

RESEARCH

Open Access

Multiscale texture retrieval based on low-dimensional and rotation-invariant features of curvelet transform

Bulent Cavusoglu

Abstract

Multiscale-based texture retrieval algorithms use low-dimensional feature sets in general. However, they do not have as good retrieval performances as those of the state-of-the-art techniques in the literature. The main motivation of this study is to use low-dimensional multiscale features to provide comparable retrieval performances with the state-of-the-art techniques. The proposed features of this study are low-dimensional, robust against rotation, and have better performance than the earlier multiresolution-based algorithms and the state-of-the-art techniques with low-dimensional feature sets. They are obtained through curvelet transformation and have considerably small dimensions. The rotation invariance is provided by applying a novel principal orientation alignment based on cross energies of adjacent curvelet blocks. The curvelet block pair with the highest cross energy is marked as the principle orientation, and the rest of the blocks are cycle-shifted around the principle orientation. Two separate rotation-invariant feature vectors are proposed and evaluated in this study. The first feature vector has 84 elements and contains the mean and standard deviation of curvelet blocks at each angle together with a weighting factor based on the spatial support of the curvelet coefficients. The second feature vector has 840 elements and contains the kernel density estimation (KDE) of curvelet blocks at each angle. The first and the second feature vectors are used in the classification of textures based on nearest neighbor algorithm with Euclidian and Kullback-Leibler distance measures, respectively. The proposed method is evaluated on well-known databases such as, Brodatz, TC10, TC12-t184, and TC12-horizon of Outex, UIUCTex, and KTH-TIPS. The best performance is obtained for kernel density feature vector. Mean and standard deviation feature vector also provides similar performance and has less complexity due to its smaller feature dimension. The results are reported as both precision-recall curves and classification rates and compared with the existing state-of-the-art texture retrieval techniques. It is shown through several experiments that the proposed rotation-invariant feature vectors outperform earlier multiresolution-based ones and provide comparable performances with the rest of the literature even though they have considerably small dimensions.

Keywords: Texture retrieval; Low dimension; Multiresolution; Curvelet transform; Rotation invariance; Principle orientation

1. Introduction

Texture classification and retrieval has been investigated by many researchers. Recognizing textures is essential in content-based image retrieval (CBIR) applications since images are actually constructed of many texture combinations. Unfortunately, textures rarely exist in a fixed orientation and scale. Hence, defining rotation-invariant features is important and rotation invariance is a hot

research topic since 1980s. In one of the early works [1], rotation-invariant matched filters are used for rotation-invariant pattern recognition. The authors of [2] applied a model-based approach, in which they used statistical features of textures for classification. Using the statistics of spatial features as in [1,2] may provide good results, however, it may include great interclass variations depending on the recording conditions of textures such as contrast, illumination, etc. Hence, multiscale techniques which have the capability of representing the feature in one or more resolution with lesser effect of these recording conditions

Correspondence: bulent.cavusoglu@gmail.com
Electrical-Electronics Engineering Department, Ataturk University, Erzurum
25240, Turkey

have been used since 1990s. The main idea behind multiscale analysis in image processing is to provide the views of the same image in different resolutions to enhance the feature that can be more apparent in a specific resolution. In this way, it is easier to analyze or classify the image based on certain properties and certain scales. Nonstationary structures such as images require their multiscale transforms to be well localized both in time and frequency. However, according to Heisenberg's uncertainty principle, it is impossible to have localization both in time and frequency simultaneously. In other words, one cannot find a particular frequency to represent a certain point in time. Hence, frequency localizations require the time to be defined over a particular time window. It is also important that these localizations can be performed over orthogonal basis of tight frames. Wavelets [3] can address all these requirements. They are generated from one mother wavelet through translations and scalings. In one of the earliest works [4], the authors used statistics of Gabor wavelet as the features over Brodatz database while performing multiscale analysis for texture retrieval. However, the effects of rotations are not considered in this work. Another drawback of this work is such that wavelet transform is able to capture singularities around a point. The textures which have curvature-like structures may not provide good results by using the wavelet transform. Other transforms such as ridgelet [5] which extends wavelets to capture singularities along a line and curvelets [6,7] which can capture singularities around a curve are proposed to overcome such issues. One promising result of curvelet is that it can capture the edge around a curve in terms of very few coefficients. This creates new opportunities in the area of image processing. Curvelets with their nice features are also used in texture retrieval [8]. However, rotation invariance is not considered in [8]. Rotation invariance in the multiscale framework was first investigated in [9] for Gabor wavelet features. In a similar work, the authors used Gaussianized steerable pyramids for providing rotation-invariant features in [10]. Wavelet-based rotation invariance is introduced in [11] using rotated complex wavelet filters and in [12] using wavelet-based hidden Markov trees. These works show the effectiveness of their methods on the average performance. The details of their work also reveal that the textures with curvature-like structures perform worse than other textures. Hence, curvelet is a good alternative to overcome such issues. However, the authors of [8] realized that curvelet is actually very orientation-dependent and sensitive to rotation. Then, they provided rotation-invariant curvelet features in [13,14] based on comparison of energies of curvelet coefficients and realigning the curvelet blocks by cycle-shifting them with reference to the highest energy curvelet block. They showed that this scheme creates great advantage when compared to rotation-variant curvelet features. They

also showed that their features provide better results when compared to wavelets and rotation-invariant Gabor filters. However, the authors of [15] indicated that the provided method of [13,14] does not work for all the images, and they proposed another method based on modeling the curvelet coefficients as generalized Gaussian distributions (GGD) and then providing a distance measure by using Kullback-Leibler divergence between the statistical parameters of curvelets. It should be noted that they also use the highest energy curvelet block for circular shifting with the exception that they use only one reference point instead of using different reference points for each scale. This approach may provide good fits for higher scales of curvelet coefficients; however, lower levels of curvelet coefficients tend to not behave as Gaussian. In this study, we investigate the distributions of curvelet coefficients and use kernel density estimation (KDE) which provides better fits for lower scales as well. Although the complexity increases with density estimations, better results are obtained. There are also some latest and comprehensive works in texture retrieval trying to address both the scale invariance and rotation invariance issues. For instance in [16], Harris-Laplace detector [17] is used for salient region detection and then scale-invariant feature transformation (SIFT) [18] is used in order to provide scale invariance and rotation-invariant feature transformation (RIFT) [19] is used for rotation invariance. The results are pretty good; however, feature vector sizes are considerably large, 5,120 (40×128) for SIFT descriptor with earth mover distance (EMD). In [20], local binary pattern (LBP) variance is used for rotation invariance, in which two principle orientations are found and local binary pattern variances are used for texture retrieval. The feature dimensions of [20] with feature reduction are in the range of 1,000 s. In [21], both the scale and rotation variance are considered together using LBP, and it provides promising results again with feature sizes around LBP variants.

The main motivation of this study is to provide good retrieval performance with low-dimensional feature sets. The multiresolution structure in the literature has low-dimensional feature sets but not in the desired range of performances. In this study, we provide solutions for low-dimensional rotation-invariant multiresolution features with good retrieval performances by using curvelet transformation. First, a novel method is introduced for obtaining rotation-invariant curvelet features. The proposed method is based on cross energy principle. Second, the low-dimensional feature set based on mean and standard deviation of curvelet coefficients, used in the literature [13,14], is modified to reflect its support region. The size of this feature vector is 84, and the increase in the performance by this modification is also shown. Third, we use kernel density estimate, a nonparametric density estimation,

of curvelet coefficients to estimate the densities and use symmetric Kullback-Leibler distance as the distance measure. Although this feature set has higher dimension, 840, it provides better results and still remains in the low complexity region when compared with the other methods in the literature. It is shown through experiments that the results of the proposed feature sets are better than those of the state-of-the-art techniques in low dimension and comparable in medium dimension feature sets. The organization of the paper is as follows. First, multiresolution transforms are introduced in Section 2. Second, Section 3 explains the proposed texture retrieval scheme. Third, the proposed rotation invariance method is provided in Section 4, and classification is explained in Section 5. Fourth, the experimental results are presented in Section 6. Then, Section 7 includes discussions and comparisons with state-of-the-art texture retrieval techniques. Finally, Section 8 includes conclusions.

2. Background

Multiscale transforms are widely used in CBIR and texture retrieval. Hence, in order to better appreciate and understand the multiscale transforms, especially the curvelet transform, we briefly define wavelets, ridgelets, and curvelet transforms in this section.

