

## EXTRACTION OF NOISE TOLERANT, GRAY-SCALE TRANSFORM AND ROTATION INVARIANT FEATURES FOR TEXTURE SEGMENTATION USING WAVELET FRAMES

MAUSUMI ACHARYYA\* and MALAY K. KUNDU†

*Machine Intelligence Unit  
Indian Statistical Institute  
203 B. T. Road, Kolkata, 700 108, India  
\*mau\_ach@yahoo.co.in  
†malay@isical.ac.in*

Received 5 May 2007

Revised 11 October 2007

In this paper, we propose a texture feature extraction scheme at multiple scales and discuss the issues of rotation and gray-scale transform invariance as well as noise tolerance of a texture analysis system. The nonseparable discrete wavelet frame analysis is employed which gives an overcomplete wavelet decomposition of the image. The texture is decomposed into a set of frequency channels by a circularly symmetric wavelet filter, which in essence gives a measure of edge magnitudes of the texture at different scales. The texture is characterized by local energies over small overlapping windows around each pixel at different scales. The features so extracted are used for the purpose of multi-texture segmentation. A simple clustering algorithm is applied to this signature to achieve the desired segmentation. The performance of the segmentation algorithm is evaluated through extensive testing over various types of test images.

*Keywords:* Feature extraction; texture segmentation; wavelet transform; wavelet frames; nonseparable filters.

AMS Subject Classification: 22E46, 53C35, 57S20

### 1. Introduction

Segmentation of an image based on textural cues into different regions is a difficult task. There are some earlier approaches on texture analysis in literature.<sup>1,2</sup> Several important studies for texture analysis using multiscale wavelet representations include Refs. 3–6, 20 and 29.

The majority of the existing work assumes that all images are acquired from the same line of view, which is an idealistic assumption. A good texture analysis

\*Corresponding author.

approach should ideally be invariant to orientation of the texture. Extraction of rotation invariant texture features is a difficult task. Kashyap *et al.*<sup>7</sup> first realized the importance of rotation-invariant texture feature extraction and developed a texture classification scheme based on autoregressive (AR) model. But the difficulty with this model is that, it could not be used for textures with strong anisotropy.

Cohen *et al.*<sup>8</sup> modeled textures as Gaussian Markov random fields and used the maximum likelihood (ML) method to estimate coefficients and rotation angles. The problem with this method is that the likelihood function is highly nonlinear and local minima may exist, also this algorithm is computationally very expensive. Chen and Kundu<sup>4</sup> used multichannel subband decomposition and a hidden Markov model (HMM) to solve this problem. The two-dimensional (2D) images were decomposed into subbands using quadrature mirror filter (QMF) banks, and modeled features of these subband images as an HMM. Texture samples with different orientations are treated to be in the same class. But it is obvious that for an image the signal components in each subband will be different for different orientations. So variability in texture feature vectors increases with increase in the number of textures to be analyzed. Wu *et al.*<sup>9</sup> tried to solve this problem by converting the 2D textured images into one-dimensional (1D) signals by spiral resampling. In this case, rotational-invariance is simulated as translation-invariance. But this method is also computationally expensive.

Haley and Manjunath<sup>10</sup> describe a method of rotation invariant texture discrimination based on Gabor filters. A multiresolution family of these wavelets is used to compute microfeatures which characterize the spatially located amplitude, frequency and the directional behavior of a texture. However, a large combination of parameters makes texture discrimination using Gabor filters computationally expensive. Most recently Fountain *et al.*<sup>11</sup> have worked on rotation invariant texture features by taking the Fourier transform (FT) of the gradient direction histograms of the textures. Chantler and McGunnigle<sup>17</sup> proposed a novel scheme that is surface-rotation invariant. It uses the eigenvalues of a surface gradient-space distribution as its features. A distribution-based classification and a set of texture measures based on center-symmetric auto-correlation and local binary patterns are applied to rotation invariant texture classification.<sup>18</sup> Later on Ojala *et al.* modified and extended this to take care of gray scale invariance.<sup>19</sup> A large number of related works may be found in the survey on rotation invariant texture classification.<sup>14</sup> Circular Gabor filter (CGF) has been used for rotation invariant texture segmentation.<sup>15</sup> The rotation invariant textures are achieved via the channel output of CGF.

Due to different imaging equipments and techniques, scenes which are imaged may have different gray scale values. The result is due to a complex process and is not easily modeled. Chen and Kundu<sup>4</sup> used a linear function with an offset to describe the phenomenon. Using this model a transformed gray-scale may be corrected by histogram equalization, but this increases the complexity of the classifier. They used coefficients of skewness and kurtosis to describe features invariant to gray-scale transformation. Wu *et al.*<sup>9</sup> solved this problem in a better way.

Despite all these research we are of the opinion that a more general way to cope with this problem is to incorporate rotational information in the texture features themselves.

In the present work, texture properties are characterized by wavelet frame analysis mainly as suggested in Ref. 12. While discrete wavelet transform gives a nonredundant representation of the textures, the discrete wavelet frame gives an overcomplete representation. This technique is employed to study the performance of a texture segmentation problem with respect to rotational invariance, in particular Refs. 13 and 16. Also invariance to gray-scale transform and immunity to noise is being extensively studied.

The two-dimensional (2D) implementation of the discrete wavelet transform (DWT) using a filterbank<sup>21,22</sup> is obtained in two steps by successive application of the 1D filtering along rows and columns of an image. Due to the separable nature of implementation of the 2D DWT, it is strongly oriented in the horizontal and vertical directions. Such a decomposition cannot efficiently characterize directions other than 0° and 90°. This is particularly inappropriate while rotated textures are taken into consideration. So we need a nonseparable implementation of the wavelet transform in 2D. We make use of a wavelet filter which is circularly symmetric and hence is invariant to rotation.

Section 2 describes the wavelet parameter computation which gives the features for texture discrimination and integration of these features. In Sec. 3, we give the formulations of the different performance measure that we have used in this experiment. Section 4 presents our results and critical remarks about the performance and finally, in Sec. 5, we conclude our study.

## 2. Multiscale Feature Extraction

The proposed feature extraction scheme consists of three main stages given in Fig. 2. It consists of a filtering stage and a subsequent nonlinear stage followed by a smoothing filter (both these constitute the local energy estimator). Basically, the purpose of the filter is to extract local frequencies of the textures, and the objective of local energy estimator, is to estimate the energy of the filter output in a local region.

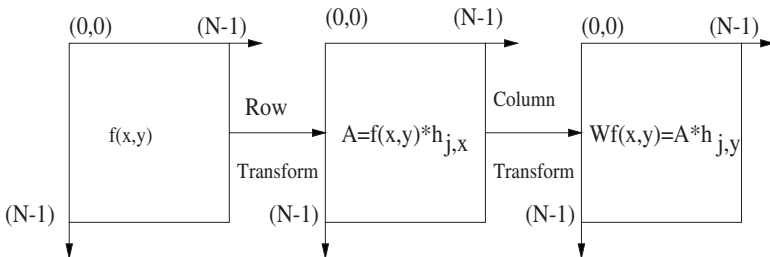


Fig. 1. Separable filtering for 2D discrete wavelet transform.

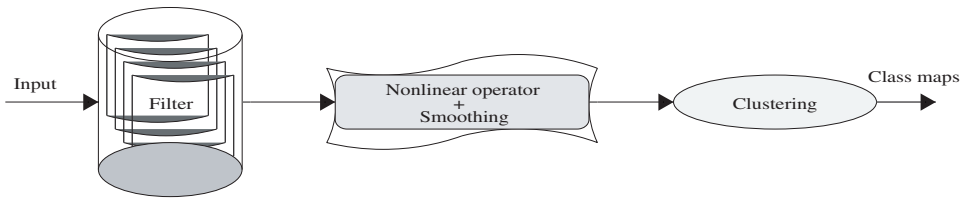


Fig. 2. Fast iterative implementation of the algorithm used for extracting texture features.

The directional information, can be incorporated in the 2D wavelet function, by including a rotational parameter in it<sup>23</sup>

$$\psi_{a,\theta}(\vec{b}) = \frac{1}{a} \psi \left( R^\theta \left( \frac{\vec{x} - \vec{b}}{a} \right) \right) \tag{2.1}$$

where  $R^\theta$  is the rotation operator denoted by the matrix

$$\begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}.$$

Now if the wavelet is circularly symmetric i.e.  $R^\theta$  has no influence in (2.1), such a wavelet would generate rotation invariant features. We have chosen the wavelet used by Mallat,<sup>24</sup> which is the second derivative of a smoothing function. This choice has been made because this closely approximates the second derivative of Gaussian, which has circular symmetry and the basis functions are symmetrical, which means that there is no phase distortion.

**2.1. Multiscale 2D wavelet transform**

The technique used by Mallat<sup>25</sup> using discrete wavelet frames has been used in the present work (Fig. 3). Let  $\phi(x, y)$  be a smoothing function. We call the smoothing

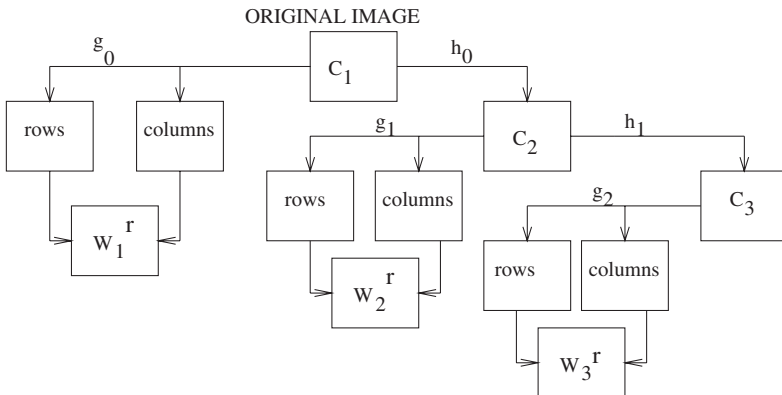


Fig. 3. The wavelet decomposition scheme.

function the impulse response of a lowpass filter with a total mass of one and compact support, i.e.

