

Fast Segmentation of Industrial Quality Pavement Images using Laws Texture Energy Measures and k -Means Clustering

Senthan Mathavan,^a Akash Kumar,^b Khurram Kamal,^b Michael Nieminen,^c Hitesh Shah,^c Mujib Rahman^d

^aNottingham Trent University, School of Architecture, Design and the Built Environment, Nottingham NG1 4BU, UK

^bNational University of Sciences and Technology, College of Electrical and Mechanical Engineering, Islamabad, Pakistan

^cFugro Roadware, 2505, Meadowvale Boulevard, Mississauga, Ontario L5N 5S2, Canada

^dBrunel University, Department of Civil Engineering, Uxbridge UB83PH, UK

Abstract. Thousands of pavement images are collected by road authorities daily for condition monitoring surveys. These images typically have intensity variations and texture non-uniformities making their segmentation challenging. The automated segmentation of such pavement images is crucial for accurate, thorough and expedited health monitoring of roads. In the pavement monitoring area, well known texture descriptors such as gray-level co-occurrence matrices and local binary patterns are often used for surface segmentation and identification. These, despite being the established methods for texture discrimination, are inherently slow. This work evaluates Laws texture energy measures as a viable alternative for pavement images for the first time. k -means clustering is used to partition the feature space, limiting the human subjectivity in the process. Data classification, hence image segmentation, is performed by the k -nearest neighbor method. Laws texture energy masks are shown to perform well with resulting accuracy and precision values of more than 80%. The implementations of the algorithm, in both MATLAB and OpenCV/C++, are extensively compared against the state of the art for execution speed, clearly showing the advantages of the proposed method. Furthermore, the OpenCV based segmentation shows a 100% increase in processing speed when compared to the fastest algorithm available in literature.

Keywords: pavement, surface inspection, condition monitoring, texture analysis, image processing, transportation.

Address all correspondence to: Senthan Mathavan, School of Architecture, Design and the Built Environment, Nottingham Trent University, Burton Street, Nottingham NG1 4BU, UK (phone: +31 686 396969; e-mail: s.mathavan@ieee.org).

1 Introduction

The total length of paved and unpaved roads, in the world is about 64 million kilometers, according to a 2013 estimate [1]. The proportion of paved roads is constantly on the increase, mainly due to infrastructure development in the developing countries, and around 65 % of the total road network is paved according to a 2009 estimate of the World Bank [2]. Using the two

data sources, it is safe to say that at least 40 million kilometers of the current world's roads are paved. Highway-related construction in Europe involves \$110 billion of public money alone yearly, whereas it costs the US government \$80 billion annually [3]. In addition to this, there is a considerable amount of private investment as well.

With the ever increasing economic activity all around the world, the burden on the transportation infrastructures is continually on the rise. While rural areas possess the possibility of expanding their traffic handling capacity - either by performing pavement widening or creating parallel routes - the urban and semi-urban settings present a real problem for expansion. Meanwhile, the urban roads also have to handle constantly increasing traffic volumes.

Condition surveys are vital for pavement deterioration monitoring, hence for the maintenance of a healthy road network. These assessments, performed manually in the first part of 20th century, have seen considerable amount automation in the last few decades due to the millions of kilometers of roads needing inspection [1-3]. The automation of the surveys has not only enabled the upkeep of the growing road networks of the rural geographies, it has also facilitated the increasing frequency at which the urban roads have to be monitored owing to more traffic.

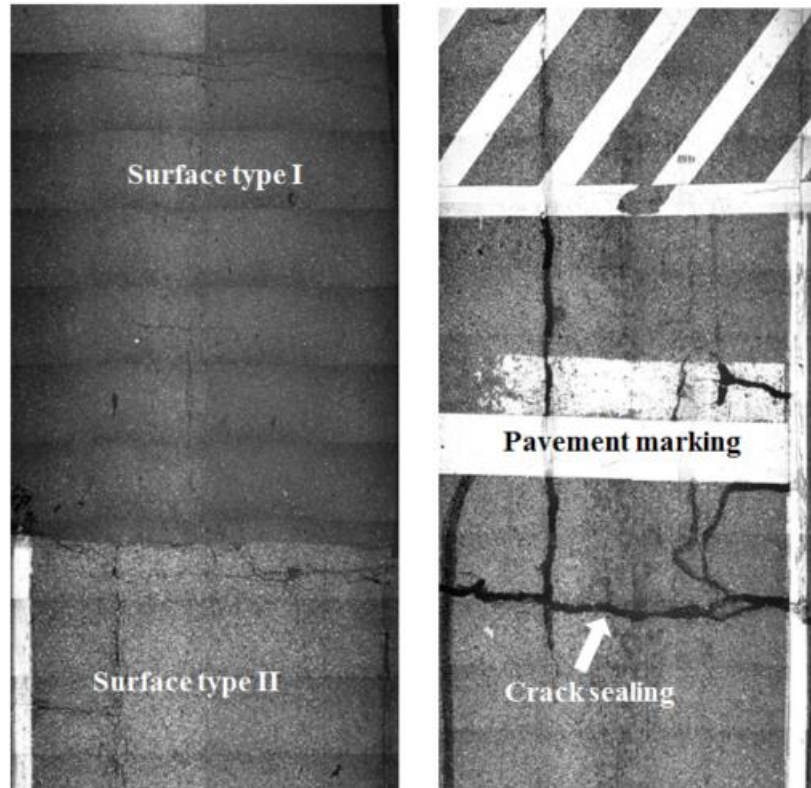


Fig. 1 Two images from a pavement survey of Fugro Roadware.

Automated pavement assessments use a wide range of sensors, such as cameras, X-rays and load cells, to evaluate the structural condition of roads [4]. Visual surveys, using 2D imaging cameras and 3D laser profilers, are very common in the transportation infrastructure management. Many pavement distresses, such as cracks, potholes and raveling [5], can be detected by automated visual surveys. Although 3D imaging systems are extensively employed to measure pavement distresses, 2D camera based systems provide a low-cost method to assess pavements, hence are incorporated into many industrial systems, e.g. ARAN7000 from Fugro Roadware and LRIS from Pavemetrics, Inc. This work exclusively treats the segmentation of images obtained by 2D industrial camera systems. Figure 1 shows two such images obtained in a survey by Fugro Roadware, Canada. Four distinct surface types are identified in the two images in Figure 1.

There are a number of pavement types; construction materials and methods vary between countries and even between different regions within a country. Asphalt (bituminous) and concrete are two of the most common pavement construction materials. Asphalt pavements consist of aggregates held together by a binder, whereas concrete roads are made up of cement, aggregates and sand. Different pavement material bases give rise to different types of pavement distresses, in general [5]. In addition, conditions evaluating distress severity, such as treatment or failure thresholds, are tightly coupled to the base material and the method of construction of a pavement. This dependency mainly arises from different strengths and failure mechanisms associated with the materials and methods of different pavement types. Hence, automatically differentiating the underlying material type from survey images is very critical for pavement monitoring and analysis. The automated differentiation of various surfaces is known as segmentation in image processing, where a collection of pixels with similar properties are assigned a common label [6]. Image segmentation can be extended, with some level of human intervention, to detect various objects in the pavement images, such as a concrete surface, white markers, grass, etc (this process is known as image classification).

