

Automatic Inspection of Textured Surfaces by Support Vector Machines

Sina Jahanbin^a, Alan C. Bovik^a, Eduardo Pèrez^b, Dinesh Nair^b

^a Department of Electrical and Computer Engineering, the University of Texas at Austin, USA;

^b National Instruments, Austin, USA

ABSTRACT

Automatic inspection of manufactured products with natural looking textures is a challenging task. Products such as tiles, textile, leather, and lumber project image textures that cannot be modeled as periodic or otherwise regular; therefore, a stochastic modeling of local intensity distribution is required. An inspection system to replace human inspectors should be flexible in detecting flaws such as scratches, cracks, and stains occurring in various shapes and sizes that have never been seen before. A computer vision algorithm is proposed in this paper that extracts local statistical features from grey-level texture images decomposed with wavelet frames into subbands of various orientations and scales. The local features extracted are second order statistics derived from grey-level co-occurrence matrices. Subsequently, a support vector machine (SVM) classifier is trained to learn a general description of normal texture from defect-free samples. This algorithm is implemented in LabVIEW and is capable of processing natural texture images in real-time.

Keywords: outlier detection, support vector machines, grey-level co-occurrence matrix, surface inspection

1. INTRODUCTION

Automatic visual inspection (AVI) is an important application of computer vision that plays an important role in quality control of manufactured products. AVI systems attempt to find functional or cosmetic defects by observing visual data from the products to be inspected. Machine inspection systems are inherently more consistent than human inspectors because they are not vulnerable to emotional, physical, and environmental distractions. Currently, a wide variety of industries are saving significant amounts of money and manpower by relying on machine vision solutions to control the quality of products such as printed circuit boards (PCBs), automotive parts, and labels on product packages. For instance, Andres *et al.*¹ developed a semi-automatic visual inspection system for car seat frames. In this system an image of a good quality car seat is picked as a template and a pattern matching step is utilized to detect a number of markings and align the image taken from the unit under inspection with the template. Next, the presence of each component such as springs, bracket pins, welding spots, and blots are determined based on significant intensity differences between template and the aligned image. In fact the majority of practical AVIs are used to inspect fixed templates or patterns in captured images. Such tasks can easily be done by simple template matching and edge detection and gauging steps provided in configurable machine vision development environments such as NI Vision Builder for Automated Inspection.²

Despite significant improvements in texture analysis techniques, inspection of products having natural textured surfaces is largely accomplished by trained human inspectors. Products such as tiles, textile, leather, and lumber are covered with textures that cannot be described by regular textures; therefore, the local distribution of intensities should be modeled stochastically. A machine vision system replacement for trained human inspectors should be flexible enough to recognize scratches, cracks, stains, and other unexpected flaws that appear in varying sizes and forms.

Generally, texture analysis techniques can be divided into two main categories: statistical and signal processing methods. Examples of well known statistical texture analysis tools include Gaussian Markov random field

Send correspondence to Alan C. Bovik: E-mail: bovik@ece.utexas.edu, Telephone: 1 512 471 5370, Fax: 1 512 471 1225, URL: <http://live.ece.utexas.edu/people/bovik/>

(GMRF),³ grey-level co-occurrence matrices (GLCM),⁴ and simultaneous autoregressive models (SAR).⁵ Signal processing methods became an important trend of texture analysis following the development of the relationship between wavelet transforms and multi-resolution theory in the late 1980s by Mallat.⁶ Features extracted over multiple resolutions can better characterize textures than those extracted from a single scale. Bovik *et. al.*⁷⁻⁹ proposed an approach for analyzing visible textures based on responses of Gabor filters with various orientation, center frequency, and radial bandwidths. Traditional wavelet transforms subsample the filtered signal at each scale generating intra-scale and inter-scale dependencies. Lu *et al.*¹⁰ addressed this problem using intra-scale and inter-scale fusion schemes. In order to obtain more complete characteristics from textures while minimizing dependencies between the coefficients of wavelet transforms, Unser¹¹ proposed multiresolution texture characterization using wavelet frames. Wavelet frames are a variation of wavelet transforms in which the output of the filter banks are not subsampled, and as a result each subband image has the same size as the input image. Unlike discrete wavelet transforms (standard critically subsampled), discrete wavelet frames are translation invariant and perform better in texture classification and segmentation.¹¹

In this paper, a texture defect detection algorithm is proposed in which a signal processing approach utilizing a discrete wavelet frame decomposition is combined with a statistical approach, GLCM based statistics, to characterize visual textures. The image captured from a textured surface is initially decomposed into several subbands (different resolutions and orientations) using overcomplete and shift-invariant wavelet frames. This multiresolution analysis facilitates inspection at multiple scales and orientations. The subband images are partitioned into non-overlapping windows and distribution of coefficient in each window is analyzed using a grey-level co-occurrence matrix. Second order statistics, commonly known as Haralick features,⁴ are calculated from the GLCM representations. Haralick features extracted from all subbands are concatenated to represent partitions in high dimensional feature space. Eventually, one-class support vector machines (SVM) are trained to learn a general description of normal texture under inspection in this feature space. This algorithm is implemented in LabVIEW and tested to inspect visual texture images collected from real tile samples in real-time.