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \phi_s(x, y) dx dy = 1, \quad \text{and} \tag{2.2}$$

$$\exists \epsilon > 0 : \phi_s(x, y) = 0, \quad \forall |x|, |y| > \epsilon.$$

Considering  $\phi(x, y)$  to be double differentiable, we define two wavelet functions,  $\psi^1(x, y)$  and  $\psi^2(x, y)$  such that,

$$\psi^1(x, y) = \frac{\partial^2 \phi(x, y)}{\partial^2 x} \quad \text{and} \quad \psi^2 = \frac{\partial^2 \phi(x, y)}{\partial^2 y}. \tag{2.3}$$

Let,  $\phi_s(x, y) = \frac{1}{s} \phi(\frac{x}{s}, \frac{y}{s})$  be the dilation of  $\phi(x, y)$  by  $s$ . Then,

$$\psi_s^1(x, y) = \frac{1}{s^3} \psi^1\left(\frac{x}{s}, \frac{y}{s}\right) \quad \text{and} \quad \psi_s^2(x, y) = \frac{1}{s^3} \psi^2\left(\frac{x}{s}, \frac{y}{s}\right), \tag{2.4}$$

be the dilations of the functions  $\psi^i$  by a factor  $s$ .

Let  $I(x, y)$  be an image in 2D and  $I(x, y) \in L^2(R)$ . The wavelet transform of  $I(x, y)$  at scale  $s$  has two components defined by,

$$w_s^1(x, y) = I * \psi_s^1(x, y) \quad \text{and} \quad w_s^2(x, y) = I * \psi_s^2(x, y), \tag{2.5}$$

where  $s$  is the dilation or scale parameter and usually has values equal to  $2^s$  with  $s = 0, \dots, d_0$ . This gives the traditional octave band wavelet transform of depth  $d_0$ .  $w_s^1$  and  $w_s^2$  contain the horizontal and vertical frequency information of  $I$  at scale  $s$  and are referred to as the detail images.

The computation of the 2D transform is carried out by means of a set of lowpass filters  $h_s$  associated with the smoothing function  $\phi$  and bandpass filters  $g_s$  associated with the wavelets  $\psi^1$  and  $\psi^2$  and have finite impulse responses. The filtering is done in an iterative manner.

$$\begin{aligned} c_{2^{s+1}}(x, y) &= [h_{s,x} * [h_{s,y} * c_{2^s}]](x, y), \\ w_{2^{s+1}}^1(x, y) &= [\delta_{s,x} * [g_{s,y} * c_{2^s}]](x, y), \\ w_{2^{s+1}}^2(x, y) &= [g_{s,x} * [\delta_{s,y} * c_{2^s}]](x, y), \end{aligned} \tag{2.6}$$

where  $c_1 I$  is the original image and  $\delta$  is the Dirac filter whose impulse response equals 1 at 0 and 0 otherwise. The “\*” denotes the convolution operator.

Then the wavelet representation of depth  $d_0$  of the image  $I(x, y)$  consists of the low resolution images  $\{c_{2^s}\}$  and detail images  $\{w_{2^s}^j\}$  for  $\{j = 1, 2\}$  and  $\{s = 1, \dots, d_0\}$ .

### 2.2. Rotation invariant features

In this section we will be explaining how we extract the rotation invariant features from wavelet transformed images.<sup>12,13</sup>

Substitution of (2.3) and (2.5) in (2.4) yields the following,

$$\begin{aligned} \begin{pmatrix} w_s^1(x, y) \\ w_s^2(x, y) \end{pmatrix} &= s^2 \begin{pmatrix} \frac{\partial^2}{\partial^2 x} (I * \phi_s)(x, y) \\ \frac{\partial^2}{\partial^2 y} (I * \phi_s)(x, y) \end{pmatrix} \\ &= s^2 \nabla^2 (I * \phi_s)(x, y) \end{aligned} \tag{2.7}$$

where  $\nabla^2$  denotes the Laplacian. It defines edge magnitudes of the image and since it has the same property in all directions, is invariant to rotations in the image. That is, the wavelet transform of an image consists of the components which give a measure of the edge magnitudes of the image, smoothed by the dilated smoothing function  $\phi_s$ . The edge magnitudes of the image is given as,

$$w_s^r(x, y) = \sqrt{(w_s^1(x, y))^2 + (w_s^2(x, y))^2}. \tag{2.8}$$

The  $w_s^r(x, y)$  contain a measure of the edge magnitude at  $(x, y)$ , which in turn gives a measure of the local gray level variation of the image. A rotation invariant multi-scale representation of the image can be formulated as  $h_s = \{(w_s^r)_{(s=0, \dots, 2^{d_0})}, c_{2^{d_0}}\}$ . Figure 4 shows some of the textured images that we have used to generate the

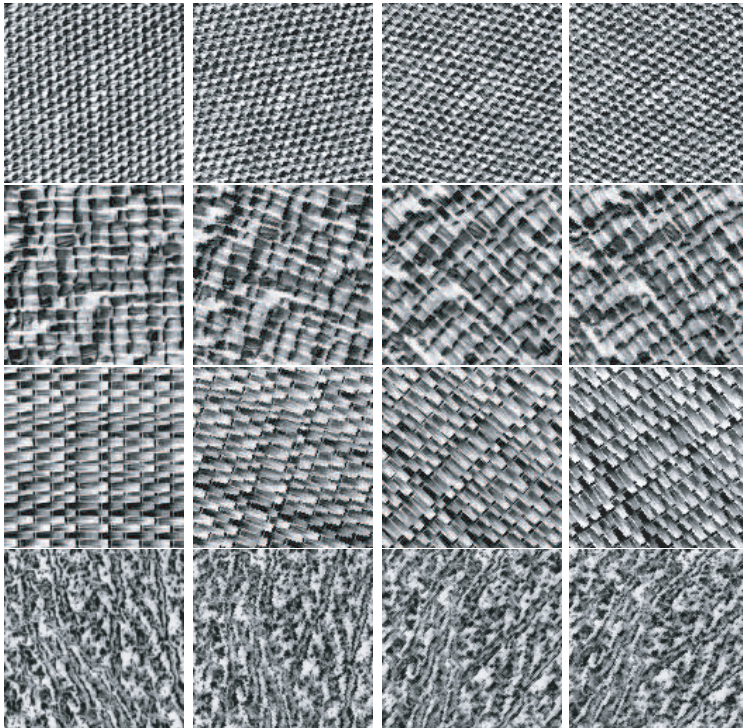


Fig. 4. (From left to right row-wise) Rotated samples ( $0^\circ, 20^\circ, 45^\circ$  and  $55^\circ$ ) of some of 4 different textured images used in our experiment and their gradient magnitude histogram plots.

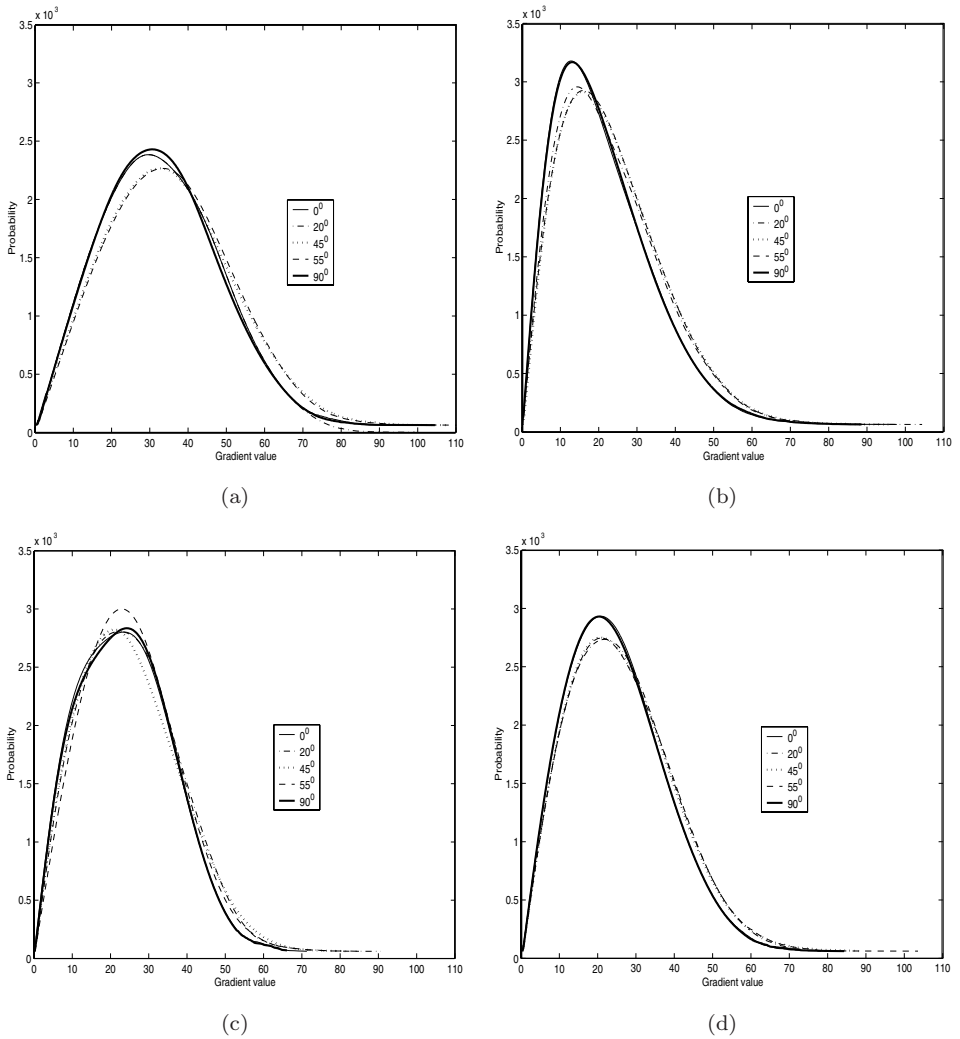


Fig. 4. (Continued)

composite textures used in our experiment. We find out the histograms of the gradient values given by (2.8) of these textures at different rotations. The histograms so obtained has many local minima and maxima which can be removed by local smoothing of the histogram. This is accomplished by local averaging of neighboring histogram elements.