It was said earlier that certain distresses are found only in specific pavement types, e.g. asphaltic or concrete. In addition, detection algorithms are distress-specific. Given these facts, if the homogenous regions in the images can be classified, the detection algorithms for such distresses can efficiently limit their search to the precise areas identified, saving computational effort. The computational savings will be enormous considering that millions of kilometers of roads are imaged every year. Moreover, distresses that span a considerable area of the pavement, like potholes and raveling, present the possibility of direct detection with region segmentation and identification. For example, potholes that usually have a rubble-like material will be segmented

as a unique region by the procedure, making its direct detection, and possibly, measurement feasible. Another advantage of segmenting different pavement types is that it will lead to detecting the wear of individual surfaces, which is a measure of the quality of pavement. Furthermore, a pavement image segmentation procedure can also assess structurally non-important defects such as worn-off white line markings. In this regard, the maintenance of pavement markings constituted a \$2 billion industry in the US in 2007 [7], making their assessment key. Since pavement markers are supposed to have well defined geometries, like straight lines, arrows, and numbers painted in constant thickness lines, a pavement image segmentation scheme can be extended to evaluate their condition as well.

As hundreds of kilometers of roads are surveyed every day, the volume of acquired imagery is very large. This necessitates the application of a fast algorithm for the segmentation, and preferably classification, procedure. In this paper, Laws texture energy measures [8] are utilized to generate features in order to segment and classify pavement images obtained by a road monitoring system. An unsupervised data partitioning algorithm, called k -means clustering (a.k.a. C-means clustering), is used on a number representative images to identify the feature values corresponding to various surfaces in the images [9]. Subsequently, another technique, known as the k -nearest neighbors algorithm, is employed to classify any new images.

2 Literature Review

2.1 Image Segmentation

Image segmentation is essentially the process where objects are separated from the background and from one another [10]. For pavement images there is no background, in general, unless the imaging system also captures the ground, grass, etc at the side of the roads. In addition, there are

no objects with definitive shapes in these images. Hence, the target here is to identify different homogeneous regions in the image, such as different pavement material, line markings and defects. In addition, any other surfaces types, such as gravel and grass that belong to the non-paved areas may also appear in images and thus have to be clearly segmented and identified.

A number of methods are used to segment images. Thresholding, edge-based methods and region growing techniques are some of the commonly used procedures [11]. Particular techniques exist for color images [12]. The list of all employed texture segmentation techniques is exhaustive. A region based approach using texture properties is exploited in this paper. Among the number of approaches used for region based segmentation [13], the current method uses a predetermined size of smaller regions to calculate texture properties leading to the identification of different surfaces.

2.2 Texture Analysis

Texture is defined as “something consisting of mutually related elements”, leading to the consideration of a group of pixels than a single pixel [10]. The fundamental collection of pixels intensities that represent the surface of an entity are known as a texture primitive or texture element. The underlying idea of texture analysis is that by identifying texture primitives, the underlying surface can be distinguished or identified. Statistical, structural, model based and transform approaches are available for texture segmentation [14]. Gray level co-occurrence matrices (GLCMs) are the most commonly used statistical texture features. Wavelets and Gabor are also some of the transforms frequently employed to differentiate textures. Fractal and stochastic analysis fall under the category of model based texture discrimination methods [15]. Laws texture energy measures is a statistical technique [8]. The list of analysis methods is

exhaustive. For an extensive treatment of texture analysis methods, refer to a review such as that of Materka and Strzelecki [14].

2.3 Pavement Image Segmentation

Although pavement monitoring involves imaging millions of kilometers of roads every year, there has not been much effort on automated image analysis to segment different road regions. Pavement crack detection has been the main concentration of the research community [16]. In another detailed work [17], many texture descriptors, including Laws feature energy measures, have been experimented with to detect pavement distresses and wear directly but it only treats images acquired under very controlled conditions with uniform image intensities across the images. However, the paper does not consider image segmentation [17]. In relation to pavement segmentation, Kapur entropy has been applied to segment aggregates from the binder layer under laboratory conditions, without giving much emphasis to identifying different surfaces encountered in real roads [18]. Pavement markings have been specifically segmented in other works, using edge pairing and neural networks [19], using Laws features in conjunction with Gabor filter [20] and edge detection with beamlets [21]. However, none of the three works do consider segmenting different regions of pavement images uniquely.

Few other works have considered the segmentation of different road surfaces. 3D stereo images have been used to classify 9 different road surface types under laboratory conditions [22]. However, 3D images usually contain richer information when compared to 2D intensity images, in general, hence the surface classification is easier with the former. 3D imagers also have the advantage of not being affected by shadows, intensity variations in their fields of view. The laboratory conditions also provide very stable lighting conditions, in general, setting up an ideal scenario for analysis. In a similar work, images of pavement, taken under two different lighting

conditions, known as dual light inspection, have been used to identify various surfaces and distress types [23]. However, imaging the same road area, under two different lighting conditions, from the same vantage point is practically difficult to achieve from a mobile inspection platform. As such, this technique does not appear to be feasible for industrial scale monitoring. Multi-resolution GLCMs have been used in conjunction with a hierarchical process for surface classification [24]. GLCM have been used with self-organizing map to segment two pavements consisting of different sized aggregates [25]. A multitude of features, including GLCM, local binary patterns, and Fourier components are employed with a support-vector machine to classify 10 different types of pavements [26]. In an extensive reporting, they also report on the processing overheads; for a 255x255 pixels image, their algorithm with the best classification performance takes about 125 ms to segment [26].

For a domain that must be among the top few in terms of the amount of images collected for analysis, and given the fact that very fast analysis is necessary to conclude on the condition of the pavement, very few research studies exist for pavement image segmentation. Moreover, most of the works, as seen earlier, are centered on GLCM, not the fastest texture discrimination technique available, as seen later in this paper.

3 Theory

This section explains some of the image processing and classification techniques on which the current paper is based.

3.1 Adaptive Histogram Equalization

For industrial images acquired under natural or in not-so-well controlled lighting conditions, some form of pre-processing is necessary. For the left image in Figure 1, one can easily see that

in the bands belonging to single exposure photographs, which are stitched together to form the whole image, intensity variations are found from top to bottom along the transverse direction of the road. The top part and the side areas of the image slices are generally darker in intensity and do not have enough contrast. Contrast limited adaptive histogram equalization (CLAHE) that operates on small image regions, is used to enhance the contrast throughout the images [27].