2.1. Wavelets

Given that $\Psi_{s,\tau}(x, y)$ is a wavelet function for scale s and translation τ , wavelet transform of a function $f(x, y)$ and the inverse transform can be obtained by using Equations 1 and 2, respectively.

$$W_f(s, \tau) = \iint f(x, y) \Psi_{s,\tau}^*(x, y) dx dy \quad (1)$$

$$f(x, y) = \iint W_f(s, \tau) \Psi_{s,\tau}(x, y) d\tau ds \quad (2)$$

where Ψ is a two-dimensional mother wavelet. Other wavelets can be generated by scaling the mother wavelet function by s and shifting in the x or y direction by τ_x or τ_y , respectively, as given in Equation 3. In wavelet transform, only the transformation framework is outlined and the wavelet functions are left to the choice of the designer. Commonly used Mexican hat wavelet is depicted

in Figure 1. The isometric shape of the wavelet can be seen from the figure. The projection of the function of interest (i.e., an image) to this isometric wavelet results in capturing point singularities very well. However, the singularities in images are generally continuous around a line or a curve. In order to provide a better solution for the detection of line-shaped geometries, the ridgelets are proposed.

$$\Psi_{s,\tau_x,\tau_y}(x, y) = \frac{1}{\sqrt{s}} \Psi\left(\frac{x-\tau_x}{s}, \frac{y-\tau_y}{s}\right) \quad (3)$$

2.2. Ridgelets

Ridgelets are proposed for effectively describing anisotropic elements such as lines or curves with small number of coefficients. In order to have the ability of detecting lines or curves, it is necessary to define functions with directional geometry. Such a function is constant along lines of $x \cos(\theta) + y \sin(\theta)$. A sample ridgelet is given in Figure 2. The ridgelet is obtained by scaling and translating the mother wavelet function $\Psi(x, y)$. The ridgelet in Equation 4 is defined for angle θ , scale s , and translation τ . Ridgelets can be used to identify the singularities along lines.

$$\Psi_{s,\tau,\theta}(x, y) = \frac{1}{\sqrt{s}} \Psi\left(\frac{x \cos(\theta) + y \sin(\theta) - \tau}{s}\right) \quad (4)$$

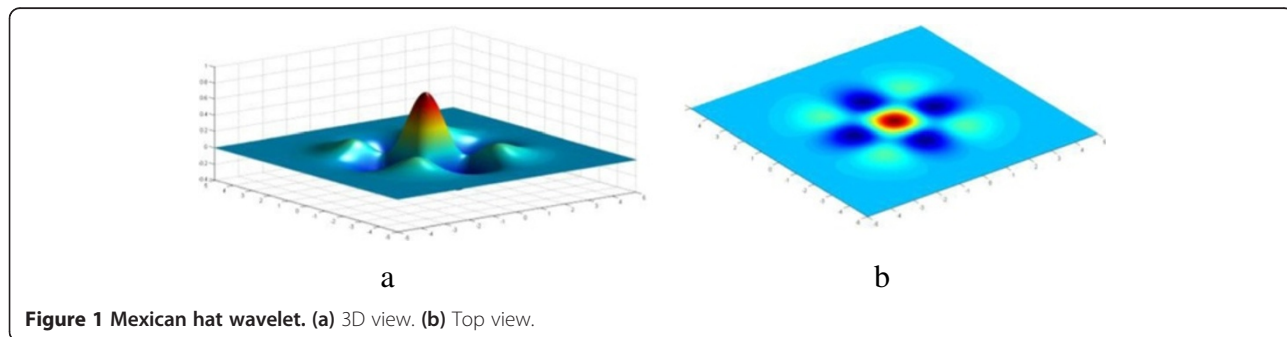
Using the ridgelet functions defined in Equations 4, the ridgelet transform and inverse ridgelet transform can be performed using Equations 5 and 6, respectively.

$$\mathcal{R} f(s, \tau, \theta) = \iint f(x, y) \Psi_{s,\tau,\theta}^*(x, y) dx dy \quad (5)$$

$$f(x, y) = \int_0^{2\pi} \int_{-\infty}^{\infty} \int_0^{\infty} \mathcal{R} f(s, \tau, \theta) \Psi_{s,\tau,\theta}(x, y) \frac{ds}{s^3} d\tau \frac{d\theta}{4\pi} \quad (6)$$

2.3. Curvelets

Curvelet transformation enables the detection of singularities along a curvature, while the ridgelets are not sufficient enough for the identification of curves due to



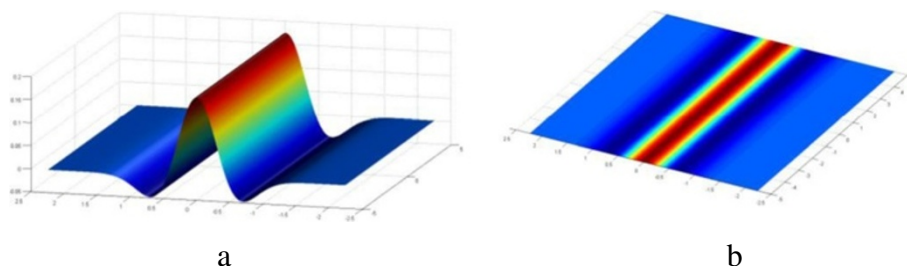


Figure 2 Mexican hat-based ridgelet. (a) 3D view. (b) Top view.

their line-directional geometry. Basically, a curvelet function is also a wavelet function which is rotated, scaled, and translated for different angles, scales, and shifts, respectively. A curvelet function can also be defined as a ridgelet function with various rotation angles. Figure 3 shows a curvelet function for specific scale, rotation, and translation. If the translations on \mathbf{Z}^2 are defined by $\mathbf{k} = (k_1, k_2)$, rotations are given by $\theta_\ell = 2\pi \cdot 2^{-s} \cdot \ell$ where $\ell = 0, 1, \dots, 2^s$ such that $0 \leq \theta_\ell < 2\pi$, parabolic scaling matrix D_s is given by Equation 7 and rotation operator is given by Equation 8, then the curvelet function is defined by Equation 9.

$$D_s = \begin{bmatrix} 2^{2s} & 0 \\ 0 & 2^s \end{bmatrix} \quad (7)$$

$$R_{\theta_\ell} = \begin{bmatrix} \cos\theta_\ell & \sin\theta_\ell \\ -\sin\theta_\ell & \cos\theta_\ell \end{bmatrix} \quad (8)$$

$$\Psi_{s,\ell,\mathbf{k}}(x, y) = 2^{3s/2} \Psi_s \left(D_s R_{\theta_\ell} \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \right) \quad (9)$$

where Ψ_s is a mother wavelet function. Based on the above definitions, the curvelet coefficient is given by Equation 10.

$$C(s, \ell, \mathbf{k}) = \iint f(x, y) \Psi_{s,\ell,\mathbf{k}}^*(x, y) dx dy \quad (10)$$

A graphical explanation of the curvelet can be depicted as in Figure 4. Here, the image is represented by a red curve over which the curvelet transform is calculated, and the blue line in black ovals represents the cross-sectional magnitude of curvelet operator. The dot product of the line, originally the image, and the curvelet function becomes maximum when the image and the signal are aligned, in other words, have the maximum number of common points (pixels). On the other end, the curvelet coefficients become zero if the two do not cross each other for any rotational and/or translational change. Hence, it is possible to follow the orientation and location of the image, red line, by just determining the maximum of curvelet coefficients. Due to this efficient property, it is possible to use curvelets for edge detection, object detection,

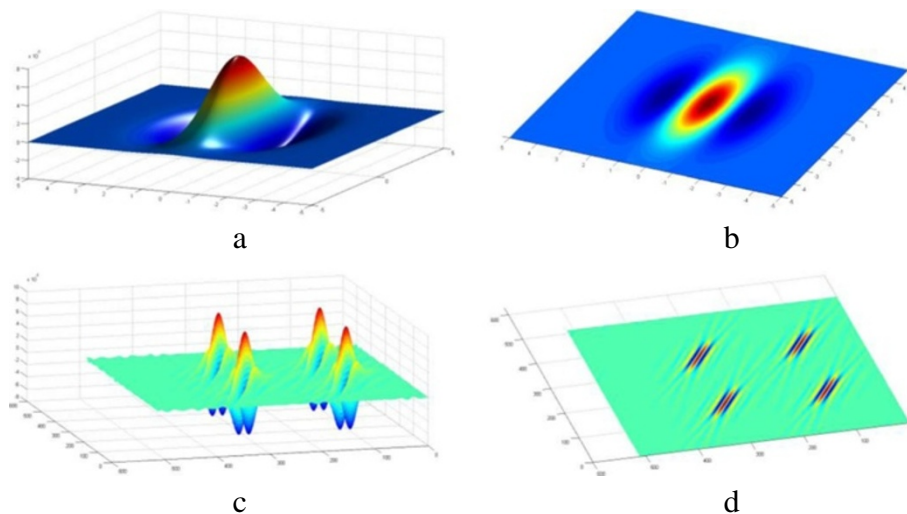


Figure 3 A curvelet function for specific scale, rotation, and translation. (a) 3D view of a Mexican hat based curvelet. (b) Mexican hat curvelet, top view. (c) 3D view of a Meyer-based curvelet. (d) Meyer curvelet, top view.