We have taken three adjacent elements at a time and the process is repeated for a number of iterations. A plot of these histograms after smoothing is given in Fig. 4 to give a feeling that the edge magnitudes of the textures are indeed rotation invariant. We calculate the mean and second and the third order central moments of the unrotated texture edge magnitude histogram and also its several rotated

versions (Table 9), to measure the similarity between these histograms, the error percentage is around 0.4%.

### 2.3. *Gray-scale transformation invariant features*

The gray-scale transform mentioned in Sec. 1 can be described as follows.

$$I_t(x, y) = \alpha I(x, y) + \delta \quad (2.9)$$

where,  $I(x, y)$  is the gray level at  $(x, y)$ ,  $\alpha$  is a positive scale factor, and  $\delta$  is a shift factor. We assume that  $0 \leq I_t(x, y) \leq 255.0$ . In texture analysis we require an algorithm that is invariant to gray-scale transformation. We remove the mean of the image to nullify the effect of  $\delta$ . The effect of scale factor  $\alpha$  is removed in formulation of the feature. The features obtained according to Eq. (2.8) can be modified while considering the gray-scale transformed image (2.9) as,

$$Wgt_s^r(x, y) = \alpha \sqrt{(w_s^1(x, y))^2 + (w_s^2(x, y))^2}. \quad (2.10)$$

The above Eq. (2.10) can be rewritten by taking logarithm on both sides as,

$$\log(Wgt_s^r(x, y)) = \log(\alpha) + \log(Wg_s(x, y)). \quad (2.11)$$

It can be written otherwise in the following form as,

$$WGT_s^r(x, y) = \Gamma + WG_s(x, y). \quad (2.12)$$

It is very apparent from Eq. (2.12) that the additive factor  $\Gamma$  can be eliminated by removing the mean from the image  $WGT_s^r(x, y)$ . The features are then obtained by exponentiating  $WGT_s^r(x, y)$  whose mean has been removed. Figure 5 shows two textured images that we have used to generate the composite textures used in our experiment. We find out the histograms of the exponentiated values given by (2.12), of these textures for different shift and scale factors. The histogram so obtained has many local minima and maxima which can be removed by local smoothing of the histogram. This is accomplished by local averaging of neighboring histogram elements as explained in Sec. 2.2.

A plot of these histograms after smoothing is given in Fig. 5 to give a feeling that the features characterizing the textures are indeed invariant to any gray-scale transforms.

### 2.4. *Noise invariant features*

Real images are often degraded by some random errors, this degradation is usually called noise. Noise arises as a result of unmodeled or unmodelable processes going on in the production and capture of the real signal. It is not part of the ideal signal and may be caused by a wide range of sources, e.g., variations in the detector sensitivity, environmental variations, the discrete nature of radiation, transmission or quantization errors, etc. It is also possible to treat irrelevant scene details as if

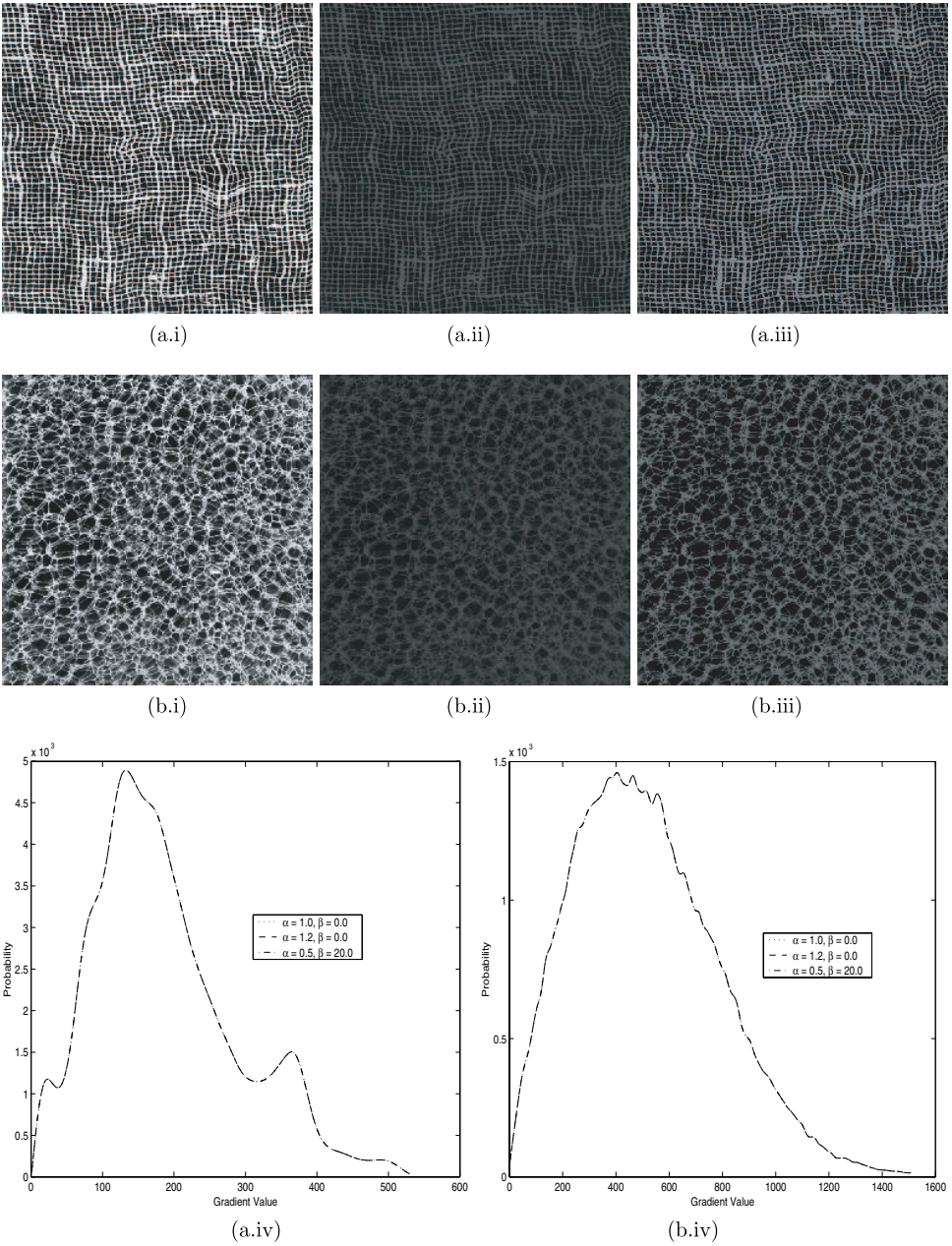


Fig. 5. (a.i)–(b.i) Textures without gray-scale transforms, textures with gray-scale transforms for different scale and shift factors; (a.ii)–(b.ii)  $\alpha = 0.5, \delta = 20.0$ ; (a.iii)–(b.iii)  $\alpha = 1.2, \delta = 0.0$ ; (a.iv)–(b.iv) Edge magnitude histogram plots.

they are image noise (e.g., surface reflectance textures). The characteristics of noise depend on its source.

Many image processing packages contain operators to artificially add noise to an image. Deliberately corrupting an image with noise allows us to test the resistance of an image processing operator to noise and assess the performance of various noise filters.

Noise can be generally grouped into two classes: independent noise and noise which is dependent on the image data.

Image independent noise can often be described by an additive noise model and is called additive noise. Whereas in the second case of data-dependent noise (e.g., arising when monochromatic radiation is scattered from a surface whose roughness is of the order of a wavelength, causing wave interference which results in image speckle), it is possible to model noise with a multiplicative, or nonlinear, model.

The 2D wavelet transform that we have implemented gives a multiscale zero crossings representation of the image. So, zero crossings appear at those locations where there are gray level differences corresponding to edges and also appear at nonedge locations with increasing noise levels. However, our purpose is not to locate the edge points rather we estimate the local energy around each pixel over small overlapping windows by some nonlinear operation followed by smoothing, to characterize the 2D signal. So the features so extracted have inherent noise regularization.

## **2.5. Local energy estimation**

The next step is to estimate the energy of the filter responses in a local region around each pixel. The local energy estimate is utilized for the purpose of identifying areas in each channel where the bandpass frequency components are strong resulting in a high energy value and the areas where it is weak into a low energy value.

Wavelet coefficients are insufficient for texture cues. They are helpful to split textured information into different frequency channels. Then we must consider the locality of the pixels with local statistics. A nonlinearity is needed in order to discriminate texture pairs with identical mean brightness and second-order statistics. We have used modulus operator as the nonlinearity. One reason for choosing this nonlinear operator is that it is parameter free, meaning it is independent of the dynamic range of the input image and also of the filter amplification. To calculate features of an image, we slide a fixed size window on the wavelet coefficients of an image and compute the local statistics in each individual window and associate these values as feature values of the centered pixels of these windows. There are a wide variety of textural measures. In this study energy measure for texture segmentation is considered. The energy features measure textural uniformity, i.e. pixel pairs repetitions. Energy is usually defined in terms of a squaring nonlinearity. However in a generalized energy function, other alternatives are used. *Average absolute deviation* from the mean small overlapping windows has been used as a generalized energy definition in this work to get separation of features for different patterns.

The local energy  $\text{Eng}_s(x, y)$  around the  $(x, y)$ th pixel of the filtered image  $F_s(x, y)$  in the subband  $s$  is formally given as,

$$\text{Eng}_s(x, y) = \frac{1}{R} \sum_{m=1}^W \sum_{n=1}^W |(F_s(m, n) - \bar{F}_s(x, y))| \quad (2.13)$$

where,  $W$  is the window size and  $R = W \times W$ , while  $\bar{F}_s(x, y)$  is the mean around the  $(x, y)$ th pixel and  $F_s(x, y)$  is the filtered image at different scales for  $s = 0, \dots, 2^{d_0}$ .