This paper is organized as follows. In section 2 mathematical background of wavelet frames are presented. Section 3 explains grey-level co-occurrence matrices and closely related Haralick features. The classifier training and outlier detection using support vector machines is explained in section 4. Finally, results and discussions are organized in section 5.

2. WAVELET FRAME DECOMPOSITION

The first step in this proposed visual texture inspection framework is a multi-resolution analysis of the input textures. The multi-resolution analysis is inspired by psycho-visual findings showing that humans and other mammals perceive images in a multi-scale manner.¹²⁻¹⁴ Many researchers in computer vision discipline have tried to simulate this multi-resolution/multichannel behavior using mathematical analysis tools such as wavelet transforms.^{10, 15} Wavelet coefficients at finer scales are appropriate for micro-texture modeling while the coarser ones can capture macro-texture attributes.

Traditional discrete wavelet transforms (DWT) are utilized in the majority of multi-resolution texture and image analysis applications. One drawback associated with the discrete wavelet transform is that a simple integer shift in the input image will result in non-trivial changes in the wavelet coefficients (translation-variant). Another problem of the DWT is that the filtered signals are downsampled at each iteration resulting in undesirable intra-scale and inter-scale dependencies. In this paper, discrete wavelet frame transforms (DWFT) proposed by Unser¹¹ are employed to obtain translation-invariant characteristics from textures with minimum dependencies between the transform coefficients. Wavelet frames are variations of wavelet transforms in which the output of the filter banks are not subsampled and as a result each subband image has the same size as the input image and perform better in texture classification and segmentation.¹¹

The hierarchical signal analysis/decomposition using discrete wavelet transform (DWT) which is relevant in the context of texture analysis can be implemented by iterative filtering and downsampling operation with analysis lowpass and highpass filters h and g . These filters and their corresponding reconstruction counterparts satisfy the general “perfect reconstruction” constraint $H(z)\tilde{H}(z^{-1}) + G(z)\tilde{G}(z^{-1}) = 1$ in the z-transform domain. At

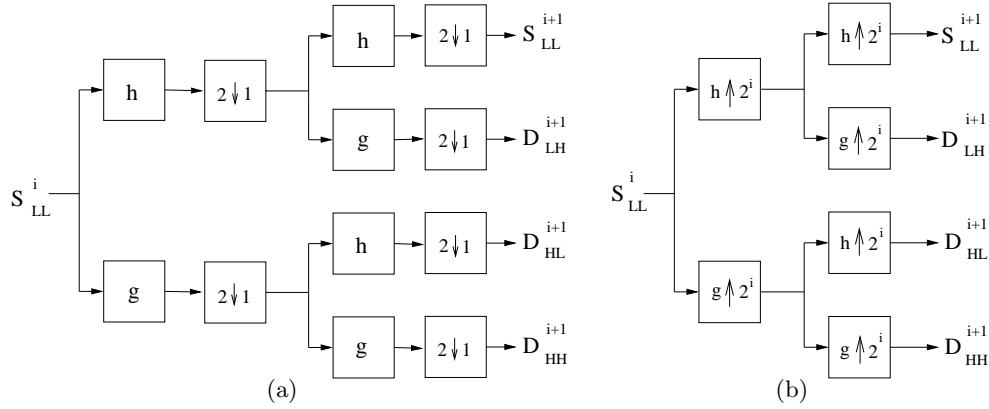


Figure 1. (a) One stage of 2D-DWT where $2 \downarrow 1$ denotes downsampling by factor of 2. (b) One stage of 2D-DWFT where $h \uparrow 2^i$ denotes inserting $2^i - 1$ zeros between taps of filter h .

each iteration the coefficients of the coarser approximation, s_{i+1} , and the detail coefficients, d_{i+1} , are calculated from current coefficients, s_i , by

$$\begin{cases} s_{i+1}(k) = [h * s_i(k)]_{\downarrow 2} \\ d_{i+1}(k) = [g * d_i(k)]_{\downarrow 2} \end{cases} \quad (i = 0, \dots, I) \quad (1)$$

where the $s_0(k) = f(k)$ is the input signal to the filter bank. This concept is extended to 2-D discrete signals (images) where 2-D filters are obtained by the tensor product of 1-D lowpass and highpass filters h and g along the rows and columns. After one stage of decomposition, the image at resolution i is decomposed into four subband images (one coarser approximation S_{LL}^{i+1} and three detail images D_{LH}^{i+1} , D_{HL}^{i+1} , and D_{HH}^{i+1}). These three detail subbands are also known as horizontal (H), vertical (V), and diagonal (D) details.

Unser¹¹ proposed an overcomplete wavelet decomposition (DWFT) and showed that it constitute a tight frame of l_2 and implemented a fast iterative decomposition algorithm for it. This DWFT decomposition is formulated in (2)

$$\begin{cases} s_{i+1}(k) = [h]_{\uparrow 2^i} * s_i(k) \\ d_{i+1}(k) = [g]_{\uparrow 2^i} * s_i(k) \end{cases} \quad (i = 0, \dots, I) \quad (2)$$

where $s_0(k) = f(k)$ is the input signal. Figure 1(b) shows one stage of 2-D DWFT where successive convolution along the rows and columns of the image is taken place with the 1-D filters $[h]_{\uparrow 2^i}$ and $[g]_{\uparrow 2^i}$ which are the basic filters h and g expanded by insertion of an appropriate number of zeros ($2^i - 1$) between taps. Notice that the dyadic subsampling does not exist in implementation of DWFT and as a result all the decomposed subbands are of the same size as the original image.

