A Thesis

Entitled

Pavement Crack Detection System Through Localized Thresholding

By

Nikhil Katakam

Submitted as partial fulfillment of the requirements for

The Master of Science in Engineering

________________________
Advisor: Dr. Ezzatollah Salari

________________________
College of Graduate Studies

The University of Toledo

December 2009
The University of Toledo

College of Engineering

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Nikhil Katakam

ENTITLED Pavement Crack Detection System Through Localized Thresholding

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Engineering

Thesis Advisor: Dr. Ezzatollah Salari

Recommendation concurred by

__________________________________________
Dr. Mohsin M. Jamali

__________________________________________
Dr. Lawrence Miller

Dean, College of Engineering
An Abstract of

Pavement Crack Detection System through Localized Thresholding

By

Nikhil Katakam

Submitted as partial fulfillment of the requirements for the Masters of

Science Degree in Engineering

The University of Toledo

December 2009

Maintenance of pavements is a very important aspect for the Departments of Transportation in any country. The first step towards maintenance is the identification of faulty pavement areas and their documentation for further action. Many methods have been devised to identify the cracks on pavements apart from the crude process of manual inspection. Image processing, ultrasonic detection and infrared detection have been the most
common methods of automated inspection. For the use of images, external factors such as shadows and improper lighting might result in noise. Localized thresholding is implemented by dividing the image into smaller blocks and identifying a local threshold and finding the crack pixels using the threshold of each block. Correlating the intensity and the relative values of the RGB components of the image, the region of interest is obtained from the original image. The image is converted into a black and white image in the end to identify the existing cracks, recreate the lost crack image, identifying the type of cracks based on the orientation of the cracks and thereby rating the pavement accordingly.
Acknowledgements

Firstly, I would like to thank my advisor Dr. Ezzatollah Salari for his support through my thesis for his support and encouragement, which was always there when I needed it. I thank him for the help and inputs that he has given me throughout the thesis and I am thankful to the outlook he has helped me to develop that enabled me to look at work in a different angle and refine my thinking process to convert my idea from a crude hypothesis to a refined application in image processing.

I would also like to thank Yao Sun and Lihao Hong who worked with me during this period for helping me when I needed help and for the discussions we have had on the concept for long hours during the work. They have helped me look not to overlook certain aspects and helped me to examine my work to every detail and did not let me ignore any aspect of the procedure that I would have, had they not been there for me.

I thank my parents, for their absolute confidence in me. Their constant encouragement helped in making the completion of my graduate work possible. Reinforcing my spirits at some crucial junctures, they provided much encouragement throughout the thesis. It is to them that I dedicate this work.

Finally I would like to thank God almighty, for this blessed opportunity
# Table of Contents

Abstract iii

Acknowledgements v

Table of Contents vi

List of Figures viii

Chapter 1 1

Introduction 1

1.1 Motivation 3

1.2 Related Research 4

1.3 Thesis outline 6

Chapter 2 8

Background Separation 8

2.1 Preprocessing 8

2.1.1 RGB Components 9

2.1.2 Standard Deviation 11

2.2 Median Filtering 12

2.3 Background Separation Procedure 15

Chapter 3 17

Crack Detection 17

3.1 Thresholding 17

3.2 Localized Thresholding 18
3.2.1 Advantage 21
3.3 Contrasting 22
3.4 Conversion to Black and White Image 24
3.5 Noise Reduction and Connectivity Establishment 27

Chapter 4 29

Line Tracing and Crack Representation 29

4.1 Watershed Method 29
4.2 Scan Line Method 31
4.3 Representation Methodology 32
  4.3.1 Noise Reduction after Compression 34

4.4 Classification using Hough Transform 35
  4.4.1 Hough Transform 35
  4.4.2 Proposed Methodology 36

Chapter 5 38

Tests and Results 38

5.1 Complete Method 38

5.2 Image with less or no Background 40
  5.2.1 Image with compound cracks 46

5.3 Image with background 57

Chapter 6 62

Discussion and Conclusion 62

6.1 Conclusion 62
  6.1.1 Limitations 63

6.2 Future Work 63

References 65

APPENDIX - Matlab Code 69
List of Figures

1.1 Infrared Image of the pavement during night time 2
1.2 Sample image processed using thresholding technique 3
2.1 Image with pavement and background 9
2.2 Image with red, green and blue color components 10
2.3 Flow chart showing median filtering 13
2.4 Color Image converted to grayscale 14
   (a) Original Image 14
   (b) Gray Scale Image before filtering 14
   (c) Gray Scale Image after filtering 14
3.1 Images divided into smaller squares 19
   (a) Original grayscale image 19
   (b) Image divided into square blocks 19
3.2 Thresholds of squares in the first row 19
3.3 Image showing traces of cracks 21
3.4 Histogram of intensities of pixels in the image 22
   (a) Before contrasting 22
   (b) After contrasting 22
3.5 Image with relatively lesser intensity 23
3.6 Histogram of an image after contrasting
   (a) Image showing crack
   (b) Histogram after contrasting
3.7 Identification of continuity between pixels
4.1 Figure showing scan lines
   (a) Scan lines in horizontal direction
   (b) Scan lines in vertical direction
4.2 Conversion of an image to 3-D
4.3 Image with scan line and crack pixels
5.1 Flow chart showing the full procedure
5.2 Image with no background
   (a) Original image
   (b) Gray Scale
   (c) Background Separation
   (d) After Contrasting
   (e) Converting into black and white image
   (f) Reversing the intensities
   (g) Identifying continuity
   (h) Image converted to black and white using classical thresholding
   (i) Compressed Image
   (j) Expanded version of the compressed image
   (k) Hough Transform of Image
5.3 Image with compound cracks

(a) Background Separation 46
(b) After contrasting 47
(c) Conversion into black and white image 48
(d) Reversing the intensities 49
(e) Identifying continuity 50
(f) Expanded version of the Compressed Image 51
(g) Hough Transform of Image 51
(h) Image in grayscale 52
(i) Image after contrasting 53
(j) Black and White image 53
(k) After reversing contrast 54
(l) Verifying continuity 54
(m) Expanded version of compressed image before noise reduction 55
(n) Expanded version of compressed image after noise reduction 56
(o) Hough Transform of Image 56

5.4 Image with some background

(a) Original Image 57
(b) Grayscale Image 58
(c) Background Separation 58
(d) After Thresholding 59
(e) After conversion into black and white and reversing intensities 59
(f) Verifying continuity 60
(g) Expanded version of compressed image 60
(h) Hough Transform of Image 61
Chapter 1

Introduction

Road transportation is the most common means of transportation in any country. The existence of well maintained roads is mandatory for efficient transportation. However, the roads are usually brittle and have a tendency to deteriorate. Despite efforts to avoid cracks due to temperature variations by using different kinds of materials, the elasticity can only be achieved by compromising the hardness of the roads which is not feasible. Dramatic temperature variations result in cracking of the roads and therefore, their quality is reduced, further reducing the average speed of commutation on them. Cracking of roads is due to expansion and contraction of roads caused by temperature variations and the absence of clearance for roads.

This is not the only reason for damage of roads. Other factors such as traffic, water deposition etc., also result in road damage. Therefore, it is necessary to perform periodic inspections and keep track of the condition of the pavements to ensure better transportation. The primitive method of manual inspection takes a long time and money. Considering the size of the road network in the current day, it becomes virtually impossible to deal with every road and maintain them in perfect condition. Therefore, automated methods have been invented to capture, identify and store the cracks on roads accurately. There have been many methods to do this including camera monitoring, infrared imaging, ultra sonic location etc. The methods of crack detection in pavements
are also influenced by the data collection mechanisms. Collection of images is the easiest of the data collection methods. Therefore, image processing for crack detection is more feasible than any of the other proposed techniques. Wisecrax [19] is an example of a commercially available device that uses infrared imaging to detect cracks on pavements. This is mounted on a vehicle with a camera that takes pictures of the pavements continuously. New methods are being devised to identify cracks more efficiently and get closer towards perfection as good as the human eye and are still in the process.