3.2 Feature generation: Laws Texture Energy Measures

Laws texture feature measures were introduced, in the early 80s, as a fast texture analysis technique [8]. Laws proposed 3x3, 5x5, and 7x7 convolution masks for texture discrimination. The mask size of 5x5 is used in this work. The Laws masks, 2D kernels, are constructed out of a set of one-dimensional kernels of five elements. These kernels, based on primitive geometric features found in images, are given below.

$$\begin{aligned}
 \text{L5 (Level)} &= [1 \ 4 \ 6 \ 4 \ 1] \\
 \text{E5 (Edge)} &= [-1 \ -2 \ 0 \ 2 \ 1] \\
 \text{S5 (Spot)} &= [-1 \ 0 \ 2 \ 0 \ -1] \\
 \text{R5(Ripple)} &= [1 \ -4 \ 6 \ -4 \ 1]
 \end{aligned} \tag{1}$$

Using these are four kernels, 16 masks of 5x5 sizes are formed: L5L5, L5E5, L5S5, L5R5, E5L5, E5E5, E5S5, E5R5, S5L5, S5E5, S5S5, S5R5, R5L5, R5E5, R5S5 and R5R5. For example, L5S5 is formed as follows.

$$\text{L5S5} = \begin{bmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{bmatrix} \times [-1 \ 0 \ 2 \ 0 \ -1]$$

$$= \begin{bmatrix} -1 & 0 & 2 & 0 & -1 \\ -4 & 0 & 8 & 0 & -4 \\ -6 & 0 & 12 & 0 & -6 \\ -4 & 0 & 8 & 0 & -4 \\ -1 & 0 & 2 & 0 & -1 \end{bmatrix} \quad (2)$$

Based on the 16 generated Laws masks, nine masks are derived, where some of the similar masks are averaged to obtain a single mask. They are: L5E5/E5L5, L5S5/S5L5, L5R5/R5L5, E5E5, E5S5/S5E5, E5R5/R5E5, S5S5, S5R5/R5S5, and R5R5, respectively [28].

The histogram equalized images are then convolved with the 9 masks as defined earlier. For each mask, the convolved image is split into square tiles of size n . The value for n is varied as described in the RESULTS section. For an image tile, the average energy is estimated using the following formula.

$$\text{Average Energy} = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n |I(i,j)| \quad (3)$$

where I is the tile image. In his original work, Laws proposed a pixel based energy calculation, where for each pixel, a square region around it was used to estimate the energy corresponding to the pixel [8]. Then a pixel based image segmentation was performed. In contrast, a region based (i.e. over a squared tile) energy estimation is used here, reducing the resolution of segmentation. This window based method help speed up the processing.

3.3 Training Data Clustering: k -Means Algorithm

Being a popular data clustering method, the k -means algorithm partitions a collection of n data vectors \mathbf{x}_j , with $j = 1, \dots, n$, into k clusters G_i , with $i = 1, \dots, k$ [9]. The algorithm determines a cluster center \mathbf{c}_i for each group G_i by minimizing the inter-cluster sum of squares J , defined in terms of the Euclidean distance, as given in Equation (4) below.

$$J = \sum_{i=1}^k J_i = \sum_{i=1}^k \left(\sum_{j, x_j \in G_i} \|x_j - c_i\|^2 \right) \quad (4)$$

Where, J_i is the cost function within cluster G_i . Instead of the Euclidean distance, any other general distance measure such as Hamming and cosine can also be used between x_j and c_i [9].

The partitioned clusters are usually denoted by a $k \times n$ membership matrix U , whose elements u_{ij} are 1 if the cluster G_i contains the j^{th} data vector and 0 otherwise. It must be noted that data clustering by the k -means algorithm, which is an unsupervised procedure. In this work, training data is clustered by the k -means algorithm.

3.4 Data Classification: k-Nearest Neighbor Algorithm

k -Nearest Neighbor (k -NN) search is a data classification algorithm. This assumes no prior knowledge about the distribution of data [29]. It comes under the category of lazy learning algorithms, hence one of the slow classification schemes. For a query (i.e. data) that must be classified, and given a set of labelled data points with class information, the k -NN algorithm searches for the k nearest data points, for example in an Euclidean sense. Other distance measures such as Chebyshev and Minkowski are also used. Once the k nearest neighbors are known, the algorithm determines the predominant class among the neighbors and assigns the same class to the query data.

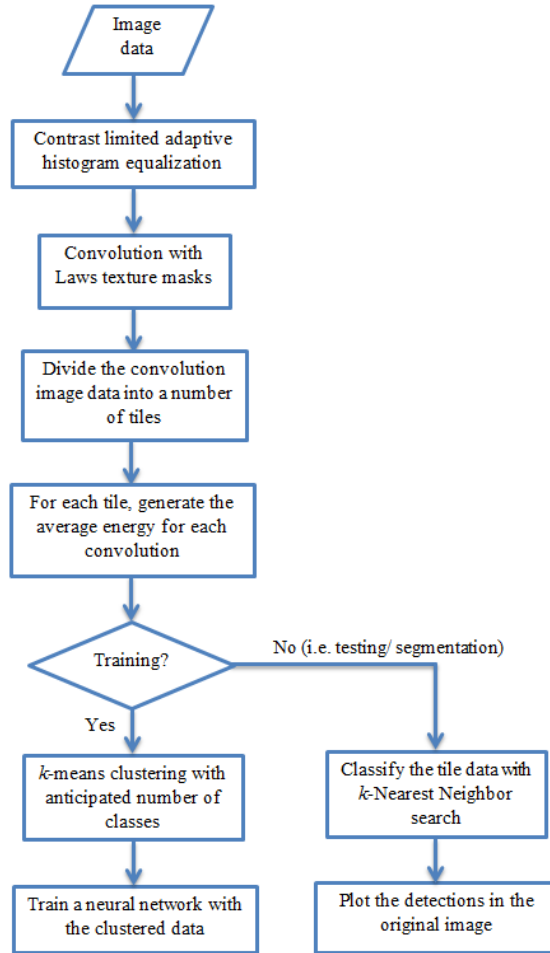


Fig. 2 Algorithm

4 Implementation

The images used in this work are obtained from Fugro Roadware, Canada. Fugro Roadware used its Pave2D system to obtain the images [30]. The Pave2D system consists of two monochrome cameras synchronized with two strobe lights and the setup is an add-on to Fugro Roadware’s ARAN mobile pavement survey system. The cameras have synchronized high-speed shutters, ensuring sharp images without blur at high vehicle speeds. The strobe lights provide artificial lighting much higher in intensity than natural sunlight. The strobe lights that are optimally angled to take care of the curvature of roads across the width, ensure a good level of evenness in

intensity in the images obtained. The images obtained by the cameras are stitched by the Pave2D system in real-time. In total, 500 images, each of which have 2428 x 4916 pixels, are used in the study. An image represents approximately an area of 4.2 x 8.5 m² of pavement surface. Hence, the resolution of the images is roughly 1.7 mm per pixel.

The algorithm is programmed in MATLAB R2011b. Another equivalent algorithm is also coded in C++ with Visual Studio 2013 using the OpenCV library functions. The algorithm is prototyped in MATLAB, due to its shorter development time. Once the algorithm is finalized in MATLAB, it is also translated into C++. The final algorithm is shown in Figure 2.