In this study, each input image undergoes two level of decomposition. The first decomposition step results in an approximations (A) as well as horizontal, vertical and diagonal details (H, V, D). The approximation, A , is decomposed again to bring about subbands AA, AV, AH , and AD . In this algorithm, Biorthogonal wavelets are used for subband decomposition, i.e., the analysis filters h and g are different from synthesis filters \tilde{h} and \tilde{g} . Compared to the orthogonal wavelets, the biorthogonal ones preserve linear phase better, have finite impulse response, and the wavelets can have higher regularity.¹⁶ This wavelet decomposition module is developed using ‘‘Advanced Signal Processing Toolkit’’¹⁷ in LabVIEW. Figure 2 shows an example of image captured from a textured surface along with results of one stage decomposition. The scratch on the texture is remarkably secluded in the subband H .

3. STATISTICAL FEATURE EXTRACTION

The statistical features extracted in this paper are Haralick features that are second-order statistics. Psychological findings by Julesz¹⁸ indicate that the human eye cannot make preattentive discrimination between textures that

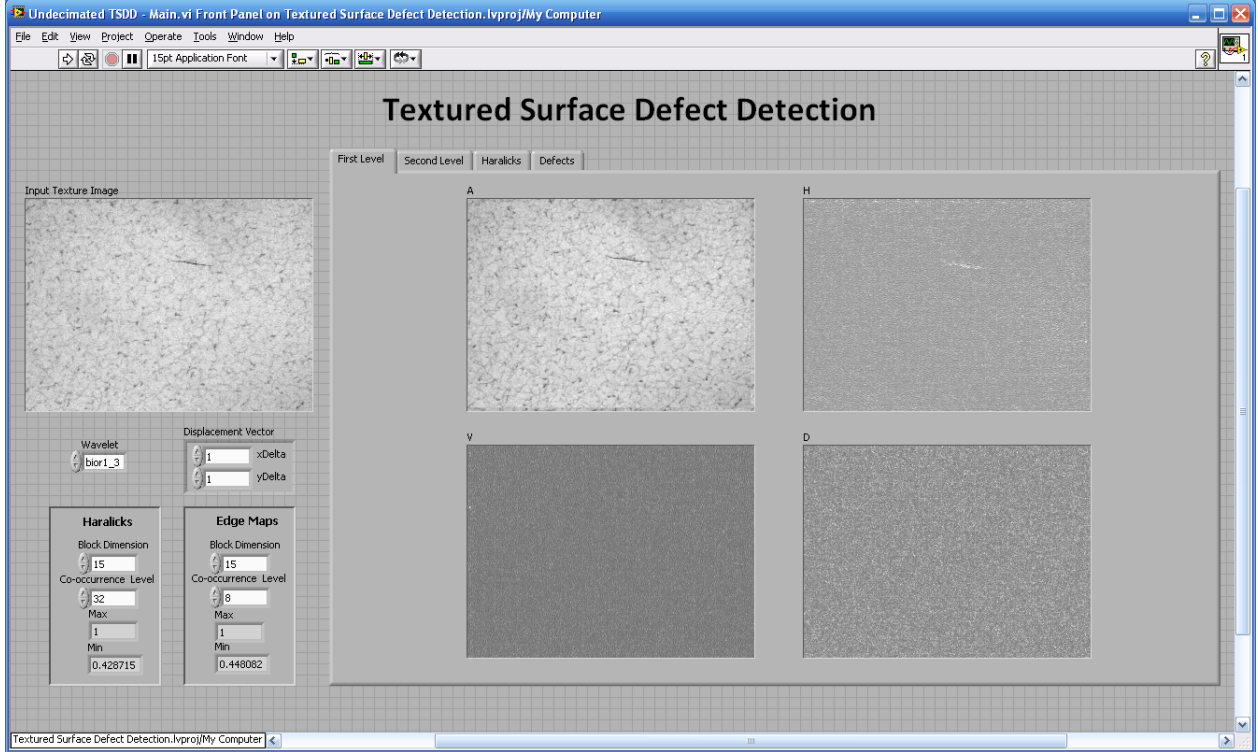


Figure 2. Results of one first decomposition stage of an example texture image with a visible scratch.

have identical first and second order statistics. Therefore, an automatic inspection system that competes with human inspectors should at least take into account the second order statistics. One of the most prevalently used second order texture features are those defined by Haralick et al.⁴ based on the grey level co-occurrence matrix (GLCM) in the 1970s.

