ABSTRACT

This paper describes the detection of rust defects on highway steel bridges, which are one of the most commonly observed defects on coating surfaces and thus have to be taken care of appropriately since they severely affect the structural integrity of bridges. A rust defect assessment method is presented that automatically detects the percentage of rust in a given digital image of bridge surface taken from a conventional digital camera. A training and detection algorithm is implemented to classify a given block of the image as rust or non-rust. The results of the algorithm are analyzed for its efficiency and possible optimization techniques are suggested.

Index Terms— Rust detection, bridges, digital image processing, discrete wavelet transform, classification.

1. INTRODUCTION

Rust defect assessment is important in order to maintain a good quality of steel bridge painting. Bridge managers can more realistically develop long-term cost-effective maintenance programs if they have dependable coating condition data. Also, they can make decisions as to whether a bridge shall be painted again immediately or later. Taking digital images of the steel bridge surface with a conventional camera to evaluate its painting surfaces offers the advantages of being inexpensive, accurate, objective, fast, and consistent. Figure 2a shows a typical color image with a region of rust.

Much research has gone into recognition of pavement cracks and the classification of crack types applying projection methods [1, 2]. Based on a similar method, a novel approach to recognize the existence of bridge coating rust defects by processing digital color images for statistical data acquisition and multivariate statistical analysis was proposed by Lee, Chang et al. in [3]. In another paper, a comparison of the eigenvalues of defective and non-defective images was proposed to determine the presence of rust in it [4]. However, this method does not provide any information about the percentage of rust content. A second stage would be needed to determine it in the set of images classified as being defective.

2. RUST DETECTION ALGORITHM

The proposed algorithm calculates the percentage of rust rather than just classifying an image as defective and non-defective. It applies the retrospective method of illumination correction using entropy minimization to correct the non-uniform illumination in the images as a pre-processing step, which perfectly eliminates the shading effects [8]. The algorithm is based on the concepts of wavelet transforms, principal component analysis, and pattern classification. A novel non-iterative approach of calculating the rust percentage in an image is proposed and hence convergence is always guaranteed. Since color images are used and rust is distinctive in color, there is no information loss.
used to determine if a given block is rust or non-rust. The feature vectors are extracted after wavelet transformation of the image. Finally the percentage of rust in the given image is calculated.

### 2.1. Feature Vectors

The feature vectors used for classification are entropy and energy values in the wavelet domain [9]. One level of wavelet transform is applied to all three color planes (RGB) of the image. The energy and entropy values in each subband \( B, (B \in HH, LH, HL, LL) \) are calculated as these values in the wavelet domain of the image capture the texture properties of an image and prove to be very useful for designing the classifier [10]. The average luminance of the image is also calculated since it helps to differentiate between the rust and non-rust pixels in images over a huge range of intensity values. The entropy is calculated as:

\[
H(N(x, y)) = -\sum_{n=1}^{G} p(n) \log_2 p(n) \tag{1}
\]

where \( p(n) \) is the probability that a pixel in image \( N(x, y) \) has value \( n \) and \( G \) denotes the number of gray levels [9]. The energy of each subband and color plane is calculated as:

\[
E_B^{X_m} = \frac{1}{N^2} \sum_{i=0}^{N} \sum_{j=0}^{N} (w_{B,i,j}^{X_m})^2 \tag{2}
\]

where \( w_{B,i,j}^{X_m} \) is the wavelet coefficient at the \((i, j)\) location in subband \( B \) and \( X_m \) is the color plane \( (m = 1, 2, 3) \).

### 2.2. Training

The feature vectors are extracted from images which are either pure rust or non-rust. A step by step training procedure consists of the following steps:

**Step 1:** All the sample images are read one after the other into a matrix. The rust and non-rust images are grouped as two separate sets.

**Step 2:** All three color planes of the images are transformed into the wavelet domain using Haar wavelet and one level of decomposition.

**Step 3:** Energy and entropy of each subband are calculated for all three color planes, resulting in a total of 24 features.

**Step 4:** Average luminance of the image is calculated by first converting the image from RGB to NTSC color space, resulting in the 25\(^{th}\) feature.

**Step 5:** Correlation among the 25 features is eliminated using Principal Component Analysis (PCA). With a loss of 0.03% in information, the dimensionality of the feature vector is reduced to 5. The transformation matrix is stored as it will be used in the detection algorithm.

**Step 6:** Since the classes (rust and non-rust) are separable, Least Mean Squares method, which minimizes the mean square error of the distance between the classified feature vectors and the classifier in the reduced feature vector, is used for designing the classifier. This approach has the advantages of being unbiased, normally distributed, and having the minimum variance. In addition, it is a non-iterative process and simple compared to the K-means clustering algorithms [6].