The nonlinear transform is followed by a Gaussian low pass (smoothing) filter of size  $G \times G$  (say). Formally, the feature image  $\text{Feat}_s(x, y)$  corresponding to filtered image  $F_s(x, y)$  is given by,

$$\text{Feat}_s(x, y) = \frac{1}{G^2} \sum_{(a,b) \in G_{x,y}} |\Psi(F_s(a, b))| \quad (2.14)$$

where  $\Psi(\cdot)$  is the local energy estimator and  $G_{x,y}$  is a  $G \times G$  window centered at pixel with coordinates  $(x, y)$ . The size  $G$  of the smoothing or the averaging window in Eq. (2.14) is an important parameter. Using a Gaussian weighting window results in less sparse points (i.e. denser feature distributions) as compared to when no weighting window is used. The local *average absolute deviation* values from the mean (2.13) of a Gaussian window give us a robust quality of features in the feature space in all of our test images. However, one can use other local statistics depending upon the applications. Another issue with this regard is the local window size itself. After a number of experiments we have found that the choice of the local window size is very crucial to find proper features. In an image with patterns of different texel sizes, we have to choose a suitable window size. If we choose a larger window for the largest texel size, we introduce more uncertainty on the boundaries regions. This problem can be solved if the effective window size changes with the level of resolution. These various window sizes will catch textures with different texel sizes. For larger texels, it is better to choose larger local window size and for smaller texels it should be kept smaller. Likewise, the size  $G$  of the Gaussian averaging window is also an important parameter. More reliable measurement of texture feature demands larger window size. On the other hand, more accurate localization of region boundaries requires smaller window. This is because averaging operation blurs boundaries between textured regions. Therefore, Gaussian weighted windows are naturally preferable over unweighted windows, because, the former are likely to result in more accurate localization of texture boundaries.

This step results in a set of feature images  $\text{Feat}_s(x, y)$  from which a set of feature vectors are derived. These feature vectors, corresponding to the decomposed images at different resolutions, are assumed to capture and characterize effectively different scales of texture of the input image. The oversampled wavelet transforms introduce redundancy in filtered images which may be useful for a reliable result in a recognition problem. The redundancy of information will present itself as more patterns for each cluster in feature space. In many clustering algorithms the reliability of final results will be increased by having more feature vectors per cluster.

Having obtained the feature images, the main task is to integrate these feature images to produce a segmentation. Energy values have to be normalized to ensure that they lie within the same dynamic range. This ensures that the energy values in the nondominant spatial frequency channels provide important information for discriminating between textures during the clustering process. The energy values are normalized to lie between 0 and 1 so that they can be conveniently used for segmentation.

We emphasize on the feature extraction (representation) part in this work. So we have used a traditional *k-means* clustering algorithm. Although other sophisticated algorithms like watershed clustering could have been used.

### 3. Performance Evaluation Measures

We define to following measures of classification accuracy, to evaluate the performance of our algorithm, where ground truth is available.

**Percentage of correct classification.** We provide the performance results in terms of percentage of correct classification, which is defined as

$$\frac{n_{ic}}{n_i} * 100 \quad (3.1)$$

where  $n_i$  is the number of pixels belonging to class  $i$ , of which  $n_{ic}$  pixels have been correctly classified.  $\frac{n_{ic}}{n_i}$  is also called the Producer's accuracy.<sup>26</sup>

In this section we provide two more statistical measures, namely User's accuracy and Kappa following Refs. 27 and 28.

**User's accuracy.** If  $n'_i$  pixels (of all pixels  $n$ ) are found to be classified into class  $i$ , then the user's accuracy ( $U$ ) is defined as

$$U = \frac{n_{ic}}{n'_i} \quad (3.2)$$

where  $n_{ic}$  is as defined earlier. The user's accuracy gives a measure of the confidence that a classifier attributes to a region as belonging to a class. That is, it denotes the level of purity associated with a region.

**Kappa.** The coefficient of agreement called "kappa" measures the relationship of beyond chance agreement to expected disagreement. The estimate of kappa ( $K$ ) is the proportion of agreement after chance agreement is removed from consideration. The estimate of kappa for class  $i$  ( $K_i$ ) is defined as,

$$K_i = \frac{n * n_{ic} - n_i * n'_i}{n * n'_i - n_i * n'_i}. \quad (3.3)$$

The numerator and denominator of the overall kappa are obtained by summing the respective numerators and denominators of  $K_i$  separately over all classes  $i$ .

#### 4. Experimental Results

We have experimented our segmentation algorithm on a number of textured images taken from the Brodatz album which clearly exhibit anisotropy. Several rotations have been considered, like  $15^\circ$ ,  $20^\circ$ ,  $37^\circ$ ,  $45^\circ$ ,  $55^\circ$ ,  $60^\circ$  and  $90^\circ$ . Several composite textured images have been generated from the rotated versions of these textured images. Since we are mainly interested in the effectiveness, reliability and robustness of the rotation invariant and noise immune features that are extracted rather than the segmentation performances, we have experimented on composite textured images consisting of a moderate number of texture classes.

Local window sizes  $w \times w$  are  $5 \times 5$ ,  $9 \times 9$  and  $17 \times 17$  for the first, second and third level of resolutions. Also the averaging window sizes are chosen in commensuration with the local windows. Figure 6 shows two class test images that we have worked on, which consist of 5 regions, out of which two regions are unrotated textures and the rest three comprise of rotated textures. The images were decomposed into three levels of resolution as given in Fig. 3. We have taken into account all the three detail images and two of the low frequency images leaving out the lowest frequency image at the third level of resolution, to estimate the local energies around each pixel at different scales. That is in all we are left with only five feature elements which

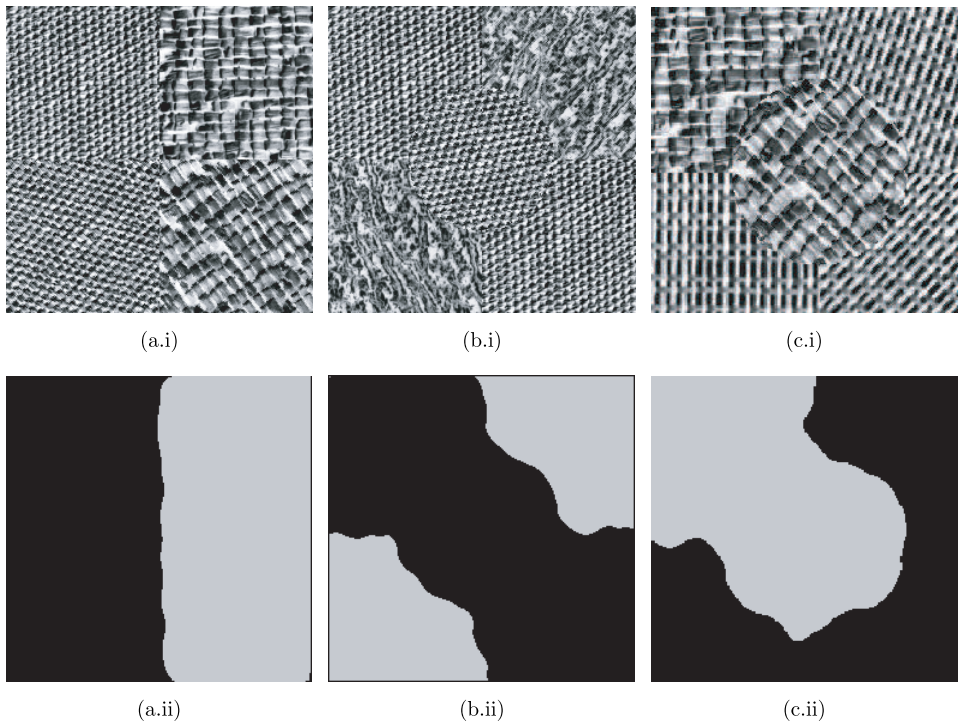


Fig. 6. Composite textured images with their corresponding segmented outputs ( $k = 2$ ).

means a huge reduction in the dimensionality of the feature space, compared to other methods so far reported in the literature. Segmentation was performed with *k-means* clustering with fixed  $k = 2$ , results are shown in Fig. 6. The classification percentages of test images in Fig. 6 are tabulated in Table 1.

Figure 7 shows some composite textured images that we have tested on, which comprise of three different texture classes. The images consist of 5 regions with three unrotated textured regions and two rotated textured regions. Segmentation is carried out using *k-means* clustering with fixed  $k = 3$  the experimental results are shown in Fig. 7. The classification percentages for test images in Fig. 7 are given in Table 1.

Figure 8(a.i) shows another composite textured image that we have tested on, which comprise of five different texture classes. The image consist of 16 regions with several rotated and unrotated textured regions. The segmentation result is given

Table 1. Percentage of correct classification corresponding to the test images Figs. 6 and 7.

Figure No.	6.a.i	6.b.i	6.c.i	7.a.i	7.b.i	7.c.i
% Classification	99.31	98.93	97.41	98.24	97.61	98.49

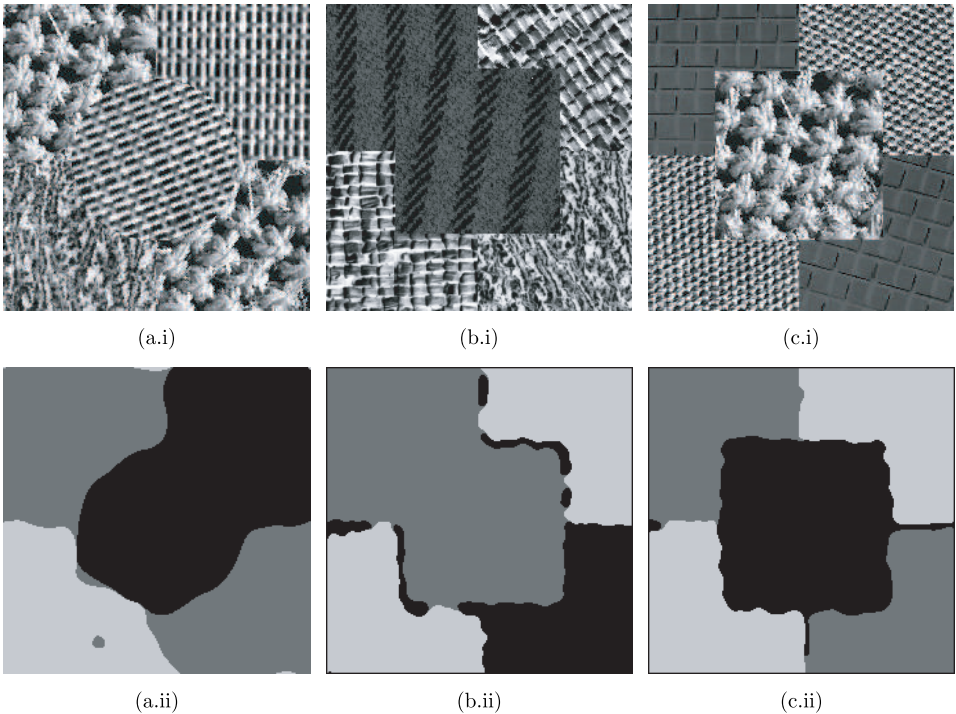


Fig. 7. Composite textured images with their corresponding segmented outputs ( $k = 3$ ).