3.1 Grey Level Co-occurrence Matrix

The first step in calculation of Haralick features from a texture image $I(x, y)$ is to construct its GLCM. The GLCM basically estimates the joint probability that a pair of pixel values occur at a distance vector \vec{d} from each other. Assume that visual texture $I(x, y)$ is an $N \times M$ matrix consisting of G different grey shades, its GLCM for displacement $\vec{d} = (d_x, d_y)$ is an $G \times G$ matrix:

$$P_{\vec{d}}(i, j) = \sum_{x=1}^N \sum_{y=1}^M \delta \{I(x, y) = i \wedge I(x + d_x, y + d_y) = j\} \quad (3)$$

where $\delta \{True\} = 1$ and $\delta \{False\} = 0$. The number in the element (i, j) of the GLCM matrix, $P_{\vec{d}}(i, j)$, counts the number of times that the pixel with value i occurred \vec{d} away from a pixel with value j . Figure 3 shows an example 4×4 image $I(x, y)$ with 8 gray levels. The corresponding GLCM for displacement $\vec{d} = (0, 1)$ is shown in Figure 3 to be a 8×8 matrix $P(i, j)$ representing how many time a pixel with value i happens to be on the left hand side of a pixel with gray level j . For example, pixels with value 1 are twice located to the left of the pixels with value 2 (circled with red line) and hence $P_{(0,1)}(1, 2) = 2$. Similarly, $P_{(0,1)}(8, 1) = 1$ because only once a pixel with value 8 is located to left of a pixel with value 1 in this example. The co-occurrence matrix can be calculated from the overall visual texture in applications such as texture classification. In the case of texture defect detection, local features are desirable. The co-occurrence matrices can be extracted locally either by sliding a window over the texture and calculating GLCM at each location and associating it with the center pixel of the window, or by partitioning the visual texture into non-overlapping windows and consequently calculating the

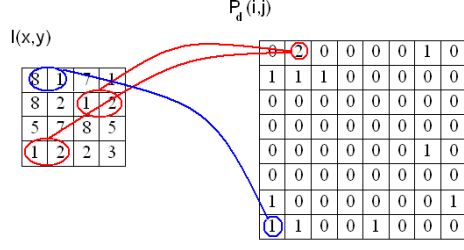


Figure 3. An example image $I(x, y)$ with 8 gray-levels is shown with its GLCM for displacement $\vec{d} = (0, 1)$.

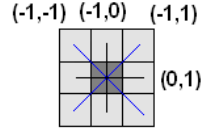


Figure 4. Eight major directions over which the GLCM is calculated.

GLCM for each partition. In this paper, co-occurrence matrices are calculated locally from wavelet decomposed texture images that are partitioned into non-overlapping square windows of size 15×15 .

One important parameter in calculation of GLCM is the displacement vector, \vec{d} , because a constant visual texture results in a different co-occurrence matrix for each displacement vector \vec{d} . In order to achieve direction independent GLCMs, average of co-occurrence matrices for 8 different displacements (Figure 4) are used in the Haralick feature computations.

3.2 Haralick Feature Extraction

The GLCM calculated at each partition of the subband texture is subsequently exploited for extraction of five discriminative second-order statistical attributes representing that partition (Haralick features). These features are *entropy*, *dissimilarity*, *contrast*, *homogeneity*, and *correlation*.

$$Entropy = \sum_{i=1}^G \sum_{j=1}^G P_{i,j} (-\ln P_{i,j}) \quad (4)$$

$$Dissimilarity = \sum_{i=1}^G \sum_{j=1}^G P_{i,j} |i - j|^2 \quad (5)$$

$$Contrast = \sum_{i=1}^G \sum_{j=1}^G P_{i,j} (i - j)^2 \quad (6)$$

$$Homogeneity = \sum_{i=1}^G \sum_{j=1}^G \frac{P_{i,j}}{1 + (i - j)^2} \quad (7)$$

$$Correlation = \sum_{i=1}^G \sum_{j=1}^G P_{i,j} \left[\frac{(i - \mu_i)(j - \mu_j)}{\sqrt{\sigma_i^2 \sigma_j^2}} \right] \quad (8)$$