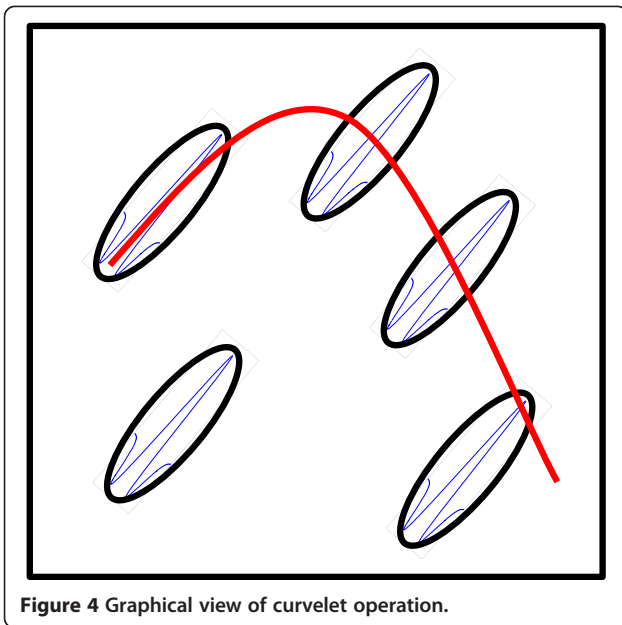


Figure 4 Graphical view of curvelet operation.

noise removal, texture identification, etc. Since orientation is an important feature of curvelet transformation, curvelet coefficients may significantly vary with rotation. Hence, the direct use of curvelet coefficients as the image features introduce rotation dependency and overall texture classification performance may deteriorate if rotated replica of a texture exists in the database. So, it is necessary to utilize curvelet coefficients in a rotation-invariant manner to overcome this downside.

3. Proposed texture retrieval scheme

The proposed texture retrieval scheme is depicted in Figure 5. In the proposed scheme, first, the query and training images are selected from the image database. Second, curvelet transform is applied to both sets of images. Third, principle orientation (PO) of each image is detected by analyzing the cross energies of the curvelet coefficients. Then, the extracted features are realigned by cycle-shifting all the features around the PO. Finally, PO-aligned features are compared for classification. Each step of the algorithm is explained in the following subsections.

3.1. Feature extraction

Broad range of feature sets are used in the literature such as entropy, energy, first- and second-order statistics, and many more. In this study, we propose and evaluate two different feature vectors. The first one is called as mean and standard deviation feature vector, $F_{\mu\sigma}$ and the second one is called as kernel density feature vector, F_{KDE} . $F_{\mu\sigma}$ includes the mean and standard deviation of curvelet coefficients, which belong to different levels and angles, scaled with a support coefficient. Similar features previously used in [13,14] without a scaling factor. Using only the first- and second-order statistics may describe the distribution fully only if the distribution is Gaussian. However, as indicated in earlier works [15], the Gaussian probability density function (PDF) may not be a perfect fit for curvelet data. Moreover, the curvelet coefficients at lower levels deviate from the Gaussian distribution as it can be seen from Figure 6, which presents second level curvelet coefficients

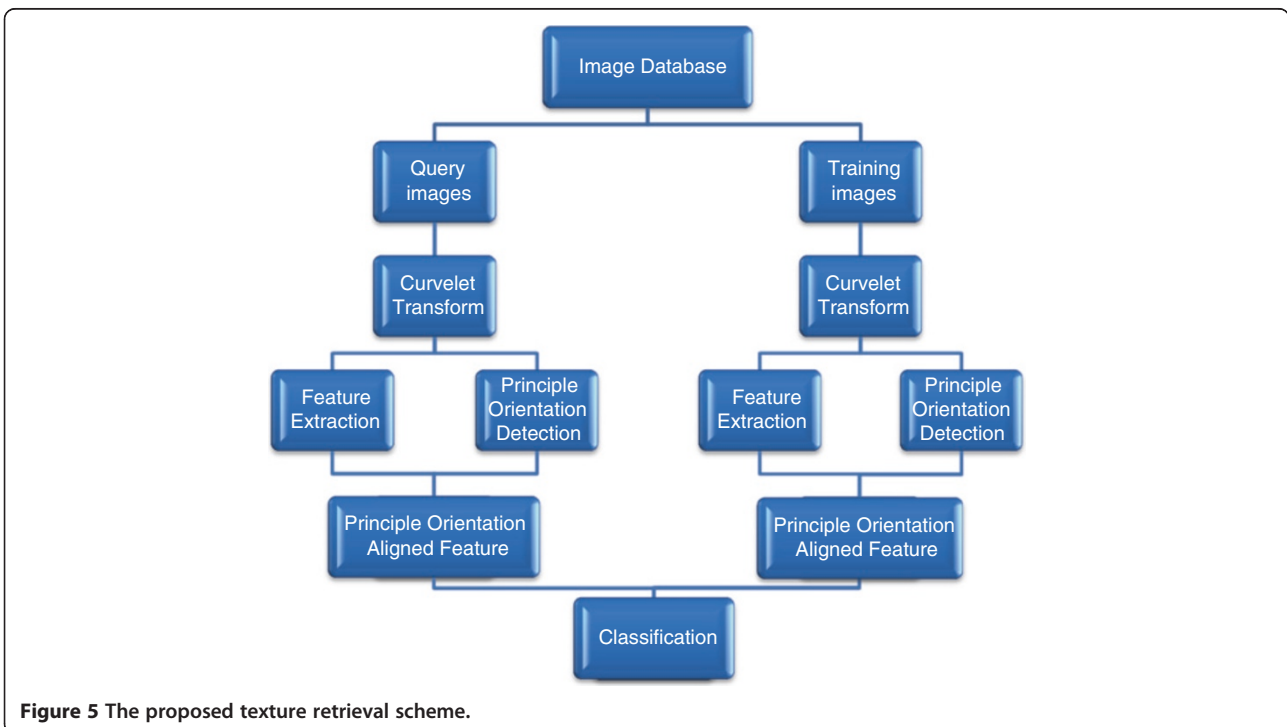
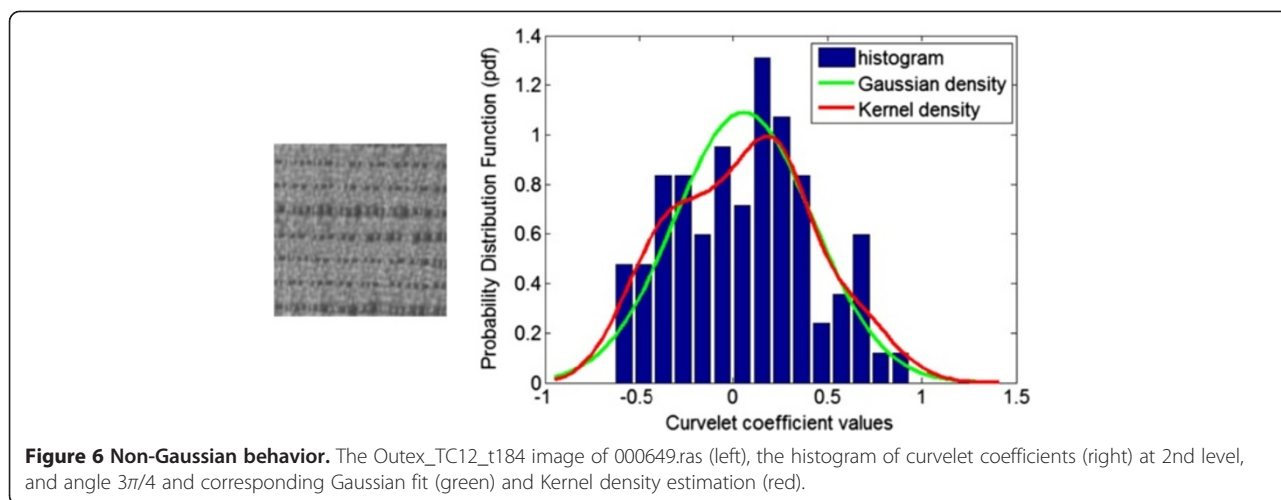


Figure 5 The proposed texture retrieval scheme.