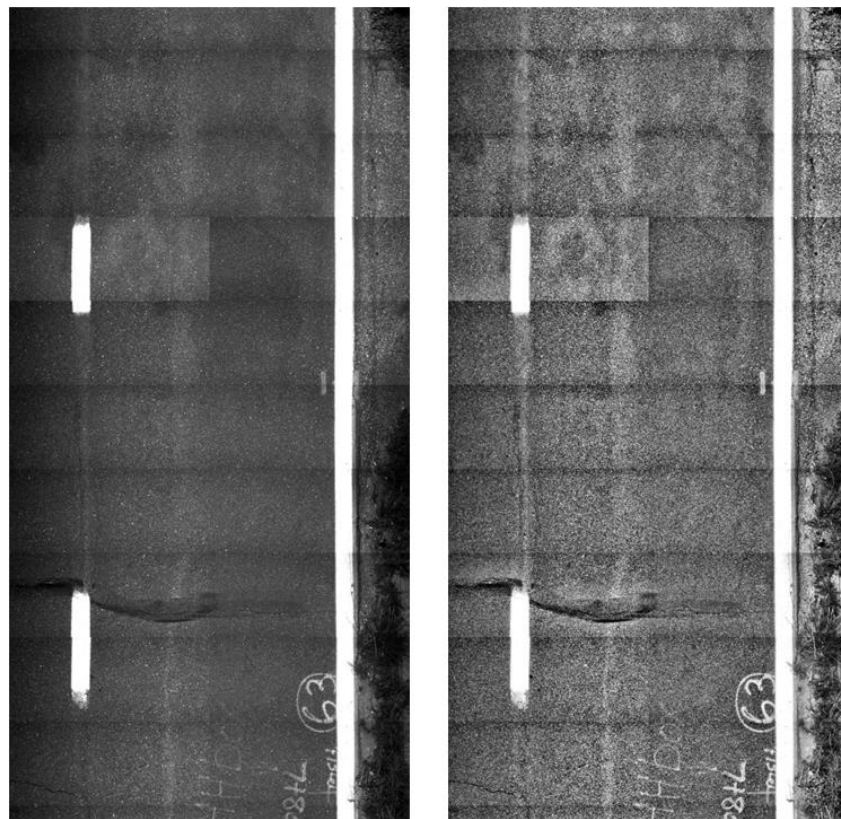


Fig. 3 Original image, left, and after adaptive histogram equalization.

Training is performed with 48 images that are selected to represent the different surface types encountered in the set of 500 images. Initially the images are histogram equalized with the *adaphisteq* function of MATLAB. In OpenCV, a *CLAHE* object is created and used for the

purpose. The training images are then convolved with the 9 Laws texture energy masks. Although in some works local smoothing is applied to the image before it is convolved with the 9 Laws masks [31], for road images used in this work it is found that the discriminating power of the masks are lost by smoothing. In OpenCV, custom masks are designed for the Laws filter and the library function *filter2D* is used. The convolved images are then split into tiles and the average energy of the tiles are calculated. Hence, for each image tile in the original image, there will be 9 average energies from each of the Laws masks. This results in a 9 dimensional feature space.

The data of the 48 training images are clustered using the *k*-means algorithm. The function *kmeans* of MATLAB is used for this purpose; an OpenCV function of the same name executes the C++ code. The *k*-means function is also passed with the anticipated number of classes in the data. The class data of the normal paved surface will be predominant when compared with other classes such as lane markings. Hence, there could be more chances of the existence of many near-identical feature instances and this will slow down the *k*-NN process unnecessarily. To avoid the overhead, the size of the clustered class containing the least amount of samples is chosen as the size for all other classes as well; a random selection is used to select the fewer class samples.

The segmentation phase consists of generating the 9 features for the tiles of each image, and then comparing the values with the *k*-means clustering data, by the *k*-nearest neighbor search and finding out which cluster points are close to it. Function *knnsearch* and object *KNearest* are employed in MATLAB and OpenCV, respectively, to search for closest neighbors. In this work, 5 nearest neighbors of a given data point are considered to determine its class. The code also

plots the class segmentation of the incoming images in different colors. Figure 2 shows the flowchart of the processes involved.

5 Results

A pavement image and its adaptive histogram equalized images are shown in Figure 3. The image is split into 32 x 16 tiles, in the vertical and horizontal directions respectively for histogram equalization. Figure 3 illustrates the higher consistency achieved after histogram equalization. However, the stripes that are in the original images, resulting from the stitching of single exposure camera acquisitions, are still visible in the adaptive histogram equalized image on the right in Figure 3.

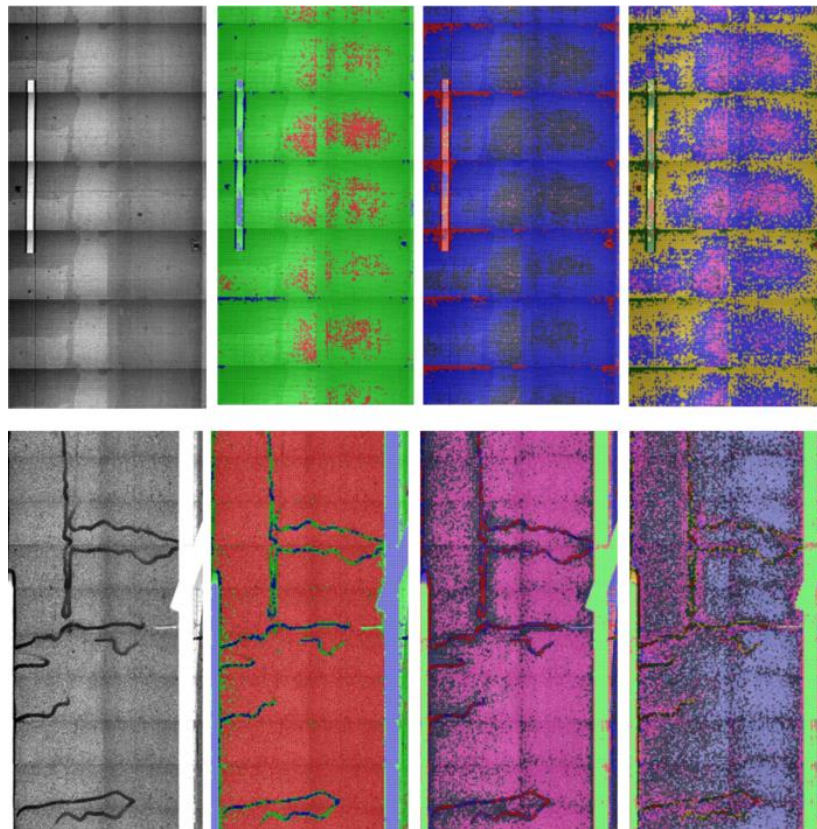


Fig. 4 Two original images and class segmentations with 3, 5 and 8 class number k -means clustering arranged along columns (window size: 20 x 20 pixels)