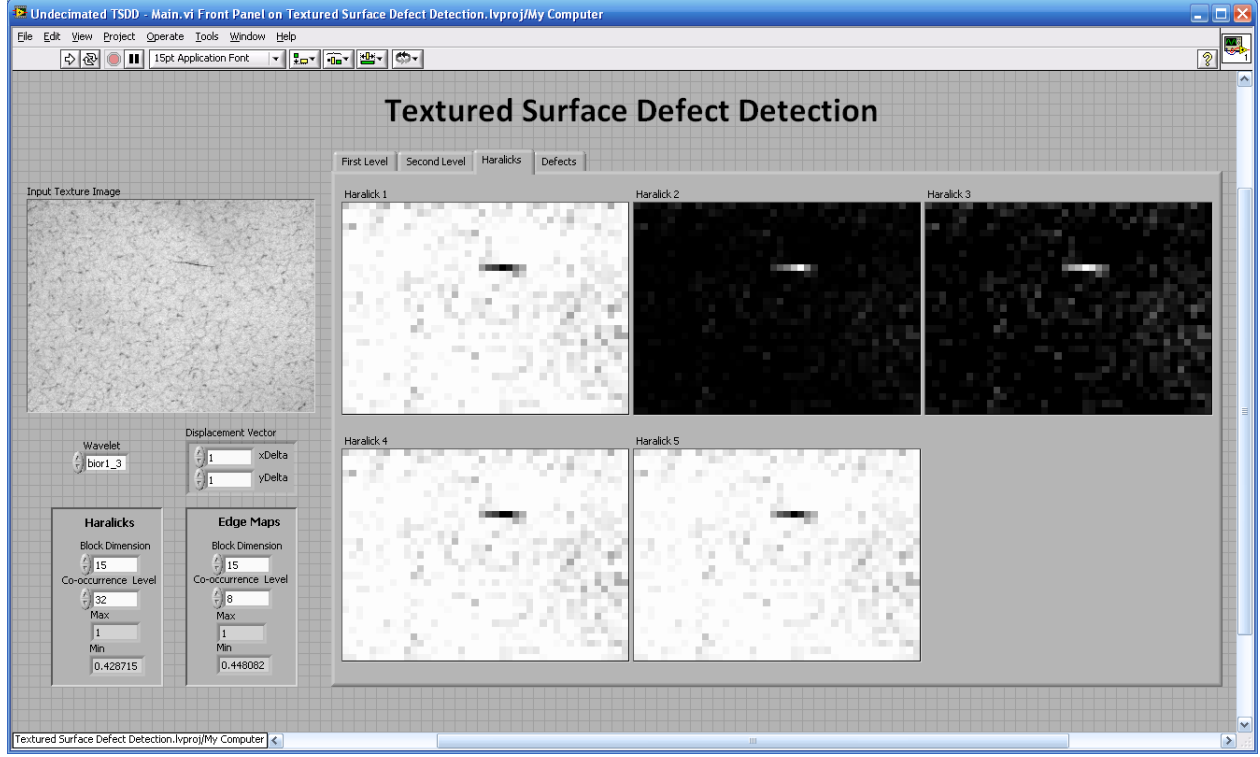


Figure 5. Haralick feature maps extracted from subband H of the example texture image.

where $\mu_i = \sum_{i,j=1}^G i P_{i,j}$ and $\mu_j = \sum_{i,j=1}^G j P_{i,j}$ are the GLCM means; likewise $\sigma_i^2 = \sum_{i,j=1}^G P_{i,j} (i - \mu_i)^2$ and $\sigma_j^2 = \sum_{i,j=1}^G P_{i,j} (j - \mu_j)^2$ are the GLCM variances. Ultimately, each partition of the visual texture is represented by a 60 dimensional feature vector created by concatenating 5 Haralick features from corresponding windows in each of 6 subbands H, V, D, AH, AV, AD , and their smoothed and edge enhanced versions. Figure 5 depicts an example image captured from a textured surface along with five Haralick feature maps extracted from its H subband.

4. SUPPORT VECTOR MACHINES

The classification problem in this study is a classical example of what in pattern recognition community is known as “outlier detection” or “One-class classification”. Here, one of the classes, the normal class, is supported with numerous training samples while the other class, the defect/abnormal class, is represented with very few or no samples at all. For example, defects appearing on manufactured products with textured surfaces have infinite combination of shapes, sizes, and orientations. Hence, even if it was feasible to collect all possible defect instances, it would have been very expensive because we actually have to damage those products. In order to tackle this problem, one class classifiers learn the description of normality by only observing defect-free samples.¹⁹

The simplest one-class classifiers are distance-base decision rules where the fate of each partition is decided by comparing its distance to the average vector of normal samples with an empirically determined threshold. Generally, distance-base classifiers are not desirable because they do not provide a good generalization of the normal class, they are not suitable for high dimensional feature spaces, and the empirically adjusted threshold might be unacceptable. In this study, one-class SVMs that are argued to be among the best outlier detectors are used to learn and generalize normal texture from provided defect-free samples.²⁰

SVM are newly introduced two-class maximum margin classifiers that have become very popular because they perform well in high dimensional feature spaces, avoid over fitting, and have very good generalization capability.²¹ SVMs classify instances of two classes by finding the separating surface that has the maximum

distance to the nearest data points on either side. These closest data points are called the “*support vectors*” and the distances from the separating surface to the nearest instance is called the “*margin*”. Figure 6 shows an example of linearly separable scenario where class membership of data points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ is presented by target variable $t_n \in \{-1, 1\}$. Two candidate linear classifiers are shown in this figure with solid lines along with their corresponding margins and support vectors. The classifying boundary on the left panel has small margin while the one on the right is the maximum margin linear classifier for this example.