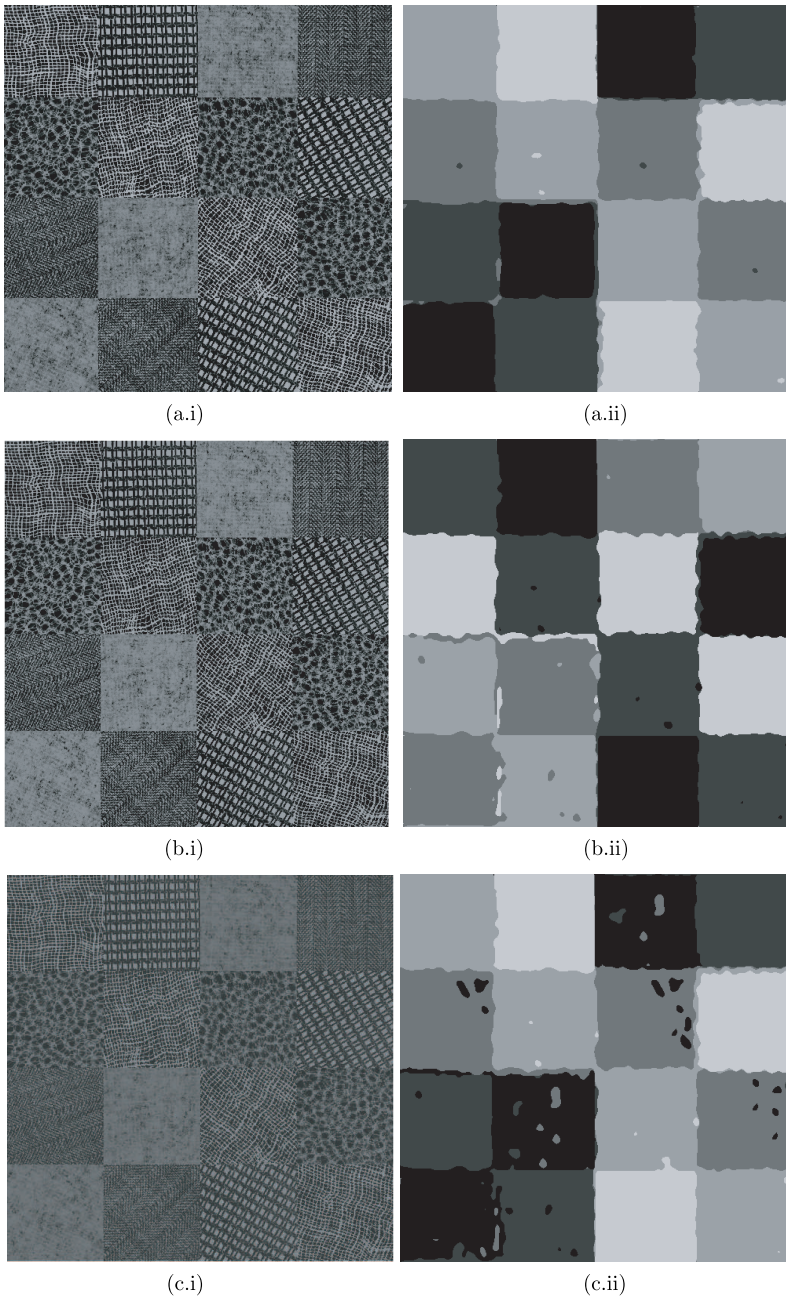


Fig. 8. Composite textured images; (a.i) Without gray-scale transform, with gray-scale transform; (b.i)  $\alpha = 1.2$ ,  $\delta = 0$ ; (c.i)  $\alpha = 0.5$ ,  $\delta = 20$ ; (a.ii)–(b.ii)–(c.ii) Corresponding segmented outputs ( $k = 5$ ).

in Fig. 8(a.ii). To verify the invariance to gray-scale transformation, we performed some experiments on the same image. For this purpose two different set of parameters were used:  $\alpha = 1.2, \delta = 0$  (Fig. 8(b.i) and  $\alpha = 0.5, \delta = 20$  (Fig. 8(c.i)), segmentation results are shown in Figs. 8(b.ii) and 8(c.ii) The performance of segmentation was not affected by gray-scale transformation, Table 2 validates this result.

The same image in Fig. 8(a.i) is corrupted with various Gaussian noise levels (Fig. 9). The result obtained is very encouraging as shown in Fig. 10, which gives

Table 2. User accuracy (U), kappa values (K) and classification in percentage for different scale and shift factors in Fig. 8.

Parameter	Values	Overall		% Classification
		U	K	
$\alpha$	1.0			
$\beta$	0	96.76	95.85	96.73
$\alpha$	1.2			
$\beta$	0	96.25	95.36	95.74
$\alpha$	0.5			
$\beta$	20	95.12	94.19	94.69

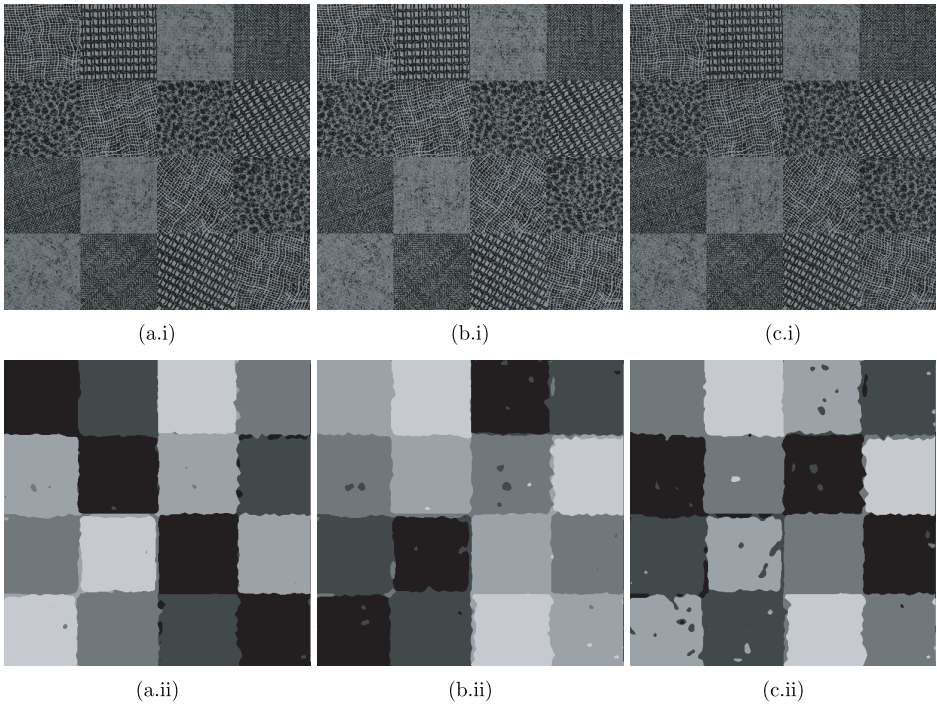


Fig. 9. (a.i)–(c.i) Composite textured images corrupted with Gaussian noise with their corresponding segmented outputs ( $k = 5$ ); (a.i) Mean = 0, Variance = 100.0; (b.i) Mean = 0, Variance = 200.0; (c.i) Mean = 0, Variance = 300.0.

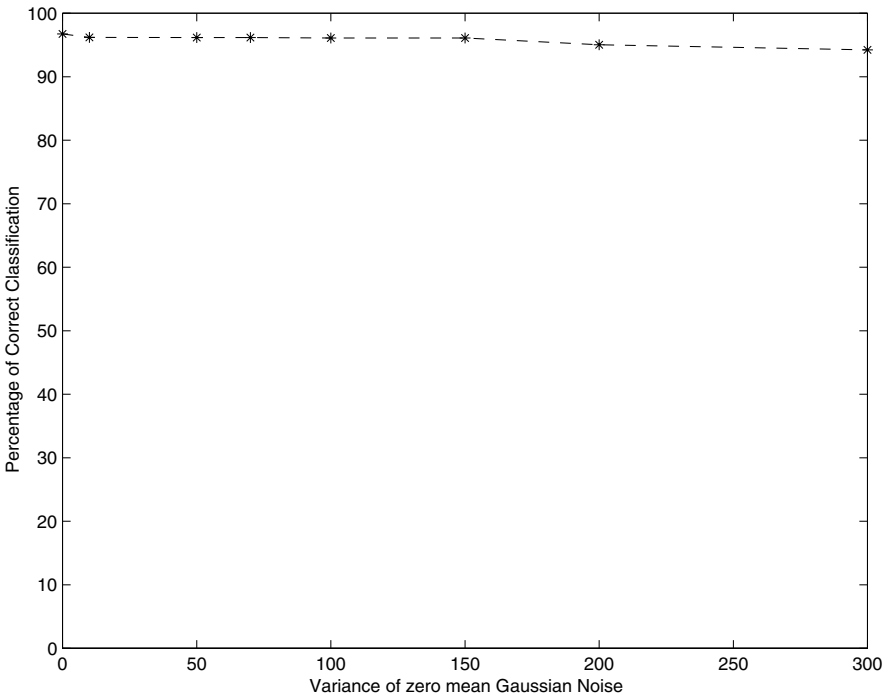


Fig. 10. Percentage of correct classification with Gaussian noise (noise parameters are given in Table 3).