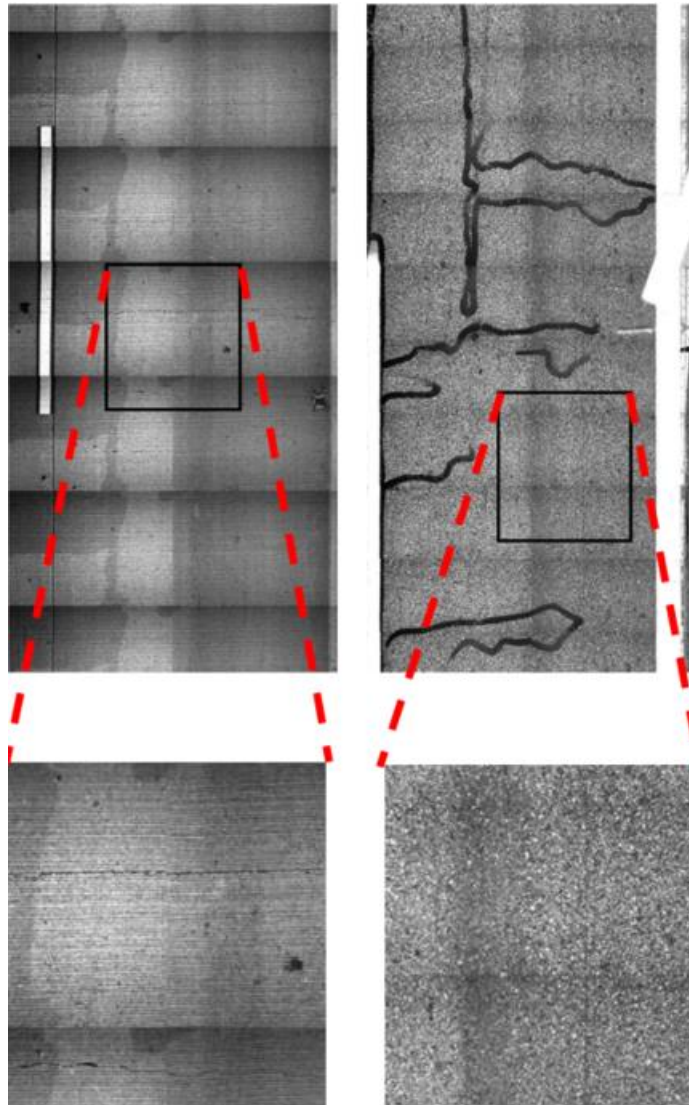


Fig. 5 Two tested images and their underlying pavement surfaces

The images processed by the CLAHE algorithm are convolved with the 9 Laws masks. Different tile sizes, n , are experimented with for feature generation. Window sizes of less than 30×30 pixels are found to be effective, mainly due to the presence of thin lines, such as crack sealings, as shown in Figure 1. A manual inspection of 500 images from Fugro showed that there could be around 8 distinct classes of surfaces. Different number of classes are tested for the k -means-clustering of the feature data of the 48 training images.

Figure 4 shows the segmentation achieved for two tested images with the class sizes of 3, 5, and 8 together with the original image. The classes are given different colors and the classified tiles, out of k -NN search, are plotted with a border of 5 pixels in the respective class color. According to Figure 4, the 2nd column of images, which result from the 3-classes k -means clustering, the class of surfaces shown in green color refer to a regular road surface, in the figure above, and to parts of crack sealing in the image directly below. This creates confusion. The 8-class, k -means clustering allocates too many classes to a single pavement surface, as shown in the last column of detections in Figure 4. Hence, the 5-class clustering shown in the third column of images in Figure 4 is chosen for classification, even though it sometimes allocates two different classes to what appears like the same surface. The dual allocation could be mainly due to the texture variations, which may be emanating from the higher wear and tear on the wheel tracks.

The two images classified have different underlying pavement surfaces as shown in the exploded view in Figure 5. The image on the left has a concrete type pavement surface, whereas the image on the right is of asphalt. The surface on the left in Figure 5 is the dual classes denoted by blue, predominantly, and dark gray colors in the top detection of the third column of Figure 4. The class shown in the red color border is at the edge between two single exposure images that are stitched together. The asphalt surface, shown on the right in Figure 5, is detected and shown in pink and dark gray colors, hence two classes, as given in the bottom picture of the third column of Figure 4. There has to be a method to distinguish the mixed two classes, for both type of surfaces, and detect the underlying surfaces as a homogenous, final detection. This makes post-processing a necessity.

5.1 Post Processing

The class denoted by dark gray color is found among both type of road surfaces, i.e. pink – asphalt or blue - concrete, shown in the exploded views in Figure 5. This misdetection is eliminated by considering smaller regions of the image. Every image on which class segmentation is performed is divided into rectangular areas by an 8 x 2 grid (i.e. vertical x horizontal). For every rectangular area a comparison is performed to determine if the detections are predominantly asphalt (pink) or concrete (blue). Then, the dark gray colored class windows are replaced with the larger of the two classes.

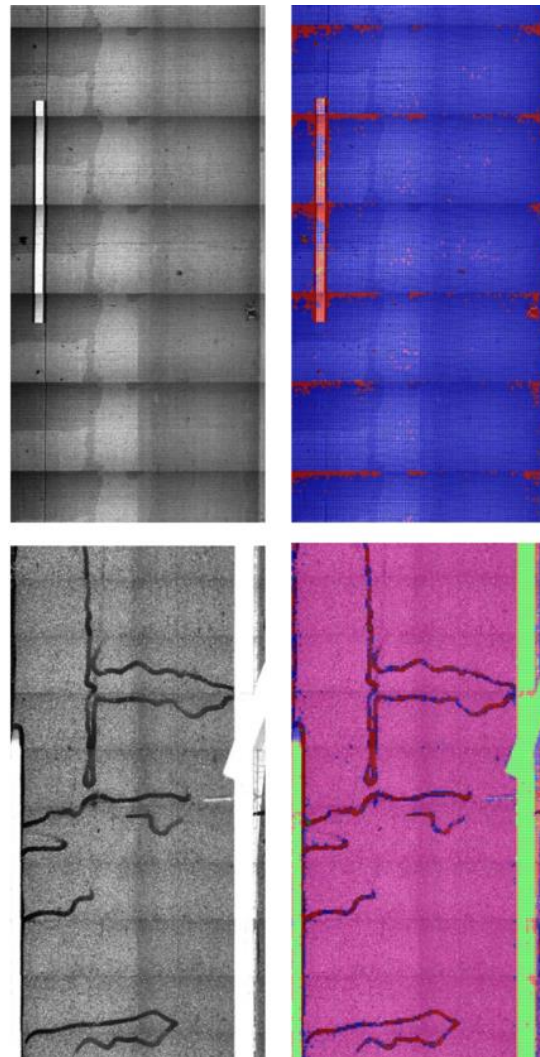


Fig. 6 Two tested images and their final segmentations.

The final detections for the pavement images shown in Figure 4 are given in Figure 6. In Figure 6, the white lines belong to the class plotted in green. The crack fillings (the irregular, wavy, black lines in the middle of the saved surface) and darker surfaces are generally detected as red, with lighter shades given in blue color (that of concrete pavement). This covers all the classes of the detection algorithm. Figure 7 shows two worst detections with their original images. In the first detection, due to the nature of the post-processing operation, one window in the 8 x 2 grid is detected to have the class concrete, predominantly, even though the road surface is asphaltic. A similar, but worst case scenario in the bottom images in Figure 7, where most of the road area is detected as concrete due to post-processing, whereas the original detections have predominantly pink colored tiles (i.e. the asphalt class), in the middle of the image mixed with dark gray tiles. However, these worst scenario detections, among the total images are found to be few.

6 Discussion

Although k -means clustering is performed with 5 classes, due to the ambiguous nature of the class represented in dark gray color, only 4 classes remain after the post-processing. Table I provides the precision and accuracy achieved for randomly selected 31 images from the set of the 500 images provided by Fugro Roadware; the 48 images used for training are excluded from the selection.