Generally, a linear classifier in a p dimensional feature space is a $p - 1$ dimensional hyperplane $\mathbf{w}^T \cdot \mathbf{x} + b = 0$ where \mathbf{w} is the normal vector and $\frac{b}{\|\mathbf{w}\|}$ is the offset from the origin. The margin of each candidate classifier can be visualized by moving two parallel hyperplanes (dashed lines in our 2-D example) on opposite directions until they reach the closest data points. The hyperplanes passing through the support vectors can be described in their canonical form after making isometric rescalings $\mathbf{w} \rightarrow \kappa \mathbf{w}$ and $b \rightarrow \kappa b$ by the equations

$$\begin{aligned} \mathbf{w}^T \cdot \mathbf{x} + b &= 1 \\ \mathbf{w}^T \cdot \mathbf{x} + b &= -1 \end{aligned} \tag{9}$$

Using simple geometry the margin is simply $\frac{2}{\|\mathbf{w}\|}$. Hence in the separable case, the goal is to minimize $\|\mathbf{w}\|$ such that no data point is located between hyperplanes or equivalently:

$$t_n (\mathbf{w}^T \cdot \mathbf{x}_n + b) \geq 1 \quad \text{for all } 1 \leq n \leq N \tag{10}$$

This optimization problem is a quadratic programming problem and the optimum normal vector \mathbf{w} is shown to have the form

$$\mathbf{w} = \sum_n \alpha_n t_n \mathbf{x}_n \tag{11}$$

where coefficients α_n are the solutions of a quadratic programming problem.²¹ Now, new data points \mathbf{x} are classified based on the sign of

$$y(\mathbf{x}) = \mathbf{w}^T \cdot \mathbf{x} + b = \left(\sum_n \alpha_n t_n \mathbf{x}_n \right)^T \cdot \mathbf{x} + b \tag{12}$$

When the points in the training data set are not linearly separable, the constraint (10) is not always satisfied. This problem can be solved by introduction of non-negative slack variables ξ_n s that results in the soft margin linear support vector machines. The goal is to find $\mathbf{w} \in \mathbb{R}^p$, $b \in \mathbb{R}$, and ξ_n s minimizing $(1/N) \sum_n \xi_n + \lambda \|\mathbf{w}\|^2$ under the constraints

$$t_n (\mathbf{w}^T \cdot \mathbf{x}_n + b) \geq 1 - \xi_n \quad \text{for all } 1 \leq n \leq N \tag{13}$$

$$\xi_n \geq 0, \forall n \tag{14}$$

where λ is a soft margin penalty parameter determined by the user. The above concepts are extended to more complex non-linear decision boundaries by Aizerman *et al.*'s kernel trick.²² A non-linear transformation is used to map data points into a higher dimensional space where a linear classifier is subsequently used to classify data points that are separable by a hyperplane in the new feature space. Thus, the decision boundary in the original space is non-linear in the original space.

One-class SVM²⁰ finds a decision surface with minimum volume which encloses a majority of normal examples clustered in the feature space and allows a fraction, ν , of training samples from the normal class to lie outside the boundary of normality. This anticipated training error leaves room for the possible presence of outliers in the training set. The one-class SVM is formulated as finding the optimal hyperplane that separates a desired fraction $1 - \nu$ of the training patterns from the origin of feature space

$$\min_{\mathbf{w}, \xi_n, b} \left(\frac{1}{2} \|\mathbf{w}\|^2 - b + \frac{1}{\nu N} \sum_n \xi_n \right) \tag{15}$$

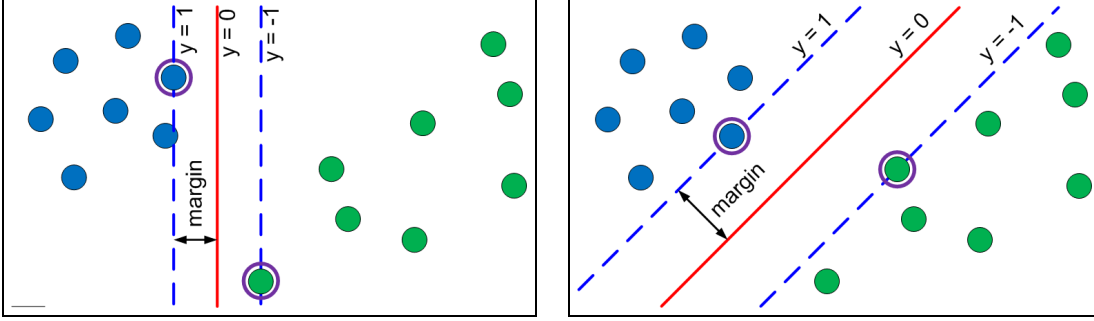


Figure 6. Two candidate linear classifiers (solid lines) shown with their corresponding margins and support vectors. The classifying boundaries on the left panel has small margin while the one on the right is the maximum margin linear classifier for this example.

subject to

$$\mathbf{w}^T \cdot \mathbf{x}_n \geq b - \xi_n, \quad \xi_n \geq 0, \quad \forall n = 1, \dots, N \quad (16)$$

where ν is the fraction of training patterns that are allowed to be rejected and N is the total number of training samples. The One-class SVM is implemented using a publicly available SVM library with interface to LabVIEW.²³

5. RESULTS AND DISCUSSION

The visual inspection algorithm explained in the previous sections is implemented using the LabVIEW graphical programming environment to create a code that exploits parallelism when running on a multi-core or hyper-threaded processors. This algorithm is tested for inspection of real-world tile products obtained from factory environment. The input gray-scale texture images of size 640×480 first undergo two levels of decomposition using over-complete biorthogonal wavelet frames explained in section 2. The wavelet decomposed subbands are subsequently partitioned into non-overlapping 15×15 square windows and each window is represented by 5 Haralick features. Eventually, each window in the original texture image is represented by a 60 dimensional feature vector resulted by concatenating Haralick features from corresponding windows in each of its subbands as described in section 3.

