”, Proceedings of the Second International Conference on Applications of Advanced Technologies in Transportation Engineer, 18-21, pp. 473-477, August 1991.

[4] H. N. Koutsopoulos and A. B. Downey, “Primitive-Based Classification of Pavement Cracking Images”, *Journal of Transportation Engineering*, vol. 19, issue 3, pp. 402 418, 1993.

[5] JaChing Chou, O'Neill, W. A., and Cheng, H.D., “Pavement Distress Classification Using Neural Networks”, Systems, Man, and Cybernetics, IEEE International Conference, vol. 1, pp. 397-401, 1994.

[6] A. Georgopoulos, A. Loizos, and A. Flouda, “Digital Image Processing as a Tool for Pavement Distress Evaluation”, ISPRS *Journal of Photogrammetry and Remote Sensing*, 50(1), pp. 23-33, 1995.

[7] H. D. Cheng, J. R. Chen, C. Glazier, and Y. G. Hu, “Novel Approach to Pavement Distress Detection Based on Fuzzy Set Theory.” *J. Comput. Civ. Eng*., 13(4), pp. 270–280, 1999.

[8] Jian Zhou, Peisen S. Huang, Fu-Pen Chiang, “Wavelet-Based Pavement Distress Detection and Evaluation”, *Optical Engineering,* 45(2), pp. 1-10, 2006.

[9] Kelvin C. P. Wang, Qiang Li, and Weiguo Gong, “Wavelet-Based Pavement Distress Image Edge Detection with À Trous Algorithm”, Transportation Research Record, *Journal of the Transportation Research Board*, No. 2024, pp. 73-81, 2007.

[10] Jian Zhou, Peisen Huang, Fu-Pen Chiang, “Wavelet-Based Pavement Distress Classification”, Transportation Research Record: *Journal of the Transportation Research Board*, vol. 1940, pp 89-98, 2005.

[11] Bahram Javidi, Jack Stephens, Sherif Kishk, Thomas Naughton, John McDonald, Atef Issac, “Pilot for Automated Detection and Classification of Road Surface Degradation Features”, *JHR 03-293*, Nov. 2003.

[12] Kelvin C.P. Wang and Weiguo Gong, 2002. Automated Pavement Distress Survey: A Review and A New Direction. Pavement Evaluation Conference, 21-25, 2002, Roanoke, Virginia.

[13] D. Meignen, M. Bernadet and H. Briand, "One Application of Neural Networks for Detection of Defects Using Video Data Bases: Identification of Road Distresses", Proceedings of Eighth International Workshop on Database and Expert Systems Applications, Page(s):459 – 464, USA, 1-2 Sep 1997.

[14] H. Cheng and M. Miyojim, "Automatic Pavement Distress Detection System", *Journal of Information Sciences 108,* ELSEVIER, Page(s) 219-240, July 1998.

[15] N. Otsu, "A Threshold Selection Method from Gray_Level Histograms", *IEEE Trans. Systems, Man, and Cybernnetics*., Vol. 9, No 1, PP. 62-66.

[16] D. L. Donoho and Xiaoming Huo, "Beamlets and Multiscale Image Analysis," in Multiscale and Multiresolution Methods, Lecture Notes in Computational Science and Engineering. Springer, 2001.