![Infrared image of the pavement during night time](image)

**Fig 1.1 Infrared image of the pavement during night time**
The image above shows an infrared image of the pavement during night time. This is an example of data acquired by which further investigation takes place to find cracks on pavements. However, the images we use are colors in broad daylight. On the other hand, a photograph with a flash of the same would also provide the right image for processing during the night.

1.1 Motivation

The routine method of image processing for crack detection on pavements undergoes three phases. They are preprocessing, thresholding and post processing. The preprocessing stage involves filtering and other mechanisms to basically convert the image into a form that is suitable to use thresholding mechanism on it. For the preprocessing stage, only the intensities are usually taken into consideration. The effective utilization of colour components is not often seen. Moreover, usage of more than one thresholding mechanism value does give a dynamic approach to crack extraction. Extracting the of Region of Interest (ROI) does give the method an edge over other methods since the input of images that have not just the pavement would also give a consistent output.

![Fig 1.2 A sample image processed using thresholding technique](image-url)
The figure above is an example of sample data that is acquired for processing. While the picture on the left is the original image, the picture on the right shows a sample method of data processing using a thresholding technique. As it is apparent, the image contains cracks that have been already identified in the image on the left while the thresholding technique filters the rest of the image from the cracks by whitening out the rest of the image and darkening the cracks. This image can be used for further processing and classification which in turn gives a usable output. However, simple thresholding without proper preprocessing mechanism might not be entirely consistent in the case of an image that has objects other than the pavement and are of similar intensity as the pavement and might look like the pavement itself when converted into a gray scale image. This might give an inconsistent result with simple thresholding mechanisms without background separation.

1.2 Related Research

The methods of reconstruction, alteration or changes in intensities of pixels etc to make the cracks on the pavement surface from the background of the rest of the image along with noise reduction/ removal are performed. The usage of median filtering and linear regression techniques for preprocessing were described by Haroun Rababaah, Dana Vrajitoru, and James Wolfer\(^2\). This method uses Thresholding and then Median filtering process is suggested by A. Lozios, A. Georgopoulos, A. Flouda\(^1\). Identifying the types of cracks by finding the mean values of intensities of each row and terming the crack horizontal or vertical was described by Egemen Teomete et al\(^4\), Super-resolution restoration from a still image may be approached by using ‘regularization’ methods which constrain the feasible solution space by employing a ‘priori-knowledge’ described
by Sean Borman, Robert Stevenson. Further processing can be done with blocks by
determining the sub-block size in pixels used in the projection histogram. The intensities
are changed to represent the pixels accordingly. To minimize the speed and effort for
crack detection, another method was proposed by David Mould at the University of
Saskatchewan.

Distress quantification and quantization are some of the tasks being done in the
post processing stage. The procedure of radon transform for distress classification and
distress quantization norms were described by Jian Huang Peisen et al. Teomete, Viren
R Amin, et al, also developed a method for distress quantification and quantization
effectively. Images are acquired from Long Term Pavement Performance Program
(LTPP). The images are cropped to 256 X 256 from 2048 X 3072 for processing.
Matlab is used to program the process and Region of Interest (ROI) is found out. In the
preprocessing stage, the intensities of pixels are noted and the image is divided into
blocks and the cracks are found to be either horizontal or vertical by identifying the
relative mean value of intensity of each row.

The detection step is done by first determining the sub-block size in pixels used in
the projection histogram. The crack detection is easy since most of the work is done in
the pre-processing stage. The intensities are changed to represent the pixels accordingly.
The drawback of this method as stated is that the method is used only for horizontal or
vertical cracks. Complex cracks cannot be identified clearly. The process, though
effective, is very sensitive to noise and noise is represented just as apparently as the
actual cracks.
Crack detection is also done by modifying the textured area into a three dimensional space with the different forms of textures having different heights and visualizing a rising water level from the bottom and trying to find out if the two regions are joined without flooding. This was explained by Timothy Evans\textsuperscript{19} in his thesis which was conducted at Monash University. The drawback of this method is that in trying to eliminate the false positives, the cracks that were supposed to be identified only are based on probability only.

1.3 Thesis outline

The thesis consists of a total of three phases like in any other pavement crack detection mechanism. They are preprocessing, thresholding and post processing. The second chapter in the thesis deals with the pre processing stage that introduces the RGB (red, green, blue) components of the image. The relation and distribution of the components in each pixel helps to determine the color and effective utilization of the distribution to identify the region of interest of the image.

The third chapter introduces the thresholding stage of the procedure. It shows the different forms of thresholding and provides an understanding of the logistics of different types of them. The chapter also introduces the concept of localized thresholding introduced in the thesis and the other sub-phases of the thresholding mechanism to identify cracks from the region of interest, noise reduction through the procedure and conversion of the given image to a black and white image in the end.

The fourth chapter talks about the post processing phase of the process. This chapter talks about the line tracing mechanism that is used to identify the crack blocks
and non crack blocks of the image. The procedure to identify blocks by retaining the continuity of the pixels and the noise reduction mechanism that takes place during the method are explained in this chapter. It is also explains crack identification and representation at the end of the procedure and the advantage of such representation.

The fifth chapter shows the step by step results that have taken place during the entire process in the three phases and the identification of the cracks in the end. It brings to light the advantages of the method over other methods and how it enables better storage and representation of the crack images in a database.

The sixth chapter explains the conclusions that can be obtained from the process and the scope of future work that could be possible from the thesis. It also identifies the scope for better identification of the cracks with lesser computation speed with little or no noise in the images.
Chapter 2

Background Separation

This chapter describes the preprocessing stage of the procedure. The chapter explains the definition of preprocessing in image processing, the components of the image, the input file and its format, the method of background separation and converting the image into a format suitable for further processing.

2.1 Preprocessing

Preprocessing is defined as the conversion of the input of a given procedure to a format that is acceptable for the later phases to obtain an acceptable output. Preprocessing stage for this method is the extraction of region of interest of the image. When an image is given as an input, there can be a fair chance of other objects being present in the image along with the pavement. The reason it is significant is that the rest of the image, which can be either dark or light can influence in the final outcome of the image. Therefore, in an effort to obtain a more refined and acceptable input, the preprocessing method is utilized.

The preprocessing method involves the separation of the background from the pavement. The objective of this method is obtaining the Region of Interest (ROI) from the image and separating the pavement from the surrounding background. As shown in
Fig 2.1, there can be images with other objects or areas other than the pavement in the image.

![Image with pavement and background](image.png)

**Fig 2.1 Image with pavement and background**

Although the region of interest is fairly large in the image, it becomes important to remove the rest of the image so that the focus of work for further stages is solely on the pavement.

### 2.1.1 RGB Components

Every colored image is made up of the RGB (Red, Green and Blue) components. They are called the primary colors of the spectrum. The combination of the three colors in different magnitudes result in different colors. The same concept is used in
photography to identify different colors. Since we are using the colored image, this becomes significant. While an image is considered, in originality, three arrays with red, green and blue components with values for the corresponding pixels are considered. The colors are identified with the relative values of the pixels in each component.