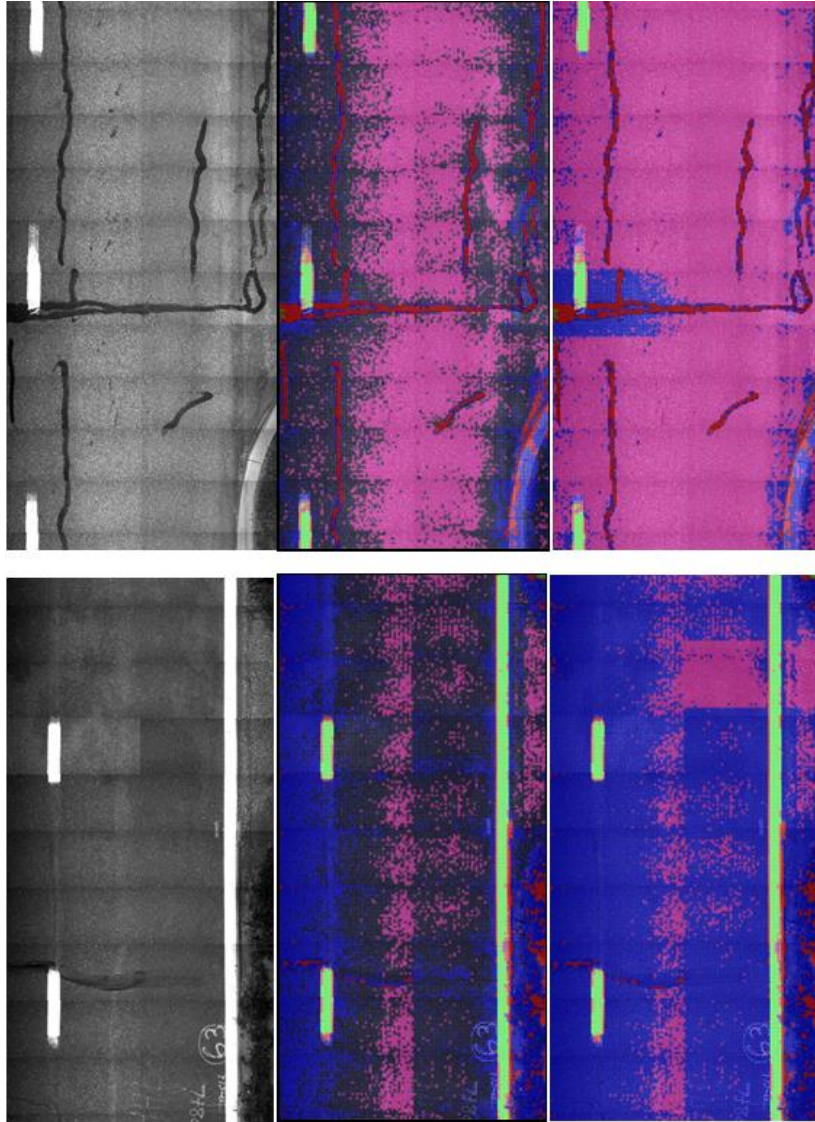


Fig. 7 Two images, their initial and post-processed detections showing the worst-case detections by the scheme

Table 1 Precision and recall values for surface classes.

Class	Percentage values ^a	
	<i>Precision</i>	<i>Recall</i>
Asphalt (pink)	99	96
Concrete (blue)	83	95
White lines (green)	100	82
Crack sealing (red)	94	86

a. Based on manual tile counting to the accuracy of $\pm 1\%$

The precision and recall values are calculated by counting the number of detected tiles, manually, and by using the following formulae.

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}}$$

$$\text{Recall} = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}}$$

When compared to the precision values for the class asphalt, concrete's precision values are lower due to two reasons. The first is because there are many more images of asphaltic surface than concrete. Secondly, there are a few asphaltic surfaces that result in erroneous detections as shown in Figure 7, giving rise to the number of false positives, which occupy a larger proportion owing to the comparatively smaller true positives for the class concrete. There have been no false positives for the class of white lines, leading to a precision of 100%. Many of the white line segments (e.g. the one in the south-east corner of the top images in Figure 7) are not detected correctly, owing to the fact that they are worn out and expose the underlying material structure, confusing the detection. This fact is reflected in more false negatives, and consequently in 82% recall. Some of the crack sealing is also not prominent due to wear, resulting in a lower recall of 86% when compared to the predominant classes.

5.1 Processing Speed

Even though the detection rates are very good, fast processing is of utmost importance for pavement images as explained earlier. Here, the processing overheads in both MATLAB and OpenCV/C++ are provided for comparison. OpenCV is usually faster and thus sets an upper limit on what can be achieved on a PC, in the absence of any parallel programming options. However, MATLAB processing times are also given as the research community extensively uses this

platform. The feature generation for the image size of 2428 x 4916 pixels takes approximately 10.6 seconds, when executed from within MATLAB on a PC with an Intel Core-i7 processor at 2.10 GHz with RAM of 8 GB. The same operation consumes about 7.2 seconds in OpenCV when executed on the same PC. This clearly shows the advantage of OpenCV.

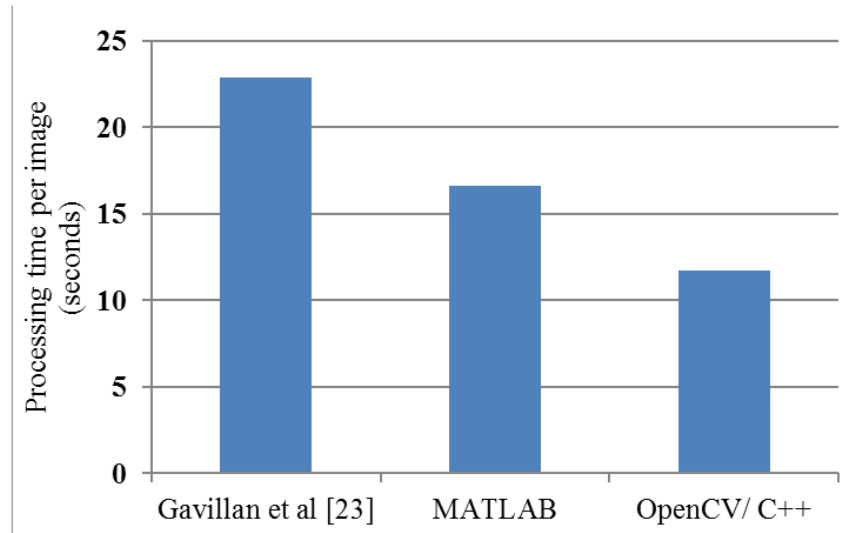


Fig. 8 Performance comparison of with the work of Gavilan et al. [26]

The time of taken for the execution of *k*-means clustering does not have to be considered as it is an off-line process only performed in the training phase. The *k*-NN algorithm on the generated training data for an image that is tested consumes 6.0 seconds in MATLAB. OpenCV takes around 4.5 seconds for the same. Hence, the overall classification time for an image is around 16.6 seconds in MATLAB. Whereas, an image is segmented in 11.7 seconds with OpenCV.

For a 255 x 255 pixel image, a classification time of 125 ms has been achieved [26]. Even though the hardware details of the processing platform has not been provided in the work of Gavilan et al.[26], linearly extrapolating their processing time to the current image size of 2428 x 4916 pixels, results in an equivalent time of 22.9 seconds. Figure 8 shows a comparison of the performance of the algorithm reported here with that of Gavilan et al.[26], resulting in 27.5% and 48.9% reductions in processing time, for MATLAB and OpenCV respectively.