Table 3. Percentage of correct classification corresponding to the test image Fig. 8(a.i) for different Gaussian noise levels.

Mean	0	0	0	0	0	0	0	0
Variance	0	10.0	50.0	70.0	100.0	150.0	200.0	300.0
% Classification	96.73	96.19	96.17	96.16	96.10	96.09	95.04	94.23

a plot of percentage of correct classification for various noise levels. The percentage of correct classification for different noise levels is tabulated in Table 3. The robustness of the proposed algorithm against noise is verified by experimenting on the test image of Fig. 8(a.i) corrupting it with different types of noise, these include Gaussian, Rayleigh, Exponential, Poisson, Uniform, Shot and Speckle noise having different parameter attributes [Figs. 9 and 11–13], and are tabulated in Tables 4 and 5 (noise was generated by “KHOROS” in Silicon Graphics Irix 5.3). Segmentation is carried out using  $k$ -means clustering with fixed  $k = 5$ . The experimental results are shown in Figs. 9 and 11–13. The classification percentages, the user accuracy (U) and kappa (K) values for each of the constituent class in the composite image are given in Tables 4 and 5.

Table 4. User accuracy (U), kappa values (K) and correct classification in percentage corresponding to the different types of noise for data in Figs. 8, 9, 11–13.

Type of Noise	Parameter Attributes	Values	Effect	Overall		% Classification
				U	K	
Without noise	—	—	—	96.76	95.85	96.73
Gaussian	Mean	0	Additive	96.09	95.10	96.10
	Variance	100.0				
	Mean	0	Additive	95.20	94.00	95.04
	Variance	200.0				
	Mean	0	Additive	94.41	93.02	94.23
	Variance	300.0				
Rayleigh	Variance	100.0	Additive	95.24	94.05	95.06
	Variance	200.0	Additive	94.44	93.05	94.32
	Variance	300.0	Additive	94.00	92.62	93.92

Table 5. User accuracy (U), kappa values (K) and correct classification in percentage corresponding to the different types of noise for data in Figs. 9 and 11–13.

Type of Noise	Parameter Attributes	Values	Effect	Overall		% Classification
				U	K	
Exponential	Variance	200.0	Additive	95.69	94.62	95.62
Poisson	Amount of time	200	Additive	95.53	94.41	95.41
	Variance	20.0				
Shot	% of spikes	50	Additive	94.88	93.60	94.87
	Value of spike	30.0				
Uniform	Minimum value	−20.0	Additive	95.31	94.16	93.31
	Maximum value	+20.0				
Speckle	Mean	1.0	Multiplicative	94.19	92.74	94.19
	Standard	+0.28				
	Deviation					

To prove the efficacy of our algorithm we have tested on another test image Fig. 14 (Fig. 7(c.i)) contaminated with Gaussian, Exponential and Rayleigh noise. The percentage of correct classification, user accuracy (U) and Kappa (K) of the segmented results are given in Table 6.

The above study reveals that we have good quality segmentation invariant to rotation and also without appreciable degradation in performances while noise is being incorporated. It is to be noted that we have done median filtering of window size (9 × 9) as a post processing step over the classified images, to simulate the benefit of local neighborhood information.

For a comparison purpose, We have studied the segmentation result using circular Gabor filter.<sup>15</sup> Figure 17(a.ii) shows the segmented output of the test image

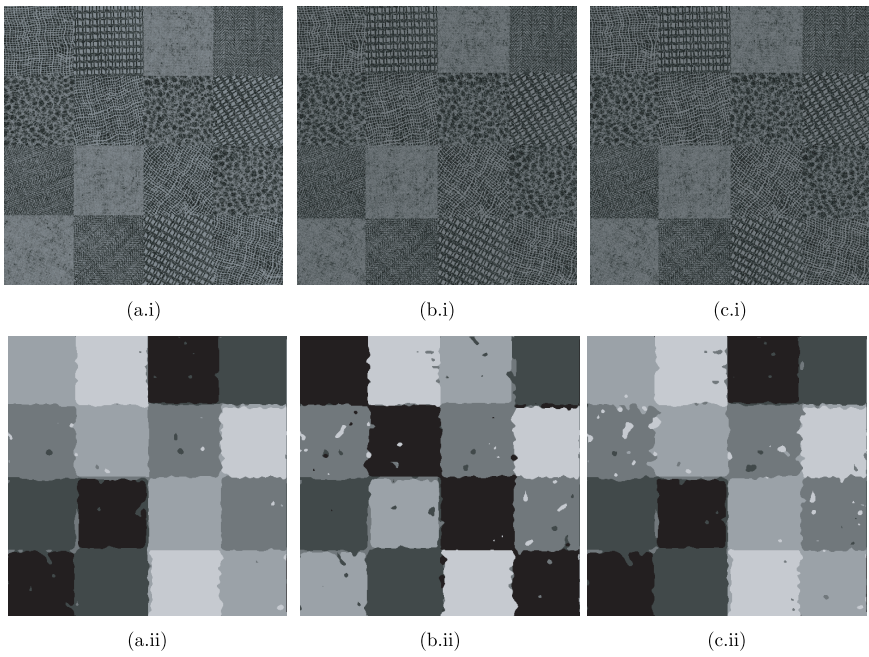


Fig. 11. (a.i)–(c.i) Images corrupted with Rayleigh noise; (a.i) Var = 100; (b.i) Var = 200; (c.i) Var = 300.

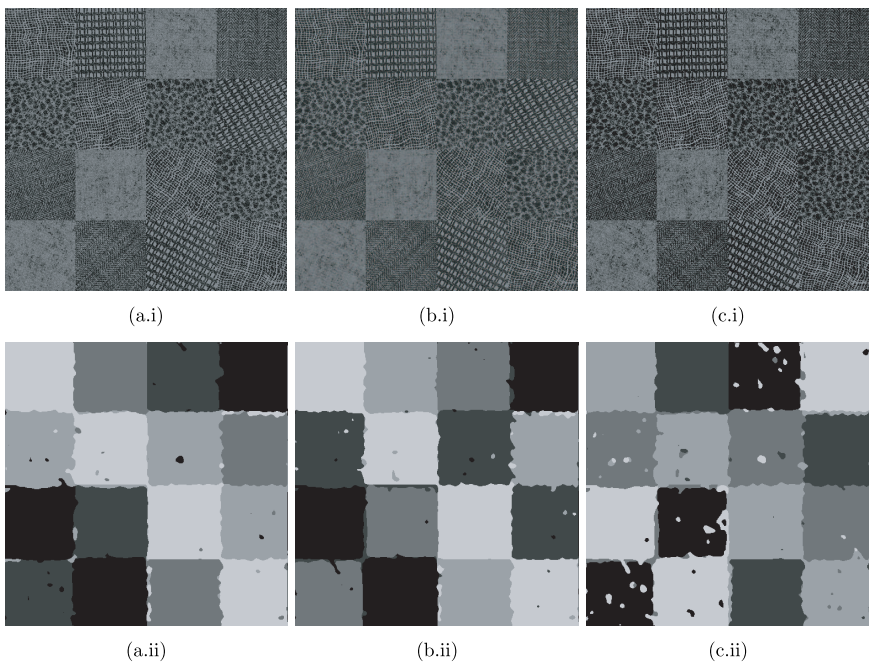


Fig. 12. Images corrupted with (a.i) Exponential; (b.i) Poisson; (c.i) Uniform noises.

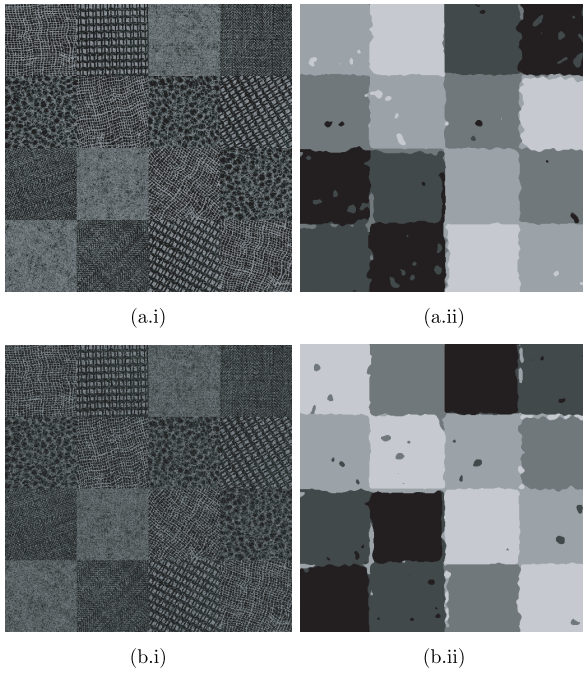


Fig. 13. Images corrupted with (a.i) Speckle; (b.i) Shot noises and their segmented output ( $k = 5$ ).

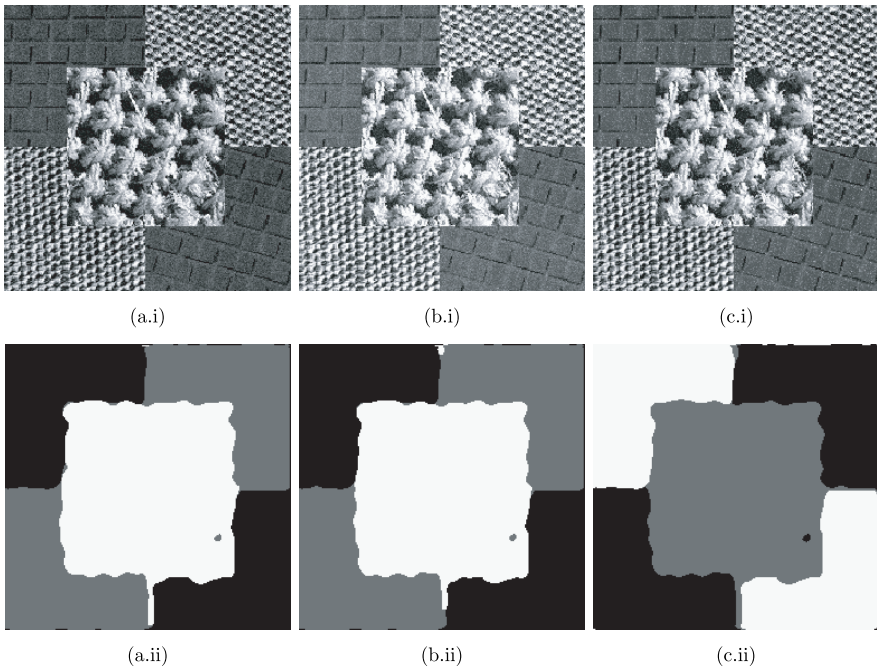


Fig. 14. Images corrupted with (a.i) Gaussian; (b.i) Rayleigh; (c.i) Exponential noises.