In addition to the color of the components, the intensities of the corresponding pixels are also identified by the values of the three components of the image. The range of components of pixels is from 0 to 255. Therefore, if the values of the components are close and are high (closer to 255), the color of the corresponding pixels would be closer to white. On the contrary, if the values of components are less than or closer to the middle of the range the pixels would be dark grey or black. When the values of the components are wide apart, the values that are relatively higher cast their influence on the color of the pixel there by resulting in a color. An example of the color components and their resultant colors formed by their combination are shown in the figure below.

![Image with the Red, Green and Blue color components](image_url)
The red, green and blue components, when superimposed over each other, result in different colors. The magnitude of the components that are superimposed result in the type and intensity of the colors.

2.1.2 Standard Deviation

Standard deviation is a mathematical procedure used to identify the variation of the red, green and blue components in each pixel. The standard deviation is defined as the square root of the fraction of the summation of mean deviations of a set of values. This can be represented in the form of an equation as follows

$$\text{Standard Deviation } \sigma = \sqrt{\frac{\sum(x_i - \bar{x})^2}{N}}$$

The advantage of the standard deviation over mean deviation is that the range of the standard deviation values can be much smaller and can result in a much more accurate result which would enable a better benchmark to extract the region of interest. The most significant aspect is that the standard deviation values are influenced by all three components of the pixels rather than one or two values. Using the standard deviation values and intensity values, the results to obtain the region of interest becomes much easier. Since the method used in this process is localized thresholding, standard deviation becomes a good benchmark to identify the deviation of each component from one another, thereby determining the color of one or set of pixels in consideration.
2.2 Median Filtering

Noise reduction is one aspect of preprocessing phase of crack detection process. Filtering is the most common form of noise reduction. Filtering is a statistical method that processes the pixels by applying a filter which is either a function or an equation to process every pixel to get a more usable output for further stages of crack detection.

Median filtering is one of the most commonly used preprocessing techniques for crack detection today. The purpose of median filtering is to iron out the inconsistencies of pixel values in the image. When an image is photographed, the relative brightness or darkness of an image is felt when there are neighboring pixels in contrast to the original pixels. This is exactly what is required to identify a crack as well. However, the same thing holds true for the rest of the image and the inconsistencies exist at a pixel level rather than in a block of a few pixels. In that case, the average values of each component in a region become inconsistent, which is undesirable. Therefore, median filtering is employed to iron out the inconsistencies in pixel values which are desired in photography but not during crack detection.

Such a property of median filtering is explained by a classic example of salt and pepper noise in Matlab© which is a random addition of black and white pixels into a gray scale image. This is removed aptly by the median filtering through smoothing of the image but the quality of the image is compromised.

In the process, median filtering is performed only after converting the color image into a gray scale image. Since the separation of region of interest is accomplished already by identifying neighborhood of a given pixel, usually 3 × 3. However, the median
filtering can be done with different neighborhoods other than a 3 × 3 like a 5 × 5. This means that with the target pixel as the centre, the matrix of pixels around it of the order \( n \times n \) is identified. The value of \( n \) is usually odd to carry out median filtering since the target pixel needs to be in the middle of the neighborhood and the process is carried out. The process is shown below.

\[
\begin{array}{ccc}
15 & 23 & 19 \\
65 & 67 & 21 \\
94 & 13 & 21
\end{array}
\rightarrow
\begin{array}{ccc}
13 & 15 & 19 \\
21 & 32 & 65 \\
65 & 67 & 94
\end{array}
\rightarrow
\begin{array}{ccc}
15 & 23 & 19 \\
65 & 67 & 21 \\
94 & 13 & 21
\end{array}
\]

Median filtering can be depicted with the help of a flow chart as shown below.

Fig 2.3 Flow chart showing median filtering procedure
The purpose of the median filtering is the removal of noise pixels by smoothing the image. However, the quality of the image suffers. The degree of smoothing is directly proportional to the size of the neighborhood identified for the filtering process. For an image converted into gray scale, the image before median filtering and after median filtering are shown below [23]. The filtered image uses a $3 \times 3$ neighborhood to apply the median filter.

Fig 2.4 Color image converted to grayscale

(a) Original Image
(b) Gray scale image before filtering
(c) Gray scale image after filtering
2.3 Background separation procedure

The background separation is the first phase of the process. The entire image is divided into blocks of equal size depending on the resolution of the images. The size of the blocks varies depending on the resolution of images since in a high resolution image, the number of pixels representing a unit area is higher than pixels of lower resolution and therefore would require a higher number of pixels in high resolution images to represent an area of unit size.

The total of each component among R, G and B in each block is identified and the standard deviation is calculated. The wavelet transform is another method employed for identifying cracks. The standard deviation is used as the benchmark for identification of region of interest. The average intensity of the block also plays an important role in identifying the rest of the image since the blocks with lesser intensity might contain a higher value of one component that might result in the elimination of the blocks and blocks with lower intensity have a higher probability of being crack blocks.

The intensity of the blocks that are considered as background are whitened. In other words, the intensity of the pixels in the block is converted to 255 since the range of intensities of pixels in the image is 0 to 255.

The next criterion to be verified is the continuity of the blocks. Like crack pixels, the region of interest can be obtained by verifying the intensities of the blocks in the neighborhood. The image is converted into grayscale for this method to look at the average intensity of each block. Each block should be complimented by other neighboring blocks that have already been erased. The erased blocks are white in color.
In other words, the average intensity of the neighboring blocks has to be 255 in a gray scale image. In the event that there are not as many blocks those are removed, the block is retained. If there are other blocks that contain more than two erased neighboring blocks, the block is erased. This enables optimum accuracy in the extraction of region of interest and also allows for the continuity of the blocks that are removed.
Chapter 3

Crack Detection

The second phase of the process is the crack detection. After identifying the region of interest cracks are extracted from the region of interest. This chapter describes the procedure to extract cracks from the region of interest obtained from the background separation phase.

3.1 Thresholding

Thresholding is the most important phase of the processes. After extracting the region of interest, the next phase is to effectively identify the cracks on the pavement. Thresholding is the most common way of identifying cracks on pavements especially in image processing. The basic principle of thresholding is to identify a value for a given property of an image to enable classification and segregation of pixels into categories. However, it is the method and the condition for threshold that differs from procedure to procedure.

Thresholding has been described in many different ways before. The crudest method of thresholding uses the intensities of pixels to find an average intensity of all the pixels in the image. This is usually followed by conversion of the grayscale image to black and white where the pixels with intensities higher than the threshold are converted
to white and the rest of the pixels are converted to black. The rule that is used most commonly in every crack detection mechanism through image processing is that each crack pixel must have neighboring pixels that are also considered crack pixels. Identifying the threshold, the area of the image considered and the kind of constraints that are established to identify the crack pixels differ from method to method. However, a hard and fast rule to identify the crack pixels results in a significant amount of noise. Therefore, it is always advisable to identify a dynamic set of rules that are adaptive to the kind of input images for effective crack detection.