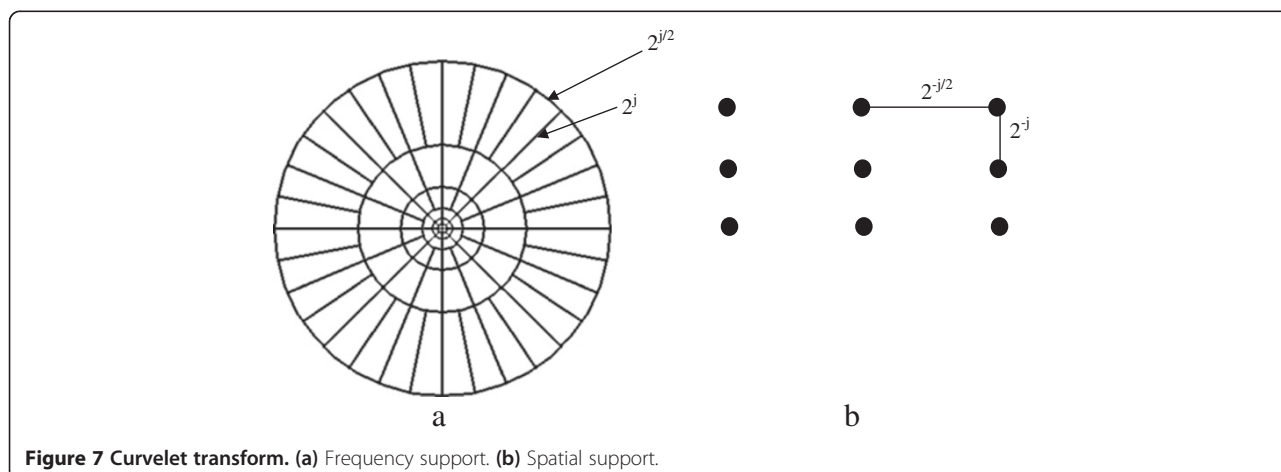
of an image. Hence, kernel density feature, F_{KDE} , which estimates the PDF of curvelet coefficients using KDE, is also proposed. It is expected to obtain better classification results when the PDF of curvelet coefficients is used since it represents full statistics. An alignment step is needed in both approaches to provide rotation invariance. Before going into the details of the alignment step, the feature vectors of this study are defined first.

3.2. Mean standard deviation feature vector $F_{\mu\sigma}$

A feature vector which includes the first- and second-order statistics of curvelet coefficients for five levels is given by Equation 11.

$$F_{\mu\sigma} = [\mu_{1,1}, \sigma_{1,1}, \mu_{2,1}, \sigma_{2,1}, \mu_{2,2}, \sigma_{2,2}, \dots, \mu_{2,8}, \sigma_{2,8}, \mu_{3,1}, \sigma_{3,1}, \mu_{3,2}, \sigma_{3,2}, \dots, \mu_{3,16}, \sigma_{3,16}, \mu_{4,1}, \sigma_{4,1}, \mu_{4,2}, \sigma_{4,2}, \dots, \mu_{4,16}, \sigma_{4,16}, \mu_{5,1}, \sigma_{5,1}] \quad (11)$$

where $\mu_{s,\ell}$ and $\sigma_{s,\ell}$ are the mean and standard deviation of curvelet coefficients at scale s and angle ℓ , respectively. It should be noted that it is enough to consider only the first half plane of the curvelet coefficients since curvelet transform is even symmetric around π . This feature vector is depicted for 5 scales and includes 84 elements. The feature vector of Equation 11 is used in [13] as well. Since the feature vector includes robust features such as the first- and second-order statistics, it can be used for comparison purposes. As it can be seen from Figure 7, the number of wedges doubles every other scale going from the lower to the higher frequencies. This means that the spatial support is halved every other scale as well. In other words, curvelet transformation is applied over a narrower region going from the lower to the higher scales. A larger spatial support region means that it is more likely to have dissimilarities. Thus, the statistics carried out from dissimilarities should be penalized. A similar approach is also used in [22], where the authors use spatially obtained



features for classification of various scene categories. In order to reflect the size of the spatial support, we apply a weighting factor, α_s , given by Equation 12 and obtain the scaled mean-standard deviation feature vector, $\mathbf{F}_{\mu\sigma}$ given by Equation 13.

$$\alpha_s = \begin{cases} 1 & , s = 1 \\ 2^{\text{ceil}(s/2)} & , s > 1 \end{cases} \quad (12)$$

$$\mathbf{F}_{\mu\sigma} = [\alpha_1(\mu_{1,1}, \sigma_{1,1}), \alpha_2(\mu_{2,1}, \sigma_{2,1}, \mu_{2,2}, \sigma_{2,2}, \dots, \mu_{2,8}, \sigma_{2,8}), \dots, \alpha_N(\mu_{N,1}, \sigma_{N,1})] \quad (13)$$

where N is the total number of scales and s is the scale. The 'ceil' function rounds up the number to the nearest integer. If there are five scales, then the corresponding feature vector is given by the following:

$$\mathbf{F}_{\mu\sigma} = [2^0(\mu_{1,1}, \sigma_{1,1}), 2^1(\mu_{2,1}, \sigma_{2,1}, \mu_{2,2}, \sigma_{2,2}, \dots, \mu_{2,8}, \sigma_{2,8}), 2^2(\mu_{3,1}, \sigma_{3,1}, \mu_{3,2}, \sigma_{3,2}, \dots, \mu_{3,16}, \sigma_{3,16}), 2^2(\mu_{4,1}, \sigma_{4,1}, \mu_{4,2}, \sigma_{4,2}, \dots, \mu_{4,16}, \sigma_{4,16}), 2^3(\mu_{5,1}, \sigma_{5,1})] \quad (14)$$

The images we use are either 128×128 or converted to 128×128 in the preprocessing stage during our work, and the feature vector used in this study has five scales. Considering 8 angles at 2nd, 16 angles at 3rd and 4th, and 1 for 1st and 5th scales, the size of the feature vector is $(1 + 8 + 16 + 16 + 1) \times 2 = 84$.

3.3. Kernel density feature vector \mathbf{F}_{KDE}

Probability density of curvelet coefficients is very close to normal distribution. However, earlier works have showed that the coefficients may not exactly be modeled by using a normal PDF. It is shown in [15] that modeling curvelet coefficients by GGD provides a better fit than that of the normal PDF. In this study, we use a nonparametric approach for estimating the density of curvelet coefficients due to the fact the Gaussianity assumption gets even weaker for lower levels. One may notice non-Gaussian behavior by observing Figure 6. Nonparametric estimation is widely used when parametric modeling of the distribution becomes infeasible. We obtain the proposed kernel density feature vector, \mathbf{F}_{KDE} , through KDE. It is given by Equation 15.

$$\mathbf{F}_{KDE} = [f_{1,1}, f_{2,1}, f_{2,2}, \dots, f_{2,8}, f_{3,1}, f_{3,2}, \dots, f_{3,16}, f_{4,1}, f_{4,2}, \dots, f_{4,16}, f_{5,1}] \quad (15)$$

where each element of \mathbf{F}_{KDE} , which represents the density of curvelet coefficients at a particular scale and angle, is estimated through KDE. The feature vector of Equation 15 is given for five scales and can be extended to include higher number of scales. In KDE, first, a kernel function is

defined [23]. Then, using n data points (X_1, X_2, \dots, X_n) of a random variable x , the kernel estimator for PDF $p(x)$ is given by Equation 16:

$$\hat{p}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x-X_i}{h}\right) \quad (16)$$

where K is the kernel function and h is the smoothing parameter called bandwidth. The kernel function used in this study is normal kernel with zero mean and unity variance. Each kernel is placed on the data points and normalized over the data to obtain the kernel estimation. A more depth analysis on KDE is given in [23]. The histogram of the curvelet coefficients, corresponding Gaussian fit, and KDE is shown in Figure 6. As it can be seen from the figure, KDE provides much better fit than Gaussian. The non-Gaussian structure of curvelet coefficients can be observed for second-level coefficients of a sample image given in Figure 6. We have evaluated the kernel density at 20 bins, resulting in a feature vector dimension of 840 (42×20).

4. Rotation invariance

4.1. Effect of rotation on curvelet transform

Following the curvelet transformation, curvelet coefficients for different orientations and specific scales are obtained. Hence, the curvelet coefficients reflect the effect of the rotation. Let us consider a particular scale s with rotation angles represented by $\{\theta_1, \theta_2, \dots, \theta_n\}$. For each rotation angle, there exists a curvelet coefficient matrix. The elements of this matrix are obtained following a translation in x and y direction. Curvelet transformation of two different images and their rotated versions are given in Figure 8. These images are in the size of 128×128 and have 5 scales in curvelet domain. Four of those scales are shown in Figure 8. The fifth scale is the highest resolution and is not divided into angles. The most inner box and the most outer box represent the lowest and highest resolutions, respectively. We can follow that the rotation is captured in all scales. It is difficult to notice the rotation by just looking at the curvelet domain image. However, high energy areas are really noticeable. The authors of [13,14] realized this feature and proposed to synchronize them by aligning the highest energy curvelet coefficients while cycle-shifting the others not to change the relative order among all. Since the curvelet coefficients are arranged in a cyclic fashion, applying this idea gave promising results. However, the obvious energy compaction is not valid for all images as the authors of [15] pointed out. It is also possible that the high energy area may exist at some other location in the rotated image after curvelet transformation is applied, especially in the figures where a nice uniform texture does not exist.