Table 6. User accuracy (U) and kappa values (K) in percentage corresponding to the different types of noise for data in Fig. 14.

Type of Noise	Parameter Attributes	Values	Effect	Overall		% Classification
				U	K	
Without noise	—	—	—	96.47	94.76	96.85
Gaussian	Mean	0	Additive	96.14	94.27	96.21
	Variance	200.0				
Rayleigh	Variance	200.0	Additive	96.19	94.36	96.08
Exponential	Variance	200.0	Additive	96.04	94.09	96.27

Fig. 17(a.i) which is same as Fig. 8(a.i). Figure 17(a.iii) is the corresponding using Zhang's method.<sup>15</sup> Similarly for other test image Fig. 17(b.i), the segmented outputs are Fig. 17(b.ii) and 17(b.iii) for the proposed and Zhang's methods respectively. The performance results are summarized in Table 8.

The objectives of the filtering and that of the local energy estimator, are to transform the edges between texture regions into detectable discontinuities. We have also studied the segmentation results particularly segmentation boundaries (strongest edges) between regions, using the Marr-Hildreth/Laplacian of Gaussian (LOG) operator as the filter, which is a zero-crossing detector as well as orientation independent. Figure 15(a.ii) shows the segmented output of the test figure Fig. 15(a.i) that we have already used before (Fig. 8), using LOG as the edge detector. Figure 15(b.ii) illustrates the performance of the LOG edge detector, with the test image having a gray-scale transform (Fig. 15(b.i)). The segmented output of Fig. 15(c.i), which is contaminated with gaussian noise is given in Fig. 15(c.ii). The performance results are summarized in Table 7. From the figures it is quite evident that our method gives a much better result over the LOG operator. Since the basic philosophy behind this work is to choose a filter which is circularly symmetric, we have proved the choice of any filter other than this will not give rotational invariance. For this we have demonstrated the segmentation on Fig. 8 using Canny's edge operator or Derivative of Gaussian (DOG). The segmentation result is shown in Fig. 16(b) and classification score of 59.23% is obtained, from which we can infer that Canny's edge detector cannot give rotational invariance for texture image.

## 5. Conclusion

This paper studies the issues of gray-scale transform and rotation invariance and immunity to noise, for texture segmentation. Simple and computationally efficient features have been extracted based on the fact that circularly symmetric wavelets give rotationally invariant features, as these wavelets are independent of orientation. We have seen that wavelet frame analysis is appropriate over the standard subsampled wavelet analysis for rotational invariance. We have also seen that the features that we have extracted are really invariant to rotation by studying the gradient

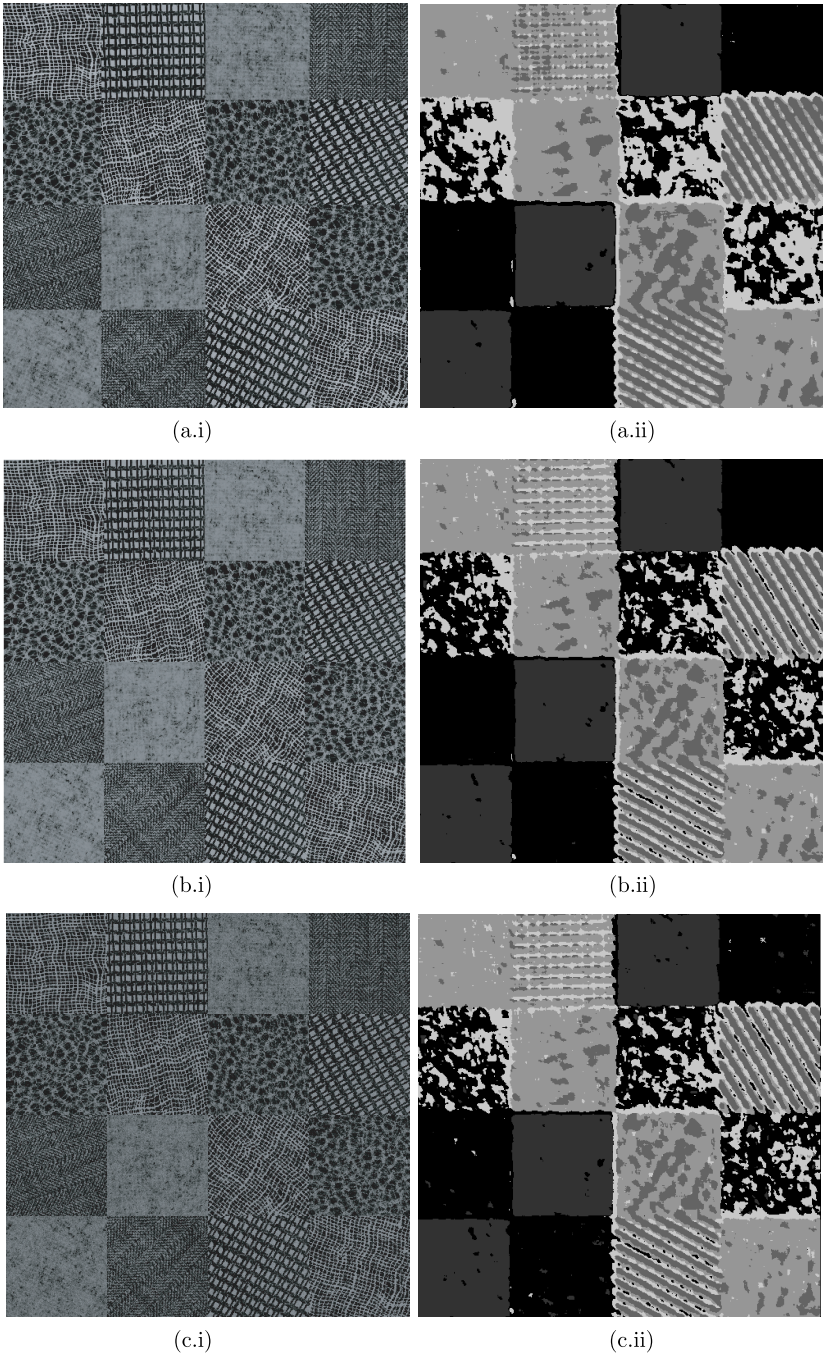


Fig. 15. (a.i) Test image of Fig. 8; (b.i) With gray-scale transform  $\alpha = 1.2$ ,  $\delta = 0$ ; (c.i) With gaussian noise (mean = 0.0, variance = 100.0); (a.ii)–(b.ii)–(c.ii) Corresponding segmented outputs ( $k = 5$ ) using Laplacian of Gaussian (LOG) as the zero crossing edge detector.

Table 7. User accuracy (U), kappa values (K) and classification in percentage for test image Fig. 8 using Marr–Hildreth operator as edge detector, results shown in Fig. 15.

Parameter	Values	Overall		% Classification
		U	K	
Without noise & gray-scale transform		72.99	66.24	71.22
With gray-scale transform	$\alpha = 1.2$ $\delta = 0$	67.68	59.63	66.56
With gaussian noise	mean = 0.0 variance = 100.0	67.60	59.61	66.57

Table 8. User accuracy (U), kappa values (K) and classification in percentage for two different test images including the Fig. 8(a.i).

Parameter	Test Figure	Overall		% Classification
		U	K	
Proposed	Fig. 17(a.i)	96.76	95.85	96.73
	Fig. 17(b.i)	96.06	97.09	98.56
Zhang’s method <sup>15</sup>	Fig. 17(a.i)	90.94	91.75	92.66
	Fig. 17(b.i)	93.96	94.09	94.56

magnitude histograms of the several rotated versions of the same texture class. The features are also invariant to gray-scale transformations and have inherent noise regularization. Yet another important aspect of our methodology is that, we make use of only five features to achieve the desired segmentation and this entails a huge reduction in feature dimensionality.

Studying the results of our experiment over several composite textured images we can conclude that our scheme performs appreciably well. But some edge inaccuracies are observed apart from some misclassification. One reason for this might be because, for the purpose of segmentation the features that are computed are pixel based, i.e. features are the measure of local energies over a small window around each pixel. A proper choice of this window size is very essential. The window size should be such that it captures the total periodicity of the micro textures of each class this calls for larger window size, where as for accurate texture boundary localization between classes the window size should be small. Experimentation with Gabor filter<sup>15</sup> and its comparison with the proposed technique, has shown that performance of the later one is found to be comparable. Although results due to Gabor filter is comparable, but a large number of parameter selection makes this method<sup>15</sup> computationally expensive.