3.2 Localized Thresholding

Localized thresholding, as the name indicates, is a method utilized to identify thresholds local to a particular area and identify the potential crack pixels by using the classical thresholding mechanism local to a smaller area of the image. The motive of this method is to facilitate the classification of pixels by contrasting the intensities of potential crack pixels from the rest. The image divided into small blocks for this procedure as shown in the figure below [23]
Fig 3.1 Image divided into smaller squares

(a) Original gray scale image
(b) Image divided into square blocks

As evident from partitioning, the total thresholds of the image are stored in an array. Unlike in a single value where only one threshold is stored, localized thresholding becomes more adaptive

<table>
<thead>
<tr>
<th>117</th>
<th>128</th>
<th>114</th>
<th>147</th>
<th>140</th>
<th>147</th>
<th>146</th>
<th>145</th>
<th>146</th>
<th>135</th>
<th>130</th>
</tr>
</thead>
<tbody>
<tr>
<td>141</td>
<td>123</td>
<td>129</td>
<td>150</td>
<td>156</td>
<td>150</td>
<td>131</td>
<td>162</td>
<td>137</td>
<td>131</td>
<td>129</td>
</tr>
<tr>
<td>141</td>
<td>129</td>
<td>137</td>
<td>162</td>
<td>120</td>
<td>140</td>
<td>136</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig 3.2 Thresholds of squares in the first row

The numbers above are thresholds for each block of one row in the image. A 230 × 173 size image is reduced to a 39 × 29 image. The numbers shown above are from the first row of the image and are therefore 29 numbers.
The image that is divided into blocks or squares of sizes depending on the resolution of the image is taken as input from the previous phase and the average value of intensity of each block is calculated which is considered the threshold of each block. The threshold values are localized and are stored separately in an array. The stored values are further processed to extract the cracks from the image. This is shown in the equation below.

For every block of the \( a^{th} \) row \( b^{th} \) column, the threshold is calculated as follows.

\[
\text{Threshold } T_{(a, b)} = \sum_{i=1}^{X} \sum_{j=1}^{Y} I((x * a) + i, (y * b) + j)
\]

Where \( x, y \) are length and breadth of the block.

Apart from the localized thresholding there are other factors to be kept in mind. Using a localized threshold only to classify the pixels is inaccurate since it is just an average value of intensity of a block. If the localized thresholding becomes the only benchmark for classification, every block would contain potential crack pixels. Therefore, other conditions could be specified for pixels to be considered crack pixels.

In the image shown in Fig 3.3., cracks are evident but form only a minor portion of the image. The numbers of blocks to be considered to identify the crack pixels are shown below. The red lines in the image show the cracks on the image [26]. From the figure below, it is evident that the number of blocks to be considered is small compared to the total number of pixels in the image.
Therefore, the localized thresholding contributes partly to the identification of the crack pixels in the image.

3.2.1 Advantage

The advantage of localized thresholding over normal thresholding is that the method is specific to only the smaller squares. In the case of a single threshold for the entire image, only one threshold, obtained by identifying the mean of intensities of all the pixels in the image generalizes the overall intensity.

The crack pixels, which usually are of much lesser intensity than the other pixels are identified with a lot of noise. This is due to the possibility that more than 50% of pixels can be lesser than the intensity of the threshold. For example, one high value among a set of 6 pixels is likely to change the mean value completely and might eventually result in the mean value being higher than the remaining 5 pixels, which leads to inconsistency.
3.3 Contrasting

The motive of the localized thresholding is to separate the crack pixels from the rest of the pixels. This is by raising the contrast between the crack pixels and the rest. By identifying the localized threshold, we can separate the pixels of smaller intensity from the pixels with a relatively higher intensity. The contrast is achieved by increasing the intensities of pixels further by a certain value and reducing the lower intensity pixels by a certain value. The average intensity of the image is taken after the conversion of image into grayscale.

The average intensity of the region of interest is computed twice, once before and once after extracting the region of interest. Therefore, the average intensity of the image is actually the average intensity of the region of interest. The minimum of average intensities before and after contrasting determines the value by which the values of intensities of both potential crack pixels and non crack pixels are changed. The reason for contrasting the image from the pixels of higher intensity to that of lower intensity is better explained by the histograms in the image in Fig 3.1 (a) is shown below

![Histogram of intensities of pixels in the image](image)

(a) Before Contrasting  (b) After Contrasting

Fig 3.4 Histogram of intensities of pixels in the image
The histograms above show the intensities of pixels in the image before contrasting and after contrasting. As evident from the histogram, in Fig 3.4 (a) before contrasting, the intensities of pixels are in the 120 – 150 region and the classification of pixels after separation is evident. The value by which the pixel intensities are incremented or decremented depends on the average intensity of the image after the separation. If the value is constant and the average intensity of the image is very less, pixels other than the crack pixels have a high probability of being decremented to an intensity value close to zero as in the case of the figure below [24].

![Image with relatively lesser intensity](image)

**Fig 3.5 Image with relatively lesser intensity**

This figure contains a high number of pixels that are of less intensity and requires the crack to be detected efficiently. Therefore, we need a smaller value by which the
pixels in the image have to be either incremented or decremented so that more pixels do not attain zero intensity.

### 3.4 Conversion to Black and White Image

After the image is processed, the grayscale image is converted into a black and white image. This can be considered a form of reverse contrast stretching. Contrast stretching is changing the range of intensities of pixels from the original range to a higher range. There are many forms of contrast stretching and the conversion of a gray scale image to a black and white image can be considered to be by using relative lengths. This can be shown by the equation below.

\[
\frac{Z-Z_i}{Z_k-Z_1} = \frac{Z'_i-Z'_1}{Z'_k-Z'_1}
\]

The equation shows the relative position of the intensity of each pixel in the range of the image. The range is 0 – 255 for a grayscale image and 0 – 1 for a black and white image. Here, \(Z\) is the intensity of the pixel in the original image and \(Z'_i\) is the intensity of the same pixel in the image with intensities in the required range.

To convert the image from grayscale to black and white, a threshold value is needed to segregate the pixels to either 0 or 1 in the image. The threshold is a number between 0 and 1 and that can be identified through the histogram and the thresholds of the image found before contrasting and after contrasting. As shown in Fig 3.4 (b) due to the contrasting, a gap of a certain range occurred between intensities which is a ‘null region’. Therefore, a good option is to identify the intensity value in the gap. A fair estimate can be achieved by using the mean intensity value of the image.
The threshold can be a value slightly lesser than the mean intensity of the region of interest since it has the highest probability of lying in the null region. This conversion from gray scale to black and white image identifies the crack pixels and makes them more apparent. The image, when converted into a black and white image, the threshold is used to convert the complete image to white and the values below, the threshold value are black. Therefore, it is necessary that the contrast values are reversed so that the final image contains the crack pixels in white and not black.

One more possibility is the presence of a large number of pixels below the null region. Since this is a contrasting procedure of the mean values, the number of values less than the null region is also important. An upper limit is established for the percentage of total number of pixels to be eligible to be crack pixels. Empirically, the upper limit was established at 10% of the pixels from the region of interest with the least intensity.

When a situation arises such that the number of pixels of lesser intensity is too high, it is important to filter the rest of the pixels from the crack pixels. This is achieved by using a lesser threshold value for the image used to convert the image from gray scale to black and white. The threshold to convert the image from gray scale to black and white is a fraction of minimum of the average intensities before and after contrasting. The fraction varies depending on the value since it is important that the crack pixels are only identified and the rest of the pixels are eliminated.
(a) Image showing crack

(b) Histogram of the image after contrasting

Fig 3.6 Histogram of an image after contrasting
The percentage of the total number of pixels in the region of interest below the threshold must be under a certain value. When a simple fraction to determine the threshold becomes inconsistent, the number of pixels serves as another criterion to establish the threshold. The empirical value for the percentage of maximum permissible number of pixels to be identified as the crack pixels is established at the value 10. The rest of the pixels which might come under noise can be removed during the noise reduction process that is explained below.

3.5 Noise Reduction and Connectivity Establishment

When the image is converted from a gray scale image to a black and white image, there might be an inaccuracy in identifying the threshold for conversion. This might result in a discontinuity of the cracks or noise in the image. This makes it important to identify the continuity between pixels in the image. The procedure is further explained below.