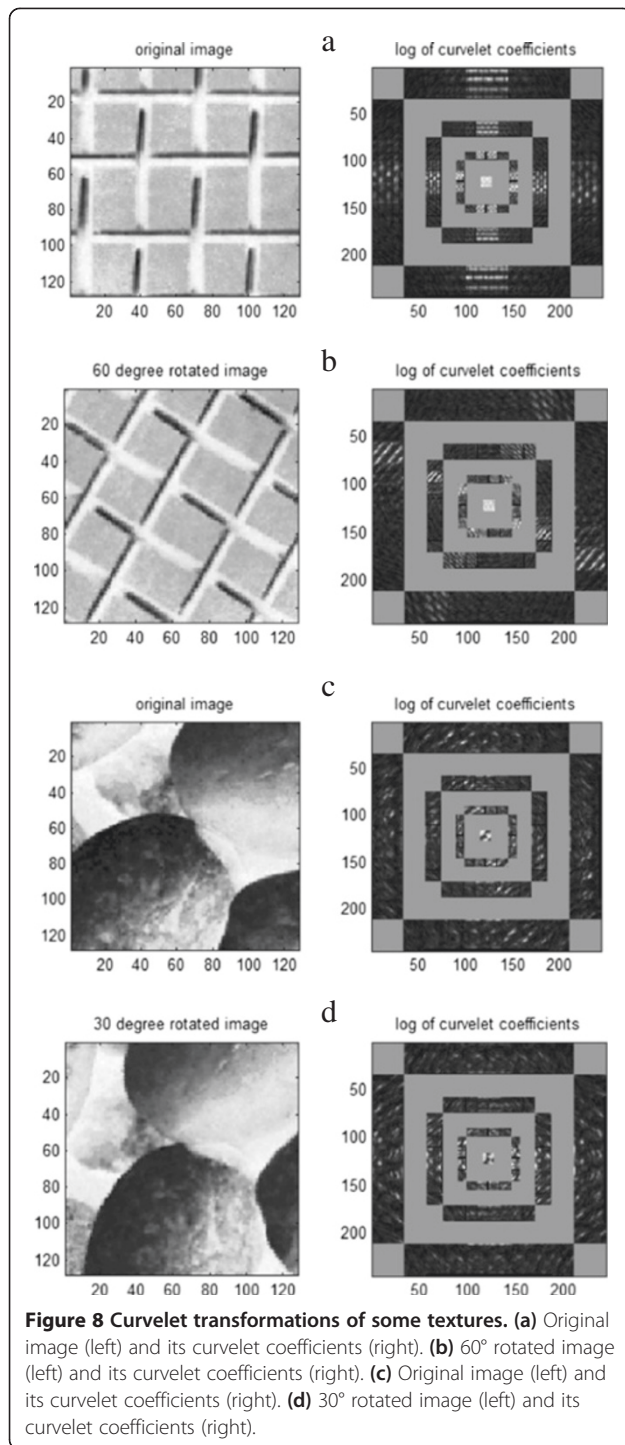


Figure 8 Curvelet transformations of some textures. (a) Original image (left) and its curvelet coefficients (right). (b) 60° rotated image (left) and its curvelet coefficients (right). (c) Original image (left) and its curvelet coefficients (right). (d) 30° rotated image (left) and its curvelet coefficients (right).

This nonuniformity can be observed in Figure 8c,d. In order to overcome this issue, first, we propose to find the most robust area of the image against rotation based on curvelet transform and mark that point as principle orientation; then perform an alignment by cycle-shifting the feature vector with reference to principle orientation. In order to find the least affected rotation angle, we

perform cross-correlation check for two adjacent curvelet coefficients at each scale.

4.2. Principle orientation detection

In order to minimize the effect of rotation in the texture, it is necessary to find a reference point, namely, principle orientation, so that all feature vectors can be synchronized by reordering the features. The rotation dependence is expected to be eliminated after the synchronization. The authors of [13,14] suggest a synchronization routine by means of the curvelet block with the maximum energy. We propose to use cross energy of adjacent curvelet blocks for the principle orientation detection, and the procedure is explained in the following subsection.

4.3. Cross-correlation and cross energy of curvelet coefficients at adjacent angles

The cross-correlation of two adjacent curvelet blocks for angles ℓ and $\ell + 1$ is given as follows:

$$R_{s,\ell}(n_1, n_2) = \sum_{k_1} \sum_{k_2} \left(\left| C(s, \ell, k_1, k_2) \right| \cdot \left| C(s, \ell + 1, k_1 + n_1, k_2 + n_2) \right| \right) \quad (17)$$

The cross-correlation function actually reflects the cross energies for different lags. In obtaining the latter curvelet coefficient on the right hand side of Equation 17, only a rotation is applied to curvelet operator while the image stands still. Also, as it can be seen from Equation 9 that this rotation operator is not supposed to cause a lag in the latter coefficient. Hence, it is expected to get the maximum value of cross-correlation function at 0th lag, that is $R_{s,\ell}(0, 0)$. As a result, Equation 17 can be used to detect the highest cross-energy blocks. Another view can be expressed as follows: by analyzing the adjacent blocks of curvelet transform in terms of their cross-correlation quantities, one may find the orientation for each scale which is the least affected by rotation. In other words, getting a high correlation between two adjacent blocks means that the directional change has little effect on curvelet coefficients for the specific two orientations at hand. In short, if curvelet coefficients of two adjacent blocks of an image at specific orientation give the highest values, they will also be the ones with the highest correlation values for the rotated version of original texture. The proposed method is structured based on this approach. Since rotation of curvelet operator and rotation of image has the same effect, the observed angle between the curvelet operator and the image for the highest correlation value remains fixed. Based on this principle, we determine the fixed angle by searching for the highest cross correlation and take the first of the

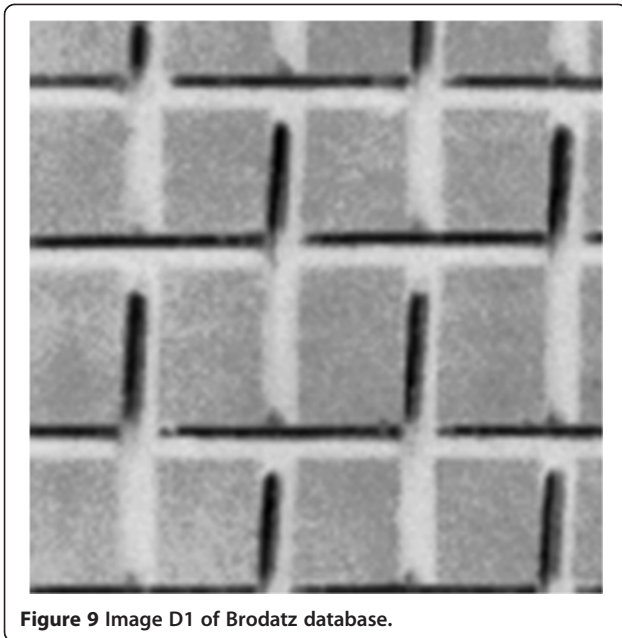


Figure 9 Image D1 of Brodatz database.

highest cross-energy (correlated) blocks as the principle block (orientation) and then cycle-shift all the coefficients in reference to the principle orientation. Hence, this operation provides an alignment based on the highest cross-energy principle. Once the cross-correlation functions are obtained for all scales except the coarsest and finest due to the fact that there is only one coefficient matrix for them, the curvelet coefficients are aligned with reference to the highest 0th lag value of cross-correlations in each scale. The dimension mismatch is generally the case faced for two coefficient matrices of adjacent orientations. If there are not enough coefficients to match the larger sized coefficient block, then the smaller sized coefficient block is padded with zero coefficients in order to overcome the dimension mismatch problem. This zero-filling solves the dimension mismatch problem and does not affect the cross energy.

4.3. Closer look on principle orientation alignment based on cross energy

In this subsection, we outline some examples to better understand the contribution of this study. In the first example, we consider an image taken from the Brodatz database as shown in Figure 9. The corresponding curvelet coefficients of this image and its 30° and 60° rotated versions are given in Figure 10. The yellow boxes on each scale show the principle orientations obtained by the proposed algorithm. Similarly, Figure 11 shows the same curvelet transforms with yellow boxes representing the reference points based on the algorithm of [13]. A close look immediately reveals that both algorithms have common reference points. But it can also be observed that the proposed algorithm captures the boxes where orientation at each scale is the same, whereas the algorithm of [13] may not detect the correct orientation at the scale 2 for this particular example. This is due to the fact that the texture of this figure does not have a uniform pattern, and rotation may cause the curvelet transform to capture the most dominant edges for that orientation. Since the proposed algorithm focuses on the amount of change in the rotation, it manages to capture the correct orientation at each scale.