Experimentation with other zero-crossing detector like the Marr–Hildreth (LOG) operator which is although isotropic, proved to be inferior to the proposed

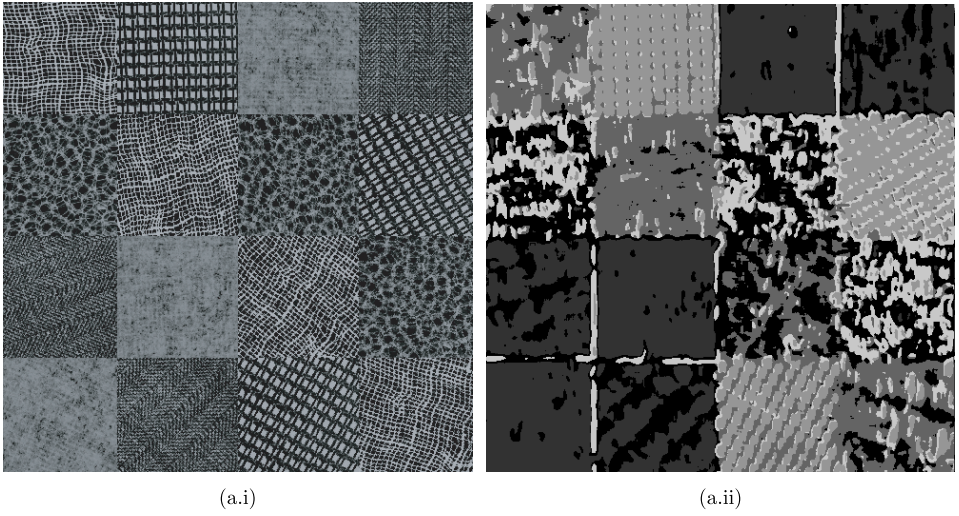


Fig. 16. (a.i) Test image of Fig. 8 (a.ii) corresponding segmented output ( $k = 5$ ) using Derivative of Gaussian (DOG) as the edge detector.

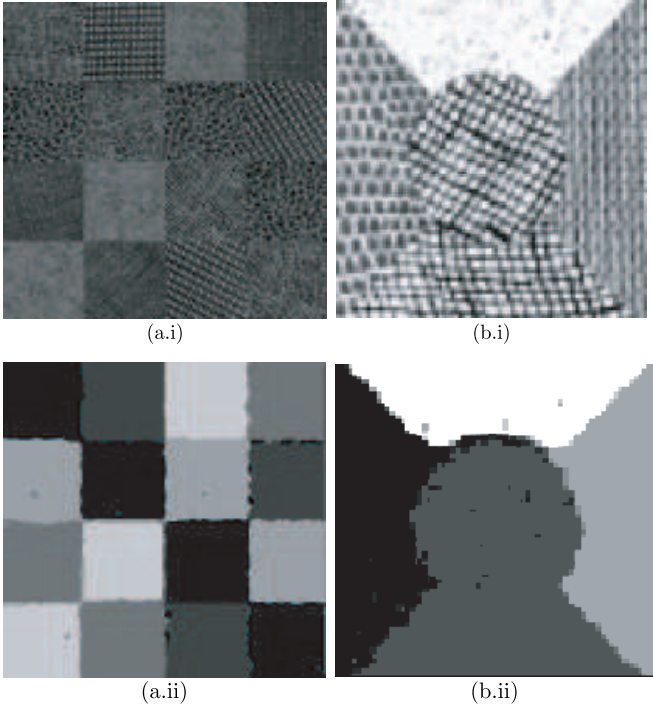


Fig. 17. (a.i)–(b.i) Test images and corresponding segmented outputs; (a.ii)–(b.ii) Using proposed method; (a.iii)–(b.iii) Using Gabor filters.<sup>15</sup>

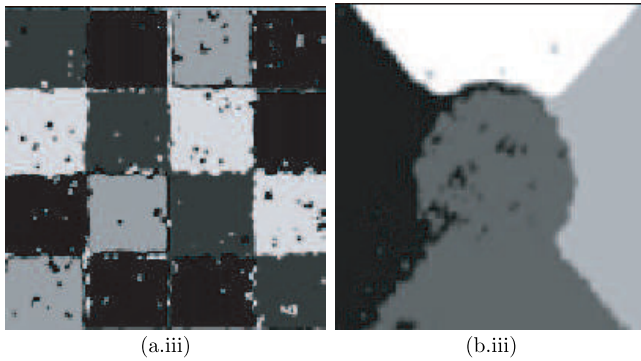


Fig. 17. (Continued)

Table 9. Moments of the magnitude histograms tabulated for comparing the similarity between the unrotated texture histograms and their rotated versions Fig. 4.

Texture Data	Rotation in Degrees	Order of Moments		
		First	Second	Third
Texture 1	unrotated	1.126	0.0014	0.0016
	20	1.127	0.0014	0.0016
	45	1.128	0.0014	0.0016
	55	1.129	0.0014	0.0016
	90	1.126	0.0014	0.0016
Texture 2	unrotated	1.288	0.0024	0.0029
	20	1.307	0.0023	0.0030
	45	1.313	0.0023	0.0031
	55	1.317	0.0024	0.0031
	90	1.291	0.0023	0.0029
Texture 3	unrotated	1.411	0.0029	0.0041
	20	1.444	0.0028	0.0039
	45	1.419	0.0029	0.0042
	55	1.432	0.0030	0.0043
	90	1.405	0.0029	0.0040
Texture 4	unrotated	1.334	0.0024	0.0032
	20	1.339	0.0024	0.0033
	45	1.336	0.0024	0.0032
	55	1.341	0.0024	0.0033
	90	1.331	0.0024	0.0032

technique for extraction of rotational invariant features, which in turn responsible for generating correct location of pixels on the segmentation boundaries between different regions Canny’s edge detector (DOG) is also failed to give rotational invariance as well.

## References

1. R. M. Haralick, K. Shanmugam and I. Dinstein, Texture feature for image classification, *IEEE Trans. Syst. Man Cybern.* **8**(6) (1973) 610–621.
2. P. C. Chen and T. Pavlidis, Segmentation by texture using correlation, *IEEE Trans. Pattern Anal. Mach. Intell.* **5** (1983) 64–69.
3. T. Chang and C. C. J. Kuo, Texture analysis and classification with tree structured wavelet transform, *IEEE Trans. Imag. Process.* **2**(4) (1993) 42–44.
4. J. L. Chen and A. Kundu, Rotation and gray scale transform invariant texture identification using wavelet decomposition and hidden markov model, *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(2) (1994) 208–214.
5. A. Laine and J. Fan, Texture classification by wavelet packet signatures, *IEEE Trans. Pattern Anal. Mach. Intell.* **15**(11) (1993) 1186–1190.
6. M. Unser, Texture classification and segmentation using wavelet frames, *IEEE Trans. Image Process.* **4**(11) (1995) 1549–1560.
7. R. L. Kashyap and A. Khotanzad, A model-based method for rotation invariant texture classification, *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(4) (1986) 472–481.
8. F. S. Cohen, Z. Fan and M. A. Patel, Classification of rotation and scaled textured images using gaussian markov random field models, *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(2) (1991) 192–202.
9. W. R. Wu and S. C. Wei, Rotation and gray scale transform invariant texture classification using spiral resampling, subband decomposition and hidden markov model, *IEEE Trans. Image Process.* **5**(10) (1996) 1423–1434.
10. G. M. Haley and B. S. Manjunath, Rotation-invariant texture classification using a complete space-frequency model, *IEEE Trans. Image Process.* **8**(2) (1999) 255–269.
11. S. R. Fountain and T. N. Tan, Efficient rotation invariant texture features for content based image retrieval, *Pattern Recogn.* **31**(11) (1999) 1725–1732.
12. M. Acharyya and M. K. Kundu, Robust texture classification using wavelet frames, *Image Process. Commun.: An Int. J.* **5**(2) (1999) 19–37.
13. M. Acharyya and M. K. Kundu, Efficient rotation invariant feature extraction for texture segmentation via multiscale wavelet frames, in *Proc. Indian Conf. on Computer Vision, Graphics and Image Processing (ICVGIP)*, Bangalore, India (2000), pp. 1–8.
14. J. Zhang and T. Tan, Brief review of invariant texture analysis methods, *Pattern Recogn.* **35** (2002) 735–747.
15. J. Zhang, T. Tan and L. Ma, Invariant texture segmentation via circular filter, *Proc Int. Conf. on Pattern Recognition (ICPR'02)*, Quebec, Canada (2002), pp. 901–904.
16. M. Acharyya and M. K. Kundu, Classification of rotated texture using overcomplete wavelet frames, *IETE J. Research* **46**(5) (2000) 363–370.
17. M. Chantler and G. McGunnigle, On the use of gradient space eigenvalues for rotation invariant texture classification, *Proc Int. Conf. Pattern Recognition (ICPR'00)*, Spain (2000), pp. 943–946.
18. Z. X. M. Pietikainen and T. Ojala, Rotation invariant texture classification using feature distribution, *Pattern Recogn.* **31** (2000) 975–985.
19. T. Ojala, Z. X. M. Pietikainen and T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary pattern, *IEEE Trans. Pattern Anal. Mach. Intell.* **24** (2002) 971–987.
20. M. K. Kundu and M. Acharyya, M-Band wavelet: Application to texture segmentation for real life image analysis, *Int. J. Wavelets Multiresolut. Inf. Process.* **1**(1) (2003) 115–149.

21. S. Mallat, A theory for multiresolution signal decomposition: The wavelet representation, *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(7) (1989) 674–693.
22. G. Strang and T. Nguyen, *Wavelets and Filter Banks* (Wellesley, Cambridge Press, 1996).
23. I. Daubechies, *Ten Lectures on Wavelets* (Soc. Ind. Applied Math, 1992).
24. S. Mallat, Zero-crossings of a wavelet transform, *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(4) (1993) 1019–1033.
25. S. Mallat and S. Zhong, Characterization of signals from multiscale edges, *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(7) (1992) 710–732.
26. J. R. Schriever and R. G. Congalton, Evaluating seasonal variability as an aid to cover-type mapping from landsat thematic mapper in the northeast, *Photogramm. Eng. Remote Sensing* **3** (1995) 321–327.
27. R. G. Congalton, R. G. Oderwald, and R. A. Mead, Assessing landsat classification accuracy using discrete multivariate analysis statistical techniques, *Photogramm. Eng. Remote Sensing* **49** (1983) 1671–1678.
28. G. H. Rosenfeld and K. Fitzpatrick-Lins, Coefficient of agreement as a measure of thematic classification accuracy, *Photogramm. Eng. Remote Sensing* **52** (1986) 223–227.
29. B. Uma Shankar, S. K. Meher and A. Ghosh, Neuro-wavelet classifier for multispectral remote sensing image, *Int. J. Wavelets Multiresolut. Inf. Process.* **5**(4) (2007) 589–611.