A One-class SVM is trained with samples collected from a defect-free tile image. The rejection parameter ν is selected such that 99% of the partitions present in the defect-free training image are correctly classified as “normal” during the cross-validation. The Gaussian kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})$ is used to transform input data points to a higher dimensional feature space. This transformation facilitates the design of elegant non-linear classifiers in the original feature space by using linear classifiers in the higher dimensional space.

The trained SVM classifier is utilized to inspect 8 images captured from tiles with similar texture patterns. The tile image under inspection is divided into non-overlapping windows and a feature vector is calculated from each window. The fate of the windows under inspection are decided by the SVM according to their location in feature space. Those windows whose corresponding feature points fall inside the decision surface enclosing the majority of defect-free samples are called “normal” and the rest are flagged as “abnormal” partitions.

Figure 7 depicts example tiles inspected by the proposed algorithm. Each row shows the input image next to the corresponding decision-map generated by the one-class SVM. The tiles in figures 7(a), 7(b), and 7(c) have easily identifiable defects while figure 7(d) depicts a tile which is considered to be free of defects by a human inspector. The “false acceptance rate” (FAR) and the “false rejection rate” (FRR) are the two performance metrics measured to evaluate the performance of this visual inspection algorithm. The FAR is the ratio of defective windows classified as normal to the total defective windows and the FRR is the ratio of normal windows mistakenly classified as defective to the total number of normal windows. The inspection results indicate that 99.94% percent of defect-free partitions (human inspected) are correctly labeled as “normal” by this algorithm. In other words, $FRR = 0.06\%$ meaning that only 0.06% of windows identified as normal by a human inspector

are mislabeled as defective. At the same time this algorithm successfully detected 87.5% of partitions with a visible defect or equivalantly $FAR = 12.5\%$.

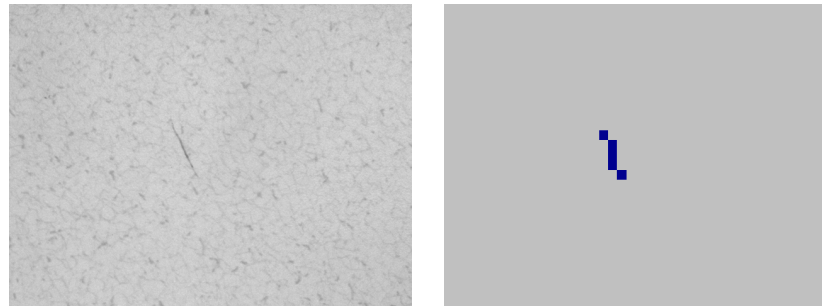
Using the LabVIEW graphical programming environment, it was easy to compose our ideas in a short amount of time into a code that exploits parallelism and runs multiple threads on a hyperthreaded processor. This algorithm is tested for inspection of real world tile products obtained from real factory environment and very promising detection rates are resulted. LabVIEW reduced the inspection time of a 640×480 image from the 100 seconds that it took using our m-file scripts down to 2 seconds measured by running the code on a 1.81 GHz Dual Core processor. LabVIEW implementation is suitable for real-time processing of input image that is essential for an automatic inspection system. In the future we plan to extract features from texture images decomposed by "Steerable Pyramid" (multi-scale oriented linear basis proposed by Simoncelli and Portilla²⁴) and compare the results with the algorithm described in this paper.

ACKNOWLEDGMENTS

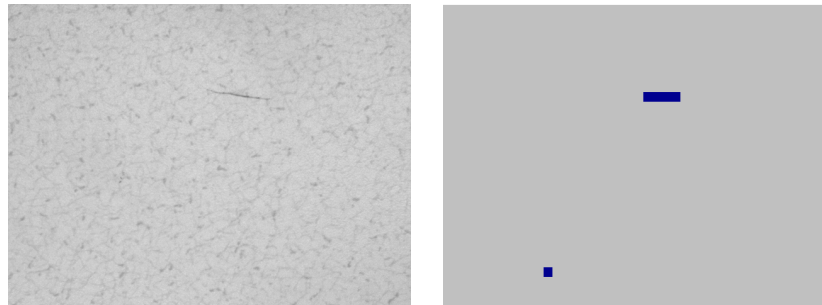
The authors would like to thank National Instruments for supporting this research. LabVIEW , National Instruments and NI are trademarks of National Instruments.