In the second example, we consider the image '000480.ras' of Outex TC12_t184 database and its rotated image of '000649.ras'. The images and their corresponding kernel density estimations are given in Figure 12. As can be observed from the figure, coefficients of right column are cycle-shifted around the highest cross-energy coefficient block, second from the top and highlighted by a bold frame. As a result, this coefficient block, level = 2 and angular parameter = 2, is reordered (cycle-shifted) in a way that this set gets angular parameter value of 1 (the one at the top of the middle column) and all the others move into the position of prior angular parameter in a cyclic manner.

It should also be noted that the curvelet coefficients of unrotated and rotated images show some differences

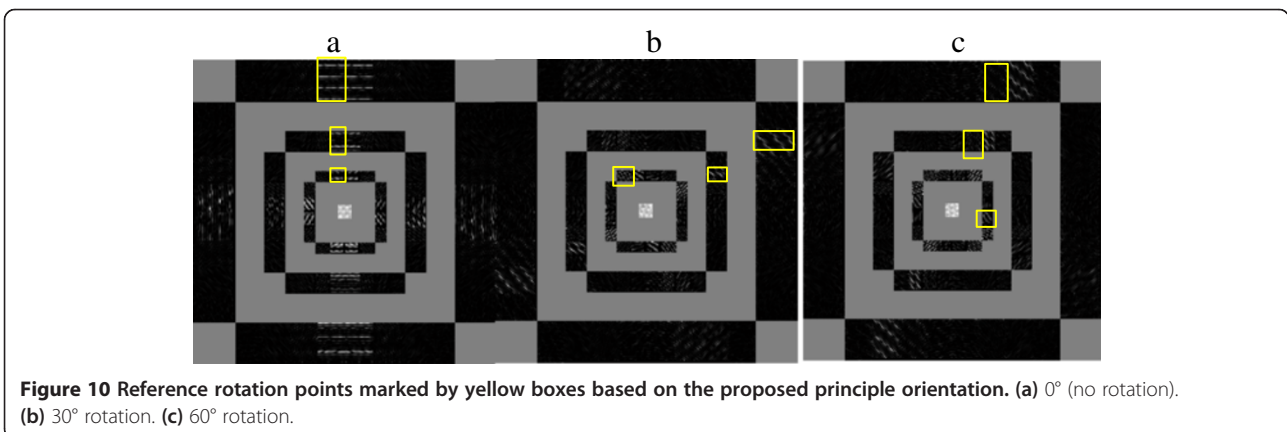


Figure 10 Reference rotation points marked by yellow boxes based on the proposed principle orientation. (a) 0° (no rotation). (b) 30° rotation. (c) 60° rotation.

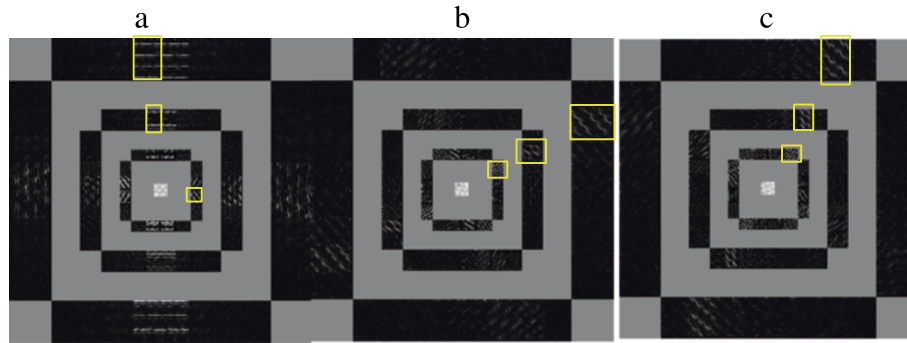


Figure 11 Reference rotation points marked by yellow boxes based on the rotation invariance of [13]. (a) 0° (no rotation). (b) 30° rotation. (c) 60° rotation.

even after principle orientation alignment. This can also be observed by comparing the first and second columns of Figure 12. This is due to the fact that the curvelet coefficients of these images may be similar; however, it is hardly likely that they will be the same. Hence, the purpose of the alignment is to make the curvelet coefficients of two images comparable as much as possible.

4.5. PO-aligned feature vectors

The mean standard deviation, $F_{\mu\sigma}$ and kernel density, F_{KDE} , feature vectors are aligned according to principle orientation, following the principle orientation detection. The aligned feature vectors are cycle-shifted versions of the initial ones. The PO-aligned mean-standard deviation feature vector and kernel density feature vector are denoted as $F_{\mu\sigma}^{PO}$ and F_{KDE}^{PO} , respectively. The rotation-invariant mean-standard deviation feature vector without scaling, $F_{\mu\sigma}'^{PO}$, is also used in our simulations for comparison purposes. The proposed PO-aligned feature vectors are used in the classification process in this study.

5. Classification

The classification is performed based on nearest neighbor (NN) classifier. In NN, the query image is compared against the training images of all the classes and the image is assigned to the class which has the minimum distance with. Separate distance measures are used in this study for each proposed feature vector. Euclidian distance is used with the mean and standard deviation feature vector and Kullback-Leibler distance measure with kernel density feature vector.

5.1. Distance measures

Euclidian distance

The PO-aligned feature vectors of training and query images are compared to find the best match based on Euclidian distance measure. The Euclidian distance, d_{ij}^{euc} ,

between the i th query image and the j th database image is calculated by Equation 18.

$$\left(d_{ij}^{euc}\right)^2 = \left(F_{i,\mu\sigma}^{PO} - F_{j,\mu\sigma}^{PO}\right) \left(F_{i,\mu\sigma}^{PO} - F_{j,\mu\sigma}^{PO}\right)^T, i \neq j \quad (18)$$

where $F_{i,\mu\sigma}^{PO}$ and $F_{j,\mu\sigma}^{PO}$ are the feature vector of query image (i th image of the database) and the training image (the feature vector of the j th database image), respectively.

Symmetric Kullback-Leibler distance

Kullback-Leibler divergence is a common method to measure the distance between two PDFs and is given by Equation 19:

$$d_{pp}^{KL} = \int p(x) \ln \left(\frac{p(x)}{p'(x)} \right) dx \quad (19)$$

Since d_{pp}^{KL} is not necessarily equal to $d_{pp'}^{KL}$, it is more appropriate to use symmetric Kullback-Leibler (SKL) distance, given by Equation 20;

$$d_{pp'}^{SKL} = \left| \frac{1}{2} \int p(x) \ln \left(\frac{p(x)}{p'(x)} \right) dx + \frac{1}{2} \int p'(x) \ln \left(\frac{p'(x)}{p(x)} \right) dx \right| \quad (20)$$

The SKL distance between the kernel density feature vectors of query image, $F_{i,KDE}^{PO}$, and the training images, $F_{j,KDE}^{PO}$ is then given by Equation 21, in which n is the dimension of the feature vector.

$$d_{F_{i,KDE}^{PO}, F_{j,KDE}^{PO}}^{SKL} = \sum_{m=1}^n d_{F_{i,KDE}^{PO}(m), F_{j,KDE}^{PO}(m)}^{SKL}, i \neq j \quad (21)$$

6. Experimental results

The proposed algorithm is evaluated over various databases, Brodatz [24], Outex TC10 [25], Outex TC12-horizon [25], Outex TC12-t184 [25], KTH-TIPS [26], and UIUCTex [19]. The setup for each database is as follows: 100 simulations are run for each database, and