Fig 3.7 Identification of continuity between pixels
As shown in the figure above, the red block indicates the considered pixel and the white blocks indicate the pixels that are white in the black and white pixels. The considered pixel, irrespective of being black or white in the image must have neighboring white pixels. A minimum number of pixels must be white for the pixel to be considered a crack pixel. The rest of the pixels that do not have neighbors in the neighborhood or have lesser number of pixels than the minimum in the neighborhood are removed.

The neighborhood considered here is a $5 \times 5$ neighborhood since the continuity can sometimes not be immediate. The considered area around the each pixel is not lesser than $5 \times 5$ because that would require the pixels around it to be white and the probable gap in the crack due to an inaccurate threshold is not filled. By using this method, the continuity of the cracks is retained and the identification of the crack pixels is shown better for further processing.
Chapter 4

Line Tracing and Crack Representation

After the converting images from gray scale to black and white image, the next phase is the representation of the crack. The crack can be identified on the image through procedures. However, there is a further chance of noise creeping into the image. Finding each pixel and its intensity would not give a proper idea of the nature of the cracks. Therefore, it is imperative that there is another way of representing the crack so that a broader picture is identified better.

4.1 Scan Line Method

The scan line method is used to identify cracks along the image for effective representation. The method is inspired from the scan line method as described by Tim Evans [19]. This method also uses the scan lines for image processing. [19] The scan lines are in the horizontal and vertical direction and the separation between them vary according to the size of the image. This method involves gathering the local minima and identifying the potential crack points and the neighboring minima to identify the probability of the crack.
The proposed method is only used for representation of cracks and is much simpler than utilizing the gradients of values from one single line by comparing adjoining pixels.

(a) Scan Lines in Horizontal Direction

(b) Scan Lines in Vertical Direction

Fig 4.1 Figure showing scan lines
4.2 Watershed method

The watershed method is an example for identifying and representing cracks. This requires a special mention because this method identifies pixels first and then goes through the identification by representation. A modified watershed method has been explained by Tim Evans [19]. The watershed method starts with using a 2-Dimensional image and examining the textures of the image. The image is converted from a 2-D image to a 3-D image based on the texture. This enables the identification of the depth of the image. Then, the ‘water level’ is raised from the bottom to identify the connectivity between the pixels and the neighborhood is looked at for the pixels that are drowned in the ‘water’ and the connectivity is established. The conversion of the image from a 2 dimensional image to a 3 dimensional image is shown in Fig 4.1 below [19].

![Conversion of a 2D image into 3D](image)

**Fig 4.2 Conversion of a 2D image into 3D**
4.3 Representation Methodology

The method implemented in this thesis uses the scan lines as described by Tim Evans [19]. However, this method is employed only on a black and white image. It is a simplified version of the original image where the identification of minima is not a part of the process. In this method, scan lines are only used to identify the set of coordinates of the white pixels in the black and white input image. The process implemented now tries to identify the squares and classify them as crack squares or non-crack squares.

Since the continuity of the pixels is identified distinctly, it can be justified to state that the white pixels are continuous in clusters. This refers to the connectivity establishment procedure described in the previous chapter. The discontinuity between cracks is not considered since it has been eliminated during the connectivity establishment procedure.

![Image with the scan line and the crack pixels](image.png)

Fig 4.3 Image with the scan line and the crack pixels
The image shown above shows the scan line in blue. The white pixels are the crack pixels and the horizontal line shown in blue is the scan line. When the scanning process takes place, this is stored in the form of two clusters. This is identified by the discontinuity of the coordinates of the pixels.

The next step is identifying the midpoint of each cluster. The mean of all the values in each cluster is considered as the midpoint of each cluster. This process is repeated for each scan line in horizontal and vertical direction. The midpoints are identified since they form the centre of crack pixels for each cluster in the image. Therefore, tracing the rest of the crack either horizontal or vertical becomes much easier.

An example of a cluster is shown below.

**Row 1:** 12 13 14 15 16 17 39 40 41 42 43 44 45 46 47 48 49 50 174 175 176 177 178

This is an indication of how the clusters are stored in the row of the scan line. The numbers indicate the y coordinates of the pixels in the case of a horizontal scan line since the x coordinate always remains the same and the x coordinate of the pixels in the case of a vertical scan line since the y coordinate always remains the same. The clusters need not be separated since their continuity is identified by their neighboring value in the array. The midpoint therefore becomes an appropriate benchmark to identify the cluster and a probable crack.

The mid points are stored in different arrays along scan lines according to the clusters. The total number of clusters on each scan line is also the total number of mid points on each cluster. The next step is matching the midpoints with the mid points of the clusters of the next scan line. The process continues from top to bottom for horizontal
scan lines and left to right for vertical scan lines. The midpoints of the current scan line are matched with the midpoints of the clusters of the next scan line. The reference taken for this process is the values of mid points since the midpoints actually denote the coordinates of its location. The difference between midpoints is more than the distance between the scan lines, they are not considered.

However, since the depiction is not just the midpoints but also the other pixels between them that are to be kept in mind, the values need to be represented in the vertical direction. If the pixels cannot be captured by the horizontal scan lines, they can be captured eventually in the vertical direction.

4.3.1 Noise Reduction after compression

Noise reduction by verification of continuity becomes very crucial during the implementation phase of the process. When the procedure and conversion into black and white image is completed and the continuity is established, there are chances that patches or blocks of data that are small and insignificant might still be left. However, these blocks of data, which might not be evident and insignificant in the original image, become significant when compressed.

The neighborhood of individual blocks of data is checked for continuity. However, this happens in a smaller yet more significant scale since the image has been scaled down by several times. Therefore, verifying each pixel for neighbors in a $3 \times 3$ neighborhood would help segregate and thereby eliminate the noise pixels efficiently.

One more condition that is to be taken into consideration is the number of crack pixels in the compressed image. Smaller the number of crack pixels in the image, lesser
the threshold for number of pixels in the neighborhood. If an image contains a small fraction of the pixels as white, it would evidently be pointless to try and further employ a strict noise reduction procedure for the original data might be removed. Therefore, the threshold is reduced considerably for exceptional cases when blocks of noise are still left behind.

4.4 Classification using Hough Transform

After all the procedure is completed, the next phase is to use the obtained image to classify the crack. A classic approach of using the Hough Transform is employed for crack classification. The effective representation of shapes in normal form is the feature of the Hough Transform and is also considered to be one of the most accurate methods to identify shapes in an image.

4.4.1 Hough Transform

Hough Transform is widely used in image processing and signal processing and is considered to be one of the best, most accurate ways of detecting shapes in an image. The commercial applications of Hough transform in the image are identifying a fetal head in ultrasound scanning methods and meteorite impact craters. It can even detect the shapes of objects when overlapped.

The Hough Transform is representation of a set of points in a coordinate system in a parametric form. In the case of a line, the shortest distance from the line to the origin is calculated and the line is represented in the form shown below [25].

\[ r (\theta) = x_o \cos(\theta) + y_o \sin(\theta) \]
Where \( r \) is the shortest distance of the line from the origin and \( \theta \) is the angle of the vector joining the closest point on the line to the origin and \( x_0 \) and \( y_0 \) are the \( x \) and \( y \) coordinates of points in the plane. For a more complex structure like a circle, the Hough Transform can be used by representing the parameters of the circle such as radius and centre and represent them in a normal form. A curve on the other hand can be considered a part of a circle or an ellipse and a continuously varying radius and centre may also be represented accordingly using the Hough Transform. Despite the restrictions of domain, the Hough Transform also is impervious to gaps or noise in the image.