5.1 Time Taken for Feature Generation: A Realistic Comparison

In the absence of the specifics of the processing platform used by Gavilan et al. [26], the comparison of processing times provided in Figure 8 does not make a fair assessment on the merits or demerits of the Laws filters. For a more realistic evaluation, GLCM features, as generated by Gavilan et al.[26], are generated to compare the improvement in feature generation times with Laws texture features. In addition to GLCM, Gavilan et al.[26] also used features such as local binary patterns with some Fourier transform parameters. However, the times to generate additional features are disregarded and only the core features, i.e. GLCM, are considered here for comparison. Hence, this evaluation here provides a conservative estimate of the advantages of the Laws method. MATLAB is solely used for this activity.

Table 2 Time taken to generate features.

Feature generation method		
	Processing time (s)	Comparison to Laws features
Laws	10.6	1
GLCM, 3-bit	684.0	64.5
GLCM, 4-bit	686.8	64.8
GLCM, 5-bit	765.6	72.2
GLCM, 6-bit	1112.4	104.9

GLCM requires the resampling of the intensity level of the image for which the features have to be generated. The resampling is performed to cut down on the processing time, as processing an 8-bit image (the industry norm) will prolong the processing time. The disadvantage with the resampling is that it reduces the sharpness of the image and texture identification becomes difficult if the resampling is very coarse (e.g. 2 or 4 bits or intensity levels). This crucial detail of

intensity resampling is not reported in the work of Gavilan et al.[26]. Hence, a number of reduced bit levels are evaluated here for feature generation. Four angular directions and four pixel distances have been used for feature generation, resulting in the final feature measures of homogeneity, entropy, contrast and energy [26]. For the images used in this work, a MATLAB program is made to estimate the features using MATLAB's in-built function *graycomatrix*. The processing times needed to perform the calculations for different bit levels and the reduction achieved by the Laws method adopted in this paper are given below in Table II (data is for an image size 2428 x 4916 pixels).

According to Table II, a time of 684.0 seconds achieved for the least possible bit level (best case scenario) translates to a time of 3700 ms, instead of the 125 ms presented in Gavilan et al.[26] for an image size of 255 x 255 pixels. This type of time reductions are only possible in embedded hardware platforms. This fact leads to the belief that the processing of Gavilan et al. [26] was performed on a dedicated hardware platform (e.g. an FPGA), and not on a regular PC. This assumption is highly likely as the paper is about a real-world road inspection system armed with line scan cameras, lasers and so on. Nevertheless, Table II illustrates the power of Laws texture energy masks over the commonly employed feature generation method of GLCM in terms of processing times.

7 Conclusions and Future Work

This work has demonstrated the use of Laws texture energy measures in conjunction with k -means clustering and k -nearest neighbor search for the use of pavement classification in a minimally supervised manner. The human intervention is mainly in terms of selecting the number of appropriate classes of surfaces found in the set of images. On top of the precision and recall values in high 80-100%, the speed of processing has been increased by ~ 100% on a PC

when compared to the state-of-the art that probably uses dedicated processors [26] (again this paper provides evidence for this reasoning).

The next stage of the project aims to implement the proposed algorithm on a PC with GPU (graphics processing unit) processing enabled. For k -NN search, a GPU implementation has been shown to run 400 times faster than the equivalent MATLAB implementation [32]. As about 35% of the overall processing time is consumed by the k -NN search, this implementation will at least bring the overall time of 16.6 seconds per image down by 30%. As k -NN is known to be a slow algorithm, it makes much sense to use the parallel computing architecture of a GPU. The feature generation can also be expected to become faster by the highly parallel processing of a GPU further reducing the time consumed. In this regard, it is planned to use the openCL functionalities available with the newest OpenCV versions. openCL (Open Computing Language) is an open source framework for writing programs on PCs making use of both their CPUs and GPUs, enabling highly parallel computation possible. By tapping into OpenCV's GPU extensions, the final aim is to bring the processing down to a few seconds on a PC, making it suitable for processing the thousands of road images collected every day.

References

1. Central Intelligence Agency, The World Factbook. [Online].
<https://www.cia.gov/Library/publications/the-world-factbook/fields/2085.html>
2. Trading Economics. [Online].
<http://www.tradingeconomics.com/world/roads-paved-percent-of-total-roads-wb-data.html>
3. The Asphalt Paving Industry: A Global Perspective. [Online].
<http://www.eapa.org/userfiles/2/Publications/GL101-2nd-Edition.pdf>

4. M. Y. Shahin, "Pavement Condition Survey and Rating Procedure" in *Pavement Manage. for Airports, Roads, and Parking Lots*, 2nd ed. New York: Springer Sci., 2005.
5. Distress Identification Manual for the Long-Term Pavement Performance Program, US Dept. of Transportation, Federal Highways Admin., 2003.
6. S. Rao, H. Mobahi, A. Yang, S. Sastry and Y. Ma, "Natural Image Segmentation with Adaptive Texture and Boundary Encoding", In Proceedings of the Asian Conference on Computer Vision (ACCV), H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.), Part I, LNCS 5994, pp. 135-146, Springer, 2009.
7. P. J. Carlson, E. S. Park, and R. R. J. Porter, "Safety of wider edge lines in The USA." Road safety on four continents: 15th international conference, Abu Dhabi, United Arab Emirates, 28-30 March 2010. Paper. 2010.
8. K. I. Laws, *Textured Image Segmentation*, Ph.D. Dissertation, University of Southern California, January 1980.
9. Jang, J.-S.R., C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing - a Computational Approach to Learning and Machine Intelligence*. First ed. New Jersey: Prentice Hall, ISBN: 0-13-679261066-3, 1997.
10. M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis and machine vision*. California: ITP, 1999.
11. N. R. Pal, and S. K. Pal, "A review on image segmentation techniques", *Pattern Recognition*, vol. 26, no. 9, pp. 1277-1294, 1993.
12. K. S. Deshmukh, "Color image segmentation: A review", *Proceedings of SPIE - The International Society for Optical Engineering*, no. 7546, art. no. 754624, 2010.
13. S. Gould, T. Gao, and D. Koller, "Region-based segmentation and object detection," in *Advances in Neural Information Processing Systems*, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, Eds., December 2009, pp. 655–663.