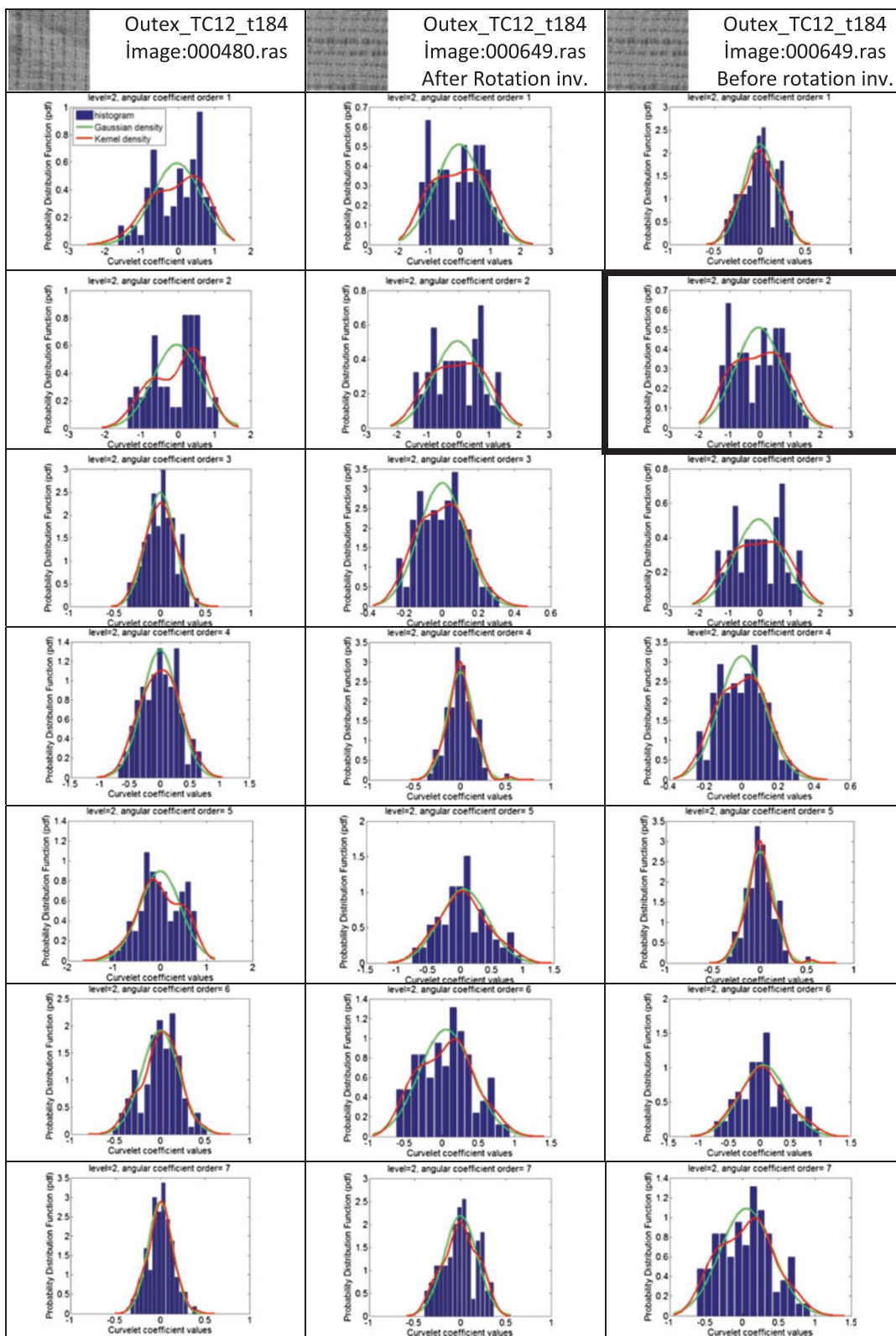


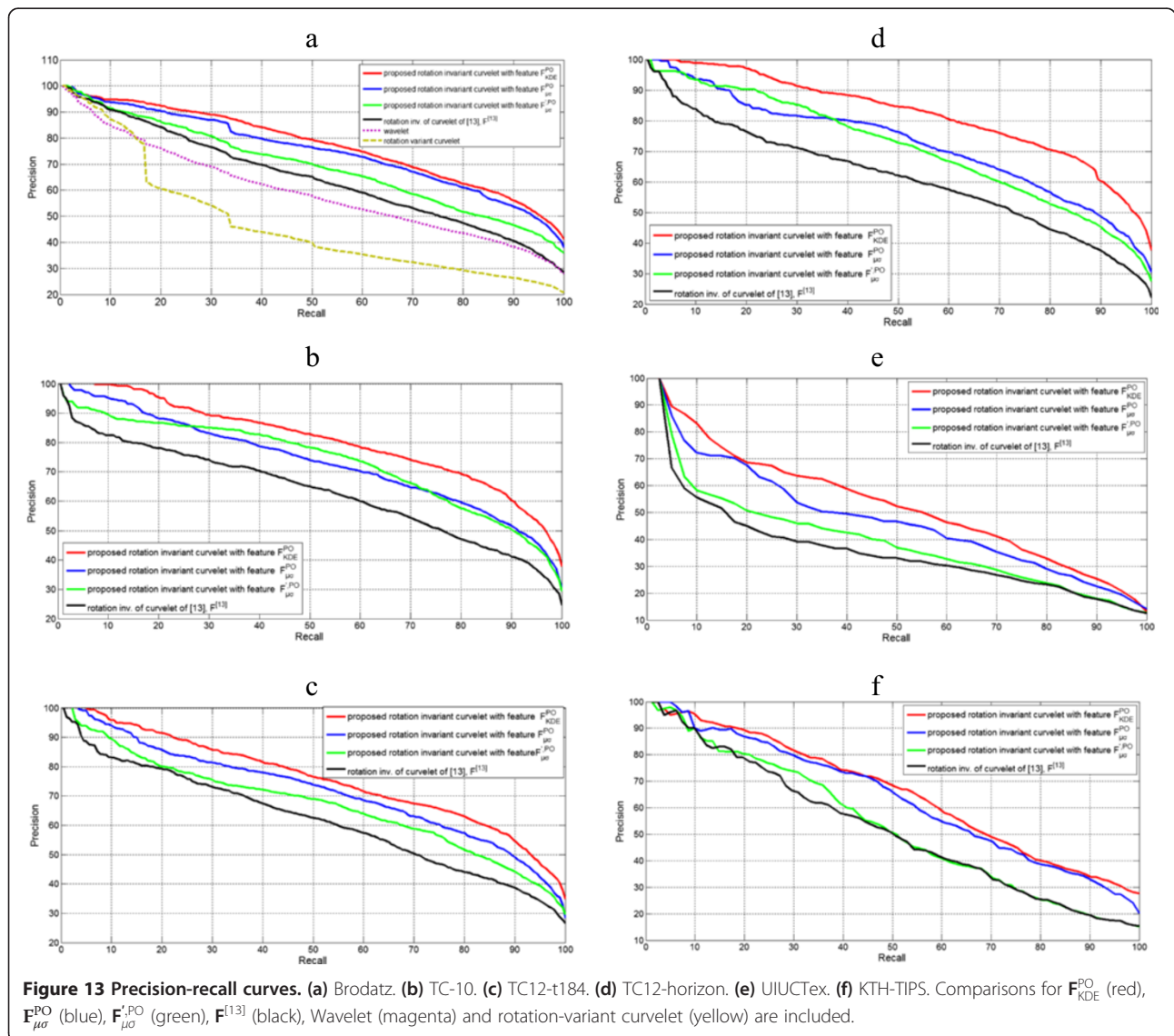
Figure 12 The effect of the proposed rotation invariance. Unrotated figure (left column), rotated figure (middle column) with PO alignment, and rotated figure without PO alignment (right column).

average precision-recall and classification performances are reported for all the simulation setups.

- (a) Training images: They are selected randomly from each class of each database. Number of training images is varied from 10 to 70 in increments of 10 s. The results are reported separately for various numbers of training images.
- (b) Query images: Training images are excluded from the database, and the remaining images are used as queries. The average classification and precision-recall results are reported.
- (c) Brodatz database: The database is proposed in [24] and includes 112 classes, each with 192 images. In order to create large enough database with translations and rotations, first nonrotated test images are created by dividing each original $512 \times$

512 image into 16 nonoverlapping 128×128 regions; then, 12 rotated test images are obtained for multiple of 30° rotations. The reason for 30° rotations is to obtain results, comparable with [24] which uses the same database with the same setup. A database of 21,504 images ($112 \times 16 \times 12$) is constructed in this way. In this setup, each class includes 192 images.

- (d) Outex TC10 database: The database is proposed in [25] and includes 24 classes each with 180 images. The images are recorded under incandescent (inca) illumination. Each class consists of 20 non-overlapping portions of the same texture with 9 different orientations (0, 5, 10, 15, 30, 45, 60, 75, 90). The database includes a total of 4,320 images ($24 \times 20 \times 9$).
- (e) Outex TC12-horizon database: The database is proposed in [25] and includes 24 classes and 180



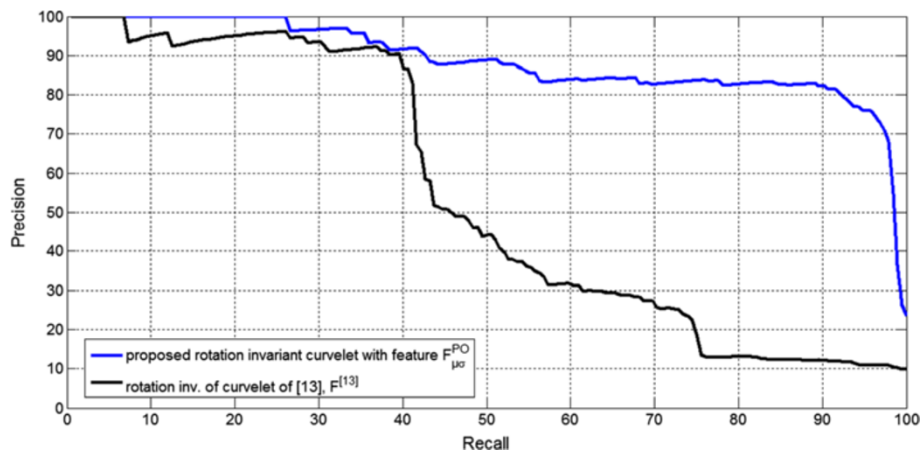


Figure 14 Precision-recall curve for D1 image of the Brodatz database.

images for each class. The same setup of Outex TC10 database is used except that the images are recorded under horizon (horizon sunlight) illumination.