Another major advantage of the Hough transform is that since its representation completely retains the original data of the image and also eliminates noise when a reversal of Hough transform is performed by calculating the coordinates of each of the points by the transform in the image.

**4.4.2 Proposed Methodology**

Hough Transform is applied directly on the image obtained from the previous phase and the cracks can be identified easily since the image is black and white only and can therefore be used efficiently to identify the cracks in the image.

Hough Transform shows the image in the form of a wave form. The image obtained from the Hough Transform can be used to identify the number and slopes of cracks by looking at the number of clusters in the image. If the image shows clear white clusters in the image, it indicates the presence of cracks and their position in the image reflects the location and orientation of the cracks. Since the coordinates of \( \theta \) and \( r \) are
represented in the x and y coordinates of the transform respectively, the position and orientation can be identified very easily.

The results for the procedure are explained in the next chapter.
Chapter 5

Tests and Results

The process was performed in Matlab© which was conducive for all the procedures. Using the RGB components was made much easier and the operations with arrays have been represented in a pictorial format easily.

5.1 Complete Method

The method described to extract the cracks from pavements is implemented in 3 phases. They are Background separation, Thresholding and Compression and representation. The background separation is undergone by using the standard deviation of the Red, Green and Blue components of the pixels of image and verifying the closeness of components of the image.

The next phase is the localized thresholding where contrasting is done by increasing and decreasing values of pixels respectively depending up on the local threshold of the blocks. The image is then converted into a black and white image depending on the average value of region of interest. Noise reduction is performed by verifying the neighborhood of pixels and looking for white pixels in the neighborhood.
Scan lines are later used to identify the continuity of crack blocks and are represented in the corresponding compressed image where each pixel represents a square formed by the horizontal and vertical scan lines. Finally, Hough transformed is applied to represent the cracks accordingly. The full procedure is described in the flow chart as shown below.

Fig 5.1 Flow chart showing the full procedure
5.2 Image with less or no background

The results were obtained by processing the image from [23] for which the processing is done extensively. This image is probably the ideal image for the process. This is the easiest image to be processed and the result is just as clear since there is not much to be separated from the background and therefore, the result can be obtained more accurately. The original image is as shown below.

![Image with no background](image.png)

**Fig 5.2 Image with no background**

(a) Original image

After the image is obtained, it undergoes a grayscale conversion and the background separation algorithm that results in elimination of some part of it. Although it affects the image, it removes only a small part of it and does not tamper with the crack at all. This is shown in the two figures below.
The next figure shows the results after identifying the continuity of the blocks separated as background and that results in a further elimination of the nodes as shown below.

**Fig 5.2 (b) Gray Scale**

The next phase is the localized thresholding. This phase separates the crack pixels from the rest by increasing the intensity of the rest of the pixels and decreasing the intensity of the crack pixels. The results obtained are shown below.

**Fig 5.2 (c) Background separation**

The next phase is the localized thresholding. This phase separates the crack pixels from the rest by increasing the intensity of the rest of the pixels and decreasing the intensity of the crack pixels. The results obtained are shown below.
Fig 5.2 (d) After Contrasting

The image is converted into the black and white image from a gray scale image. Apparently, the image has the background and the rest separated appropriately. The white pixels is the rest of the image and the black pixels are classified the crack.

Fig 5.2 (e) Conversion into a black and white image
The intensities of the image are reversed so that the compression phase becomes much easier.

Fig 5.2 (f) Reversing the intensities

The phase to search for continuity is shown below. It shows a smoothened version of the image. Yet, that does not make a difference since the compression only depicts the nodes and not the rest.

Fig 5.2 (g) Identifying continuity
However, in the case of classical thresholding method, the threshold established is quite inconsistent considering the size of the image and the range of intensities of pixels for which the threshold is applied. A threshold is identified for the region of interest for an image that has undergone median filtering. The result is as shown below.

Fig 5.2 (h) Image converted to black and white using classical thresholding

As evident from the above image, the image undergoes thresholding and can identify cracks but the amount of noise in the resulting image is very high.

The image is later compressed to a smaller image. The size of the compressed image and the scale of compression depend on the distance between each scan line in the image.

The compressed image and its expanded version are shown below.
Fig 5.2 (i) Compressed Image

The image is compressed to a size of $29 \times 39$ from a $173 \times 230$ image.

Fig 5.2 (j) Expanded version of the compressed image after noise reduction

Fig 5.1 (k) Hough Transform of Image
The result from the Hough Transform shows that the image contains a single large crack due to a single big cluster in the image and also, it is closer to $0^\circ$ which indicates the presence of a vertical crack.

### 5.2.1 Image with compound cracks

In the image with more than one crack, the method still yields good results. The sequences of images after background separation are shown below.

![Fig 5.3(a) Background separation](image-url)
This image shows the contrasting method to separate the potential crack pixels from the rest of the image.

Fig 5.3 (b) After contrasting
Fig 5.3 (c) Converting into black and white image
Fig 5.3 (d) Reversing the intensities
Fig 5.3 (c) Identifying Continuity
Fig 5.3 (f) Expanded version of compressed Image

Fig 5.3(g) Hough Transform of Image
The Hough transform indicates the presence of a number of cracks that are vertical due to the presence of clusters at around 0°. The presence of curved lines is explained by the presence of extended clusters and not just dots in the transform.

More than one image is processed to verify the credibility of the process.

Fig 5.3 (h) Image in grayscale
Fig 5.3 (i) Image after contrasting

The image after contrasting is converted to black and white and the potential crack pixels are identified.

Fig 5.3(j) Black and White Image
Fig 5.3 (k) After reversing contrast

Continuity of the pixels are verified in the image as shown below.

Fig 5.3 (l) Verifying continuity
When the number of crack pixels is less when compared to the pixels in the image, the noise reduction procedure adopts a lenient threshold. This is explained as shown below.

![Expanded version of compressed Image before noise reduction](image)

**Fig 5.3 (m) Expanded version of compressed Image before noise reduction**

The image undergoes noise reduction with a smaller minimum number of white pixels in the neighborhood and therefore looks quite similar to the image before noise reduction.
Fig 5.3 (n) Expanded version of compressed image after noise reduction

The hough transform of the image shows the presence of a number of cracks in the horizontal and vertical direction. The whiter lines and dots in the Hough transform are indicative of the presence and orientation of cracks. The presence of horizontal white lines in the Hough transform shows the presence of horizontal cracks in the image.

Fig 5.3(o) Hough Transform of Image
5.3 Image with background

This is the second case where the image contains background in some part of it. Although this image does not contain a background around the pavement, the presence of unwanted data still exists and the region of interest is still extracted. The image is shown below [27].

![Image with some background](image.png)

**Fig 5.4 Image with some background**

(a) Original Image

The image, when converted to grayscale is shown below. However, this would not be an appropriate image for crack detection as it contains the background and that can influence the outcome of the image.
The image, after the median filtering and background separation are done, is as shown below. The separated background can also be seen below.

Fig 5.4 (c) After Background Separation

The images obtained after each step is done sequentially are shown below.
Fig 5.4 (d) After Thresholding

The image is converted to black and white using a threshold and determining the potential crack pixels.

Fig 5.4 (e) After Conversion into black and white and reversing intensities
The continuity of cracks is verified by identifying the neighborhood of the crack pixels as shown below.

![Fig 5.4 (f) Verifying Continuity](image)

The image from original size is compressed to $39 \times 52$ from $197 \times 256$ as a result of the compression.

![Fig 5.4 (g) Expanded version of the compressed image](image)
The hough transform of the image shows virtually a horizontal highlighted line indicative of a horizontal crack.