REFERENCES

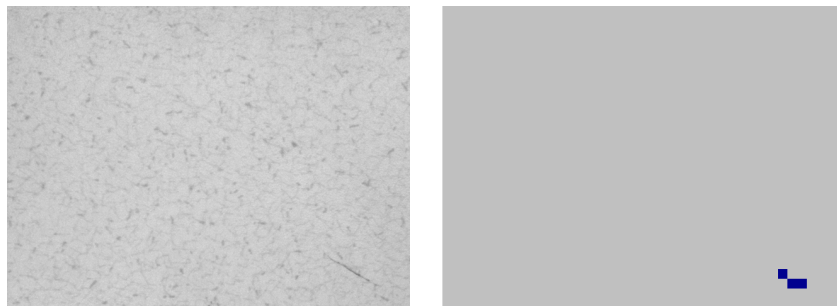
- [1] Andres, N., Marimuthu, R., Eom, Y., and Jang, B., "Development of a machine vision system for automotive part inspection," *Proc. SPIE* **6041**, 60412J, SPIE (2005).
- [2] <http://www.ni.com/vision/vbai.htm>.
- [3] Chellappa, R. and Chatterjee, S., "Classification of textures using Gaussian Markov random fields," *Acoustics, Speech, and Signal Processing, IEEE Transactions on* **33**(4), 959–963 (1985).
- [4] Haralick, R., Shanmugam, K., and Dinstein, I., "Textural Features for Image Classification," *Systems, Man and Cybernetics, IEEE Transactions on* **3**(6), 610–621 (1973).
- [5] Kashyap, R. and Khotanzad, A., "A model-based method for rotation invariant texture classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(4), 472–481 (1986).
- [6] Mallat, S., "A theory for multiresolution signal decomposition: the waveletrepresentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **11**(7), 674–693 (1989).
- [7] Clark, M., Bovik, A., and Geisler, W., "Texture segmentation using Gabor modulation/demodulation," *Pattern Recognition Letters* **6**(4), 261–267 (1987).
- [8] Clark, M. and Bovik, A., "Experiments in segmenting texton patterns using localized spatial filters," *Pattern Recognition* **22**(6), 707–717 (1989).
- [9] Bovik, A., Clark, M., and Geisler, W., "Multichannel texture analysis using localized spatial filters," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **12**(1), 55–73 (1990).
- [10] Lu, C., Chung, P., and Chen, C., "Unsupervised texture segmentation via wavelet transform," *Pattern Recognition* **30**(5), 729–742 (1997).
- [11] Unser, M., "Texture classification and segmentation using wavelet frames," *Image Processing, IEEE Transactions on* **4**(11), 1549–1560 (1995).
- [12] Daugman, J., "Two-dimensional spectral analysis of cortical receptive field profiles.," *Vision Res* **20**(10), 847–56 (1980).
- [13] Petkov, N. and Kruijzinga, P., "Computational models of visual neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: bar and grating cells," *Biological Cybernetics* **76**(2), 83–96 (1997).
- [14] Beck, J., Sutter, A., and Ivry, R., "Spatial frequency channels and perceptual grouping in texture segregation," *Computer Vision, Graphics, and Image Processing* **37**(2), 299–325 (1987).
- [15] Laine, A. and Fan, J., "Texture classification by wavelet packet signatures," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **15**(11), 1186–1191 (1993).
- [16] Antonini, M., Barlaud, M., Mathieu, P., and Daubechies, I., "Image coding using wavelet transform," *Image Processing, IEEE Transactions on* **1**(2), 205–220 (1992).
- [17] <http://sine.ni.com/nips/cds/view/p/lang/en/nid/1395>.



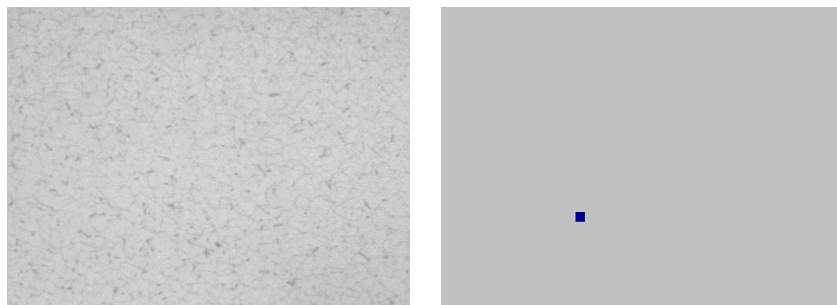
(a)



(b)



(c)



(d)

Figure 7. Example tiles inspected by the proposed algorithm. Each row shows the input image next to the corresponding decision-map generated by the one-class SVM. The tiles in figures 7(a), 7(b), and 7(c) have easily identifiable defects while figure 7(d) depicts a tile which is considered to be free of defects by a human inspector.

- [18] Julesz, B., Gilbert, E., Shepp, L., and Frisch, H., "Inability of humans to discriminate between visual textures that agree in second-order statistics revisited," *Perception* **2**(391-405) (1973).
- [19] Hodge, V. and Austin, J., "A Survey of Outlier Detection Methodologies," *Artificial Intelligence Review* **22**(2), 85–126 (2004).
- [20] Scholkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J., and Platt, J., "Support vector method for novelty detection," *Advances in Neural Information Processing Systems* **12**, 582–588 (2000).
- [21] Vapnik, V., "Statistical Learning Theory. 1998," *NY Wiley* .
- [22] Aizerman, M., Braverman, E., and Rozonoer, L., "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and Remote Control* **25**(6), 821–837 (1964).
- [23] Chang, C.-C. and Lin, C.-J., *LIBSVM: a library for support vector machines* (2001). Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [24] Simoncelli, E. and Portilla, J., "Texture characterization via joint statistics of wavelet coefficient magnitudes," *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on* **1**, 62–66 vol.1 (Oct 1998).