**Step 7:** The values of the classifier and transformation matrix obtained from PCA are stored in memory.

### 2.3. Detection

A classifier was successfully designed from the training algorithm explained in the previous section. Here, the step by step procedure of the implementation of the detection algorithm, which involves calculating the percentage of rust in an image, is explained. A color steel bridge coating image is given as the input to the algorithm which then provides the percentage of rust as output. The detection algorithm involves the following steps:

**Step 1:** The color image taken from a conventional digital camera, of the steel bridge coating surface is read into a matrix.

**Step 2:** Illumination correction using entropy minimization technique [8] is applied to the images, resulting in images with a uniform background illumination such that shading effects do not interfere with the detection results.

**Step 3:** When a sample image containing rust is analyzed for its color properties, it is found that the lack of blue color in the blue plane is very prominent in areas where rust is present. The blue plane of the image is cross-correlated with a template that is purely rust. The resulting matrix shows a low value at potential areas of rust in the image.

**Step 4:** To set an adaptive threshold value, below which pixels are considered to be potential areas of rust, the wavelet transform technique is used. It is applied to the resulting cross-correlated matrix where the HH band captures the edge information. This edge information of rust is nothing but the threshold value. The pixel that gives highest value in the HH band is mapped back to the cross-correlated image and its value at that point is considered as the threshold value for that image. Hence, the threshold value varies from image to image and is generated automatically. All the pixels that show values below a threshold are stored in the form of \( 8 \times 8 \times 3 \) blocks. The pixels which are a part of this block should form a continuous region in the image. Pixels that do not form a continuous region of \( 8 \times 8 \) will be left out.

**Step 5:** The wavelet transform is applied to all three color planes in the \( 8 \times 8 \times 3 \) blocks obtained from the previous step. This is similar to Step 2 of the training algorithm.

**Step 6:** The energy and entropy values for all the subbands of the three color planes are calculated along with the average luminance of the image. This step is again similar to Step 3 and Step 4 of the training algorithm. Each feature vector will have 25 components.
Step 7: Using the transformation matrix obtained during training, the size of the feature vectors is reduced to 5. This operation is performed on all the feature vectors. 

Step 8: Using the classifier from the training algorithm, it can be determined if a given $8 \times 8$ block is rust or non-rust. In a similar way, all the blocks are classified as rust or non-rust blocks of $8 \times 8$ pixels. 

Step 9: After the blocks, which belong to a rust region, have been determined, the overlap that exists between blocks is eliminated. Next, the percentage of rust is calculated.

3. RESULTS AND DISCUSSION

A step by step processing example is shown in Figure 1. For designing the classifier, a set of 90 images is used. Of these, 55 images are complete rust. They are obtained by cropping the rust part of the steel bridge coating images. The remaining 35 images are purely non-rust which again are cropped versions of the steel bridge coating images. These cropped images are from real data. During the detection of rust, when the blue plane of the color image is cross-correlated, shades of red depict higher cross-correlation whereas shades of blue show lower cross-correlation in the image. Lower cross-correlation is an indicator of potential areas of rust content. These areas are clustered into blocks of $8 \times 8$ and are then classified as rust or non-rust blocks. The blocksize was chosen to be $8 \times 8$ as it is the minimum size required for a one level Haar wavelet transform to extract useful feature vectors. Several block sizes were tested and $8 \times 8$ blocks proved to be best suited for this application.

Figure 2 shows perfect detection of the percentage of rust in two different images. However, it can be seen from the Figure 3 that some pixels of rust are not detected or left out another two images. This happens because these rust areas have dimensions less than the $8 \times 8$ block. In such a case, they cannot be analyzed and hence are left out in the processing. Feature extraction in the wavelet domain helps to extract the features of the rust and non-rust classes. The same feature vectors and classification applied in the spacial domain of image fails to detect the rust pixels.

30 images were classified using this algorithm. None of the pixels that were non-rust were detected as rust. As can be seen from Figure 3, all the true pixels of rust were detected when the blocksize was $8 \times 8$. However, there were several instances where rust regions larger than $8 \times 8$ were not detected. Increasing the block size to $8 \times 8$ would help to detect these regions, but it would also increase the processing time. To overcome this problem, we can use a combination of multiple block sizes to detect rust regions of different sizes. 

4. CONCLUSION

Existing well-established techniques in digital image processing were used to propose a novel rust detection algorithm. This approach uses the concepts of pattern recognition and wavelet transform for detection of rust in an image. It does not involve iterations and hence never has convergence issues. To increase the computational efficiency, dimensionality of the feature vectors can be reduced after analyzing the effect of each vector on the classification of rust and non-rust pixels. The process of selecting $8 \times 8$ blocks can further be optimized to reduce the overlap of blocks to be tested and yet not miss any pixels which also help to improve the computational ef-
efficiency. This rust detection algorithm can be extended and adopted to many other fields by varying the parameters suitable to the properties of the set being tested for. For every new application, a new set of training images will have to be generated and fed into the training algorithm. Also, feature vectors extracted for each case might be different depending on the application.

5. REFERENCES