Fig 5.4 (h) Hough Transform of Compressed Image
Chapter 6

Discussion and Conclusion

This chapter gives a complete overview of the procedure discussed and paves the way for future research for better crack detection and usage of similar procedures for crack identification on different forms of images. This chapter also shows a new dimension by taking not just the pavement images but also random images that might contain pavements for crack detection thereby eliminating the repetitive acquisition of data due to faulty images.

6.1 Conclusion

The three phases of background separation namely filtering, the crack detection with localized thresholding and the representation of data by compression described in the procedure. The background separation enables a new dimension of identifying pavement images. Although texture has been used as a benchmark before, the apparent texture of the pavement can be different in different pictures and becomes hard to recognise. Therefore, a basic procedure of distribution of components among pixels by statistical analysis to identify the pavement and separate it from the background is implemented.
Using localized thresholding to separate probable crack pixels and not to identify the cracks completely gives a better separation procedure from the crack pixels and the rest of the image. Division of the image into smaller nodes that are of variable sizes paves a way for further representation of the image when converted to a black and white image.

The final phase of compressing the original image into a smaller black and white image facilitates better storage of data by reducing the image size by a number of times. This results in a better storage and reduces the complexity of the result.

In conclusion, the elimination of noise periodically throughout the procedure finally results in less or no noise in the end image and the usage of basic methods for computation means that the processing speed is comparatively lesser which makes the method less complex.

6.1.1 Limitations

The limitation to this procedure is that it cannot detect potholes. Since the image uses mid points of clusters, a large circular patch of land that is supposed to be detected might be detected as a line only.

6.2 Future Work

The shown process opens a new dimension of considering any kind of color images that might be taken but can still give an accurate result. The future research can be the usage of different forms of color, types of images like an HSV image and other properties to identify the location of the pavement and the cracks further into the process.
Using artificial intelligence to establish rules to identify the background by analyzing the pattern of the image and rate the cracks accordingly can be a better means of identifying cracks as well as differentiating potholes from circular patches on pavements and can be expected from further research.
References


[9]. Bugao Xu and Yaxiong Huang “Automated Pavement Cracking Rating System: A Summary” CENTER FOR TRANSPORTATION RESEARCH THE UNIVERSITY OF TEXAS AT AUSTIN Project Summary Report 7-4975-S Project 7-4975


[14]. **Martin Herold, Dar Roberts, Val Noronha, Omar Smadi** “Imaging Spectrometry and Asphalt Road Surveys” *Transportation Research Emerging Technologies Volume 16, Issue 2, April 2008, Pages 153-166*


APPENDIX - Matlab Code

% Convert and store image in grayscale
I9 = .2989*crack(:,:,1)+.5870*crack(:,:,2)+.1140*crack(:,:,3);
% Identifying RGB components, square sizes
crack1 = crack(:,:,1);
crack2 = crack(:,:,2);
crack3 = crack(:,:,3);
I = medfilt2(I9);
[a,b] = size(I);
A1 = zeros(1,64);
A = zeros(1,64);
B = zeros(1,64);
red = zeros(1,64);
green = zeros(1,64);
blue = zeros(1,64);
% Identifying the size of squares depending on the image
if((a >300) && (b>300))
    sx = 8;
sy = 8;
elseif((a>256) && (b>256))
    sx = 7;
sy = 7;
else
    sx = 6;
sy = 6;
end
% Initialize variables to calculate the total of R, G and B components
thresh = zeros(a/sx,b/sy);
thresRGB = zeros(a/sx,b/sy);
pixels = a*b;
l=1;
m=1;
x=0;
y=0;
thres1 = 0;
thes = 0;
thresR = 0;
thresG = 0;
thresB = 0;
I1 = I;
ss = sx * sy ;
% Background separation
for x = 0:sx:a
    for y = 0:sy:b
        for i=1:sx
            for j=1:sy
                for k = 1:ss
                    if (((x+i) <= a) && ((y+j) <= b))
                        A(1,k) = I(x+i,y+j);
                        red(1,k) = crack1(x+i,y+j);
                        green(1,k) = crack2(x+i,y+j);
                        blue(1,k) = crack3(x+i,y+j);
                    end
                end
            end
        end
    end
end

% Calculate the mean of R, G and B components in each node
for i=1:ss
    thres = thres + A(1,i);
    thresR = thresR + red(1,i);
    thresG = thresG + green(1,i);
    thresB = thresB + blue(1,i);
end
threshold = thres/ss;
thresholdR = thresR/ss;
thresholdG = thresG/ss;
thresholdB = thresB/ss;

% Calculate the standard deviation between means of each component
De(1) = thresholdR;
De(2) = thresholdG;
De(3) = thresholdB;
thresRGB(1,m) = std(De);
m = m+1;
m1 = m;
end
l = l+1;
l1 = l;
m = 1;
end
l=1;
m=1;
tval = 0;
[rnum,cnum] = size(thresh);
for i = 1 : rnum
    for j = 1 : cnum
        tval = tval + thresh(i,j);
    end
end
size21 = rnum * cnum;
finthres = tval / size21;

% Whiten the nodes that have a high standard deviation
for g = 1:rnum
    for h = 1:cnum
        if((thresRGB(g,h) > 10) && (thresh(g,h)> finthres))
            for i = 1:sx
                for j = 1:sy
                    I((((g-1)*sx)+i),(((h-1)*sy)+j)) = 255;
                end
            end
        end
    end
end
end
figure,imshow(I,[],);

% Verify for continuity of background
for x = 1:rnum
    for y = 1:cnum
        count = 0;
        for i = -2:2
            for j = -2:2
                if((x+i <=rnum) & (y+j <=cnum) & ((x+i) >= 1) & ((y+j) >= 1))
                    if((thresRGB(x+i,y+j)) > 10 & (thresh(x,y) > finthres))
                        count = count + 1;
                    end
                end
            end
        end
        % If the number of neighboring nodes that are white is more than a certain value, then
        % erase the node
        if(count>2)
            for i = 1:sx
                for j = 1:sy
                    I((((x-1)*sx)+i),(((y-1)*sy)+j)) = 255;
                end
            end
        end
    end
end
% Contrasting
% Identifying the value by which the intensities of pixels are changed
if (finthres >= 120)
    changeval = 50;
elseif (finthres >= 110 && finthres < 120)
    changeval = 45;
elseif (finthres > 100 && finthres < 110)
    changeval = 40;
elseif (finthres < 100 && finthres >= 90)
    changeval = 30;
elseif (finthres < 90 && finthres >= 75)
    changeval = 20;
elseif (finthres < 75 && finthres >= 60)
    changeval = 15;
elseif (finthres < 60)
    changeval = 10;
end
% Altering the values of pixels
for x = 1:a-sx
    for y = 1:b-sy
        for i=0:sx
            for j=0:sy
                if I(x+i,y+j) <= thresh(l,m)
                    I1(x+i,y+j) = I(x+i,y+j) - changeval;
                else
                    I1(x+i,y+j) = I(x+i,y+j) + changeval;
                end
                m = m+1;
            end
            l = l+1;
            m = 1;
        end
        l = 1;
    end
    figure,imshow(I1,[]);
end
% Verify maximum permissible threshold for intensity
[number,intensity] = imhist(I1);
RI = a * b / 10;
numbers = 0;
for i = 1:255
    numbers = numbers + number(i);
    if(numbers < RI)
        intbord = i;
    end
end
intbord = intbord/255;