14. A. Materka, and M. Strzelecki, "Texture Analysis Methods—A Review", Technical Report, University of Lodz, Cost B11 Report, 1998.
15. G. Scarpa, R. Gaetano, M. Haindl, and J. Zerubia, "Hierarchical multiple Markov chain model for unsupervised texture segmentation," *IEEE Trans. Image Process.*, vol. 18, pp. 1830–1843, 2009.
16. S. Chambon and J. Moliard, "Automatic road pavement assessment with image processing: Review and comparison," *Int. J. Geophys.*, vol. 2011, pp. 989354-1–989354-20, 2011.
17. S. Amarasiri, M. Gunaratne, S. Sarkar, and A. Nazef, "Optical texture-based tools for monitoring pavement surface wear and cracks using digital images", *Transportation Research Record (2153)*, pp. 130–140, 2010.
18. Y. Hao, W. Qiu-sheng, and Y. Hai-wen, "An improved image segmentation algorithm and measurement methods for asphalt mixtures," in *Cybernetics and Intelligent Systems (CIS), 2011 IEEE 5th International Conference on*, Sept. 2011 pp.36-41.
19. Y.-B. Yoon, and O. Se-Young, "Vision-based detection and classification of pavement mark using neural network for autonomous driving system" in *Ubiquitous Robots and Ambient Intelligence (URAI), 2011 8th International Conference on*. IEEE, 2011.
20. S. Mathavan, M. Rahman, M. Stonecliffe-Jones and K. Kamal, "Pavement raveling detection and measurement from synchronized intensity and range images", in *Transportation Research Board, Annual General Meeting*, Washington D.C., Jan. 2014.
21. Z.-G.Xu, X.-M. Zhao, L. Yang, N. Wei, and L.-C. Zhang, "Quick and precise road marking segmentation algorithm based on Beamlet", *Chang'an Daxue Xuebao (Ziran Kexue Ban)/Journal of Chang'an University (Natural Science Edition)*, vol. 33, no. 5, pp. 101-108, 2013.
22. A. El Gendy, and A. Shalaby, "Pavement texture classification using two dimensional fourier transform of image-based recovered surface" in *Proceedings, Annual Conference - Canadian Society for Civil Engineering*, 2007, pp. 268-276.
23. Y.-S. Su, S.-C. Kang, J.-R. Chang, and S.-H. Hsieh, "Dual-light inspection method for automatic pavement surveys", *J. of Computing in Civil Engineering*, vol. 27, no. 5, pp. 534-543, 2013.

24. S. Paquis, V. Legeay, H. Konik, and J. Charrier, "Multiresolution texture analysis applied to road surface inspection", in *SPIE Conference on machine vision application in industrial inspection*, San Jose, USA, 1999, pp. 242-249.
25. S. Mathavan, M. Rahman, and K. Kamal, "Application of Texture Analysis and the Kohonen Map on pavement images for Region Segmentation in Crack Detection," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2304, pp. 150-157, 2013.
26. M. Gavilán, D. Balcones, O. Marcos, D. F. Llorca, M. A. Sotelo, I. Parra, M. Ocaña, P. Aliseda, P. Yarza, and A. Amírola, "Adaptive road crack detection system by pavement classification", *Sensors*, vol. 11, no. 10, pp. 9628-9657, 2011.
27. K. Zuiderveld, "Contrast Limited Adaptive Histogram Equalization." *Graphic Gems IV*. San Diego: Academic Press Professional, 1994. pp. 474–485.
28. L. Shapiro, and G. Stockman, *Computer Vision*, pp Prentice-Hall, New Jersey, USA, 2001.
29. L. E. Peterson, K-nearest neighbor, *Scholarpedia*, 4(2), p. 1883.
30. Pave2D 2014.
Available: http://www.roadware.com/products/survey_equipment/aran_subsystems/distress/
31. H. Habib, M. Yousaf, and M. Mohibullah, "Modified laws energy descriptor for inspection of ceramic tiles", in *National Conference on Emerging Technologies*, pages 137–140, 2004.
32. V. Garcia, E. Debreuve, and M. Barlaud, "Fast k nearest neighbor search using gpu", in *Computer Vision and Pattern Recognition Workshops, CVPRW'08*, 2008, pp. 1–6.

Senthan Mathavan is an engineer and researcher working in the areas of machine vision and mechatronics. He received the BSc degree in Mechanical Engineering from the University of Peradeniya, Sri Lanka, in 2005. He then obtained the PhD degree in Mechatronics from Loughborough University in the UK, in 2009. He held a Rolls-Royce funded Post-Doctoral Research position at Loughborough University from 2010 to 2011. From 2011 to 2016, he had been a Mechatronics Engineer with ASML in the Netherlands. Currently he is a machine vision architect at Nobleo – Technology, The Netherlands. He

is also a Visiting Fellow at Nottingham Trent University, UK since 2012. He has carried out research and development in the aerospace, semiconductor and transportation industries.

Akash Kumar is a 4th year undergraduate student in Mechatronics Engineering at the College of Electrical and Mechanical Engineering (CEME), National University of Sciences and Technology (NUST), Islamabad, Pakistan. Currently, he is enrolled in a non-degree program at the Mississippi State University for one semester. His interests include planetary robotics, medical imaging and sensor systems.

Khurram Kamal is serving as an Assistant Professor at the Department of Mechatronics Engineering, National University of Sciences and Technology, Islamabad, Pakistan. Prior to his PhD research, at Loughborough University, UK, he has also worked as Assistant Manager of Production Planning and Control at the Press Shop department of Pakistan, Suzuki Motor Company Ltd. from 2003 to 2004. His areas of interest are Artificial Intelligence, Image Processing, Embedded Systems, Intelligent Machines, Condition Based Maintenance, Unmanned Aerial Vehicles, and Sustainability.

Michael Nieminen has over 12 years of experience transferring discontinuous innovations from research and development into production. At Fugro Roadware, he led a cross-functional team of operations and R&D staff to re-engineer the data collection and analysis platforms for the enterprise. This resulted in the release of next generation van-based sensor technology (ARAN) as well as the Vision and iVision processing software, which included new vision systems for the measurement of road defects. Michael enjoys collaborating with excellent technical minds on pressing operational challenges in the real world. Michael has a BAsC degree in mechanical engineering from the University of Waterloo, and is a licensed Professional Engineer in the Province of Ontario, Canada.

Hitesh Shah received B. Engg. degree in Computer Science from Sardar Patel University, Vallabh Vidhyanagar, India, in 2003. He then obtained M. Tech degree in Communication Engg. From IIT Bombay, India, in 2006. Mr. Shah has over 8 years of experience and a proven track record of developing and implementing computer vision, image processing, and machine learning algorithms to suit target applications in diverse areas like scientific measurement to sports cinematography. He is an inventor on issued patents in field of 3D reconstruction and visual odometry. In his current role as Director of

Technology Development at Fugro Roadware, he continually works on adapting the state of art data acquisition and mining approaches to ensure accurate and precise results for the clients.

Mujib Rahman is presently working as a Senior Lecturer in Civil Engineering at the School of Engineering and Design, Brunel University. He received a BS degree in Civil Engineering from Bangladesh Engineering University and Technology (BUET), Bangladesh, and an MSc in Structural Engineering from City University London, UK, and the PhD degree from University of Nottingham, UK. He has been actively involved in pavement engineering related research, consultancy and teaching for more than twelve years. Dr. Rahman was a recipient of an Institution of Civil Engineers East Midland Merit Award in 2013 for his research on the performance of asphalt patch repair.

Caption List

Fig. 1 Two images from a pavement survey of Fugro Roadware.

Fig. 2 Algorithm

Fig. 3 Original image, left, and after adaptive histogram equalization.

Fig. 4 Two original images and class segmentations with 3, 5 and 8 class number k -means clustering arranged along columns (window size: 20 x 20 pixels)

Fig. 5 Two tested images and their underlying pavement surfaces

Fig. 6 Two tested images and their final segmentations.

Fig. 7 Two images, their initial and post-processed detections showing the worst-case detections by the scheme

Fig. 8 Performance comparison of with the work of Gavilan et al. [26]

Table 1 Precision and recall values for surface classes.

Table 2 Time taken to generate features.