% Localized thresholds after contrasting are identified
for x =0:sx:a
    for y = 0:sy:b
        for i=1:sx
            for j=1:sy
                for k = 1:ss
                    if(((x+i) <= a) && ((y+j) <= b))
                        A1(1,k) = I1(x+i,y+j);
                    end
                end
            end
        end
    end
    k = 1;
end
for i=1:ss
    thresl = thresl + A1(1,i);
end
threshold1 = thresl/36;
thresl = 0;
thresh1(l,m) = threshold1;
m = m+1;
m1 = m;
end
l = l+1;
l1 = l;
m = 1;
end
for i = 1 : rnum
    for j = 1 : cnum
        if(thresh1(i,j) ~= 255)
            tval1 = tval1 + thresh1(i,j);
        end
    end
end
size21 = rnum * cnum;
finthres1 = tval1 / size21;
if (finthres1 > finthres)
    finthres1 = finthres;
end

% Mean value of the intensities in region of interest is used to find a threshold to convert
image to black and white
if (finthres1 >= 120)
    bord = 0.17;
elseif (finthres1 >= 110 && finthres1 < 120)
    bord = 0.13;
elseif (finthres1 > 100 && finthres1 < 110)
    bord = 0.11;
elseif (finthres1 < 100 && finthres1 >= 90)
    bord = finthres1 / 4;
elseif (finthres1 < 90 && finthres1 >= 75)
    bord = 0.15;
elseif (finthres1 < 75)
    bord = 0.05;
end
while (bord > intbord)
    bord = bord * 0.75;
end

% Convert to black and white image
I2 = im2bw(I1,bord);
figure, imshow(I2,

for i= 1:a
    for j=1:b
        I3(i,j) = 1 - I2(i,j);
    end
end

figure, imshow(I3,

% Verify for continuity by verifying the neighborhood of pixels
for x = 3:a
    for y = 3:b
        count = 0;
        if(I3(x,y) == 0)
            for i = -2:2
                for j = -2:2
                    if( ((x+i) <= a ) && ((x+i) >= 1) && ( (y+j) <= b ) && ((y+j) >=1) )
                        if(I3(x+i,y+j) == 1)
                            count = count + 1;
                        end
                    end
                end
            end
        end
    end
end
if(count>2)
I4(x,y) = 1;
else
I4(x,y) = I3(x,y);
end
end
end

figure, imshow(I4,[]);

% Acquire data using scan lines
[a,b] = size(I3);
x = 1;
y = 1;
count = 0;
for i = 1:5:a
    for j = 1:b
        if(I3(i,j) == 1)
            Hor(x,y) = j;
            y = y + 1;
        end
    end
    x = x + 1;
y = 1;
end

for j = 1:5:b
    for i = 1:a
        if(I3(i,j) == 1)
            Ver(x,y) = i;
            x = x+1;
        end
    end
    y = y + 1;
x = 1;
end

% Identifying the midpoints of clusters
[xh,yh] = size(Hor);
xv,yv] = size(Ver);

for i = 1:xh
    count = 0;
    row = 1;
k = 0;
    if(Hor(i,1) == 0)
        Hmid(i,1) = 0;
    else
        Hmid(i,1) = 1;
    end
    % Additional code for identifying midpoints
end
if (Hor(i,1) ~= 0)
    for j = 1:yh-1
        count = count + Hor(i,j);
        k = k + 1;
        if( ( (Hor(i,j+1) - Hor(i,j)) ~= 1 ) || (j == yh-1))
            Hmid(i,row) = round(count/k);
            row = row + 1;
            count = 0;
            k = 0;
        end
    end
end
end
end
end
for j = 1:yv
    count = 0;
    col = 1;
    k = 0;
    if(Ver(1,j) == 0)
        Vmid(1,j) = 0;
    else
        if (Ver(1,j) ~= 0)
            for i = 1:xv-1
                count = count + Ver(i,j);
                k = k + 1;
                if(Ver(i+1,j) - Ver(i,j) ~= 1)
                    Vmid(col,j) = round(count/k);
                    col = col + 1;
                    count = 0;
                    k = 0;
                end
            end
            if(i == xv-1)
                if(Ver(i+1,j) ~= 0)
                    count = count + Ver(i+1,j);
                    k = k + 1;
                    Vmid(col,j) = round(count/k);
                    count = 0;
                    k = 0;
                end
            end
        end
    end
end
end
end
end
Identifying the crack blocks by identifying the nearest midpoints from the considered scan line to the next scan line

```matlab
[a1,b1] = size(Hmid);
[a2,b2] = size(Vmid);
sqrep = zeros(xh,yv);
slope = 0;
diff1 = 0;
k = 1;
x = 0;
y = 0;
for i = 1:a1
    for j = 1:b1
        for k = 1:b1
            if (i+1 <= a1)
                val = Hmid(i,j) - Hmid(i+1,k);
                y = round(Hmid(i,j)/5);
                if (y == 0)
                    y = 1;
                end
            else
                if ((val <= 5) && (val > -5))
                    sqrep(i,y) = 1;
                    if(y+1 <= a1)
                        sqrep(i,y+1) = 1;
                    end
                elseif ((val <= -5) && (val >= -10))
                    sqrep(i,y) = 1;
                end
            end
        end
    end
end
for j = 1:b2
    for i = 1:a2
        for k = 1:a2
            if(j+1 <= b2)
                val = Vmid(i,j) - Vmid(k,j+1);
                x = round(Vmid(i,j)/5);
                if(x == 0)
```

77
x = 1;
end
if( (val <= 10) && (val > 5))
    sqrep(x,j) = 1;
    if(x+1<= a2)
        sqrep(x+1,j) = 1;
    end
elseif ((val <= 5) && (val > -5))
    sqrep(x,j) = 1;
elseif ((val <= -5) && (val >= -10))
    sqrep(x,j) = 1;
    if(x-1 >= 1)
        sqrep(x-1,j) = 1;
    end
end
end
end
end
end
end
[xs,ys] = size(sqrep);
for i = 1:xs
    sqrep(i,1) = 0;
end
for j = 1:ys
    sqrep(1,j) = 0;
end
figure,imshow(sqrep);
%Noise reduction in compressed image by verifying the neighborhood nodes in the compressed image
c1 = 0;
xy = xs * ys;
sqrepl = sqrep;
for i = 2:xs-1
    for j = 2:ys-1
        count = 0;
        if(sqrep(i,j) == 1)
            c1 = c1 + 1;
        end
        for l = -1:1
            for m = -1:1
                if(sqrep(i+l,j+m) == 1)
                    count = count + 1;
                end
            end
        end
    end
end
if (c1 < (xy/20))

78
c = 1;
else
    c = 2;
end
if(count < c)
    sqrep1(i,j) = 0;
end
end
figure, imshow(sqrep1);
[x1,y1] = size(sqrep1);
k = 1;
x = 1;
y = 1;
count = 0;
for i = 1:x1
    for j = 1:y1
        if(sqrep1(i,j) == 1)
            SHmid(x,y) = j;
            y = y + 1;
        end
    end
    x = x + 1;
y = 1;
end
for j = 1:y1
    for i = 1:x1
        if(sqrep1(i,j) == 1)
            SVer(x,y) = i;
            x = x+1;
        end
    end
    y = y + 1;
x = 1;
end
%Hough Transform of the final image
[H,T,R] = hough(sqrep1);
subplot(2,1,2);
imshow(imadjust(mat2gray(H)),'XData',T,'YData',R,...
    'InitialMagnification','fit');
subplot(2,1,2);
imshow(imadjust(mat2gray(H)),'XData',T,'YData',R,...
    'InitialMagnification','fit');
title('Hough transform of sqrep');
xlabel('\theta'), ylabel('\rho');
axis on, axis normal, hold on;