- (f) Outex TC12-t184 database: The database is proposed in [25] and includes 24 classes and 180 images for each class. Same setup is used as Outex TC10 database except that the images are recorded under t184 (fluorescent 184) illumination.
- (g) KTH-TIPS database: The database is proposed in [26] and includes 10 classes and 81 images for each class. The images are recorded under varying illumination, pose, and scale. The database includes total of 810 images (10×81).
- (h) UIUCTex database: The database is proposed in [19] and includes 25 classes and 40 images for each class. The images include significant scale and viewpoint variations as well as rotations. The database includes a total of 1,000 images (25×40).

The experimental results are reported under two main performance measurement categories, precision-recall curves and classification accuracies. The studies in the

literature make use of both performance measures. In order to make our work easily comparable with future works as well as the literature, we have provided our results under these two categories. In order to see only the effect of principle orientation alignment and performance of two feature vectors of this study, the results of the proposed methods are compared generally with only one reference from the literature. The results of [13] are used for general comparison purposes with our results since the authors of [13] also use curvelet features. We make a broader comparison with the literature in the discussion section.

6.1. Precision-recall curves

Precision is the ratio of number of relevant retrieved images to number of all retrieved images whereas recall is the ratio of number of relevant retrieved images over total number of relevant images in the database. The precision-recall curves for all the databases are provided in Figure 13. Figure 13a compares the performances of the proposed rotation-invariant F_{KDE}^{PO} and $F_{\mu\sigma}^{PO}$ features with the feature $F_{\mu\sigma}^{\prime,PO}$ where scaling is not used, the

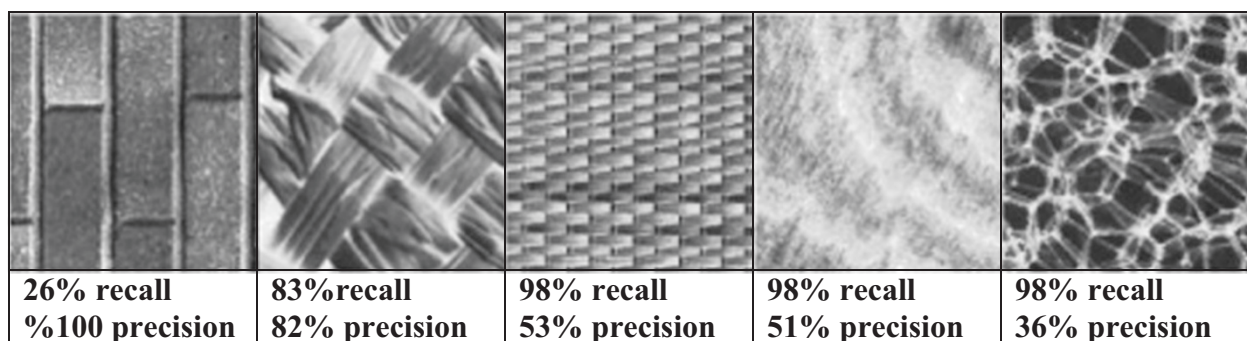


Figure 15 Mixed classes for the query image of D1 when rotation-invariant $F_{\mu\sigma}^{PO}$ is used.

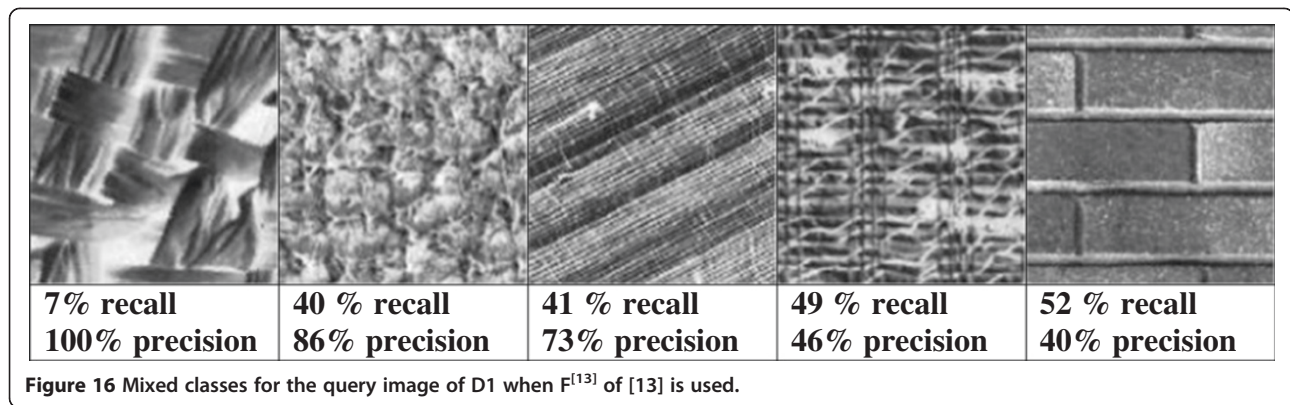


Table 1 Classification rates (%)

	Classification	TC10	TC12-horizon	TC12-t184	KTH-TIPS	Brodatz	UIUCTex
Train 10	F_{KDE}^{PO}	91.94	92.06	90.24	80.70	95.39	65.63
	$F_{\mu\sigma}^{PO}$	91.15	91.48	88.37	75.41	92.73	65.21
	$F_{\mu\sigma}^{\prime,PO}$	89.47	89.39	85.78	69.21	90.86	52.10
	$F^{[13]}$	82.97	82.48	78.23	68.11	82.27	46.82
Train 20	F_{KDE}^{PO}	94.87	95.66	93.59	86.69	97.24	71.41
	$F_{\mu\sigma}^{PO}$	94.80	95.15	93.32	82.56	95.81	70.36
	$F_{\mu\sigma}^{\prime,PO}$	92.83	92.81	90.22	76.36	94.23	57.95
	$F^{[13]}$	88.72	87.94	84.78	76.20	88.69	51.73
Train 30	F_{KDE}^{PO}	96.60	96.64	95.25	88.31	98.02	74.08
	$F_{\mu\sigma}^{PO}$	96.12	96.50	95.06	87.92	96.88	73.79
	$F_{\mu\sigma}^{\prime,PO}$	94.62	94.46	91.91	79.57	95.72	61.81
	$F^{[13]}$	90.77	90.99	87.39	77.37	91.08	53.95
Train 40	F_{KDE}^{PO}	97.53	97.53	95.94	90.83	98.47	
	$F_{\mu\sigma}^{PO}$	96.86	97.49	95.86	89.42	97.70	
	$F_{\mu\sigma}^{\prime,PO}$	95.24	95.41	92.85	83.27	96.71	
	$F^{[13]}$	92.58	92.36	88.89	82.15	93.04	
Train 50	F_{KDE}^{PO}	97.82	98.05	96.62	92.19	98.91	
	$F_{\mu\sigma}^{PO}$	97.60	97.82	96.46	91.10	98.05	
	$F_{\mu\sigma}^{\prime,PO}$	95.90	95.99	93.74	84.32	97.53	
	$F^{[13]}$	93.30	93.20	90.23	82.77	94.01	
Train 60	F_{KDE}^{PO}	98.00	98.40	97.20	92.84	99.04	
	$F_{\mu\sigma}^{PO}$	97.73	98.22	96.92	91.52	98.37	
	$F_{\mu\sigma}^{\prime,PO}$	96.16	96.40	94.54	84.00	97.64	
	$F^{[13]}$	94.10	93.74	91.03	83.14	94.76	
Train 70	F_{KDE}^{PO}	98.33	98.46	97.44	93.09	99.18	
	$F_{\mu\sigma}^{PO}$	98.22	98.29	97.28	92.36	98.72	
	$F_{\mu\sigma}^{\prime,PO}$	96.52	96.44	94.86	86.00	98.10	
	$F^{[13]}$	94.40	94.63	91.79	85.64	95.34	

All the databases are evaluated for various (10,20,...,70) number of training images except the UIUCTex database since it only includes 40 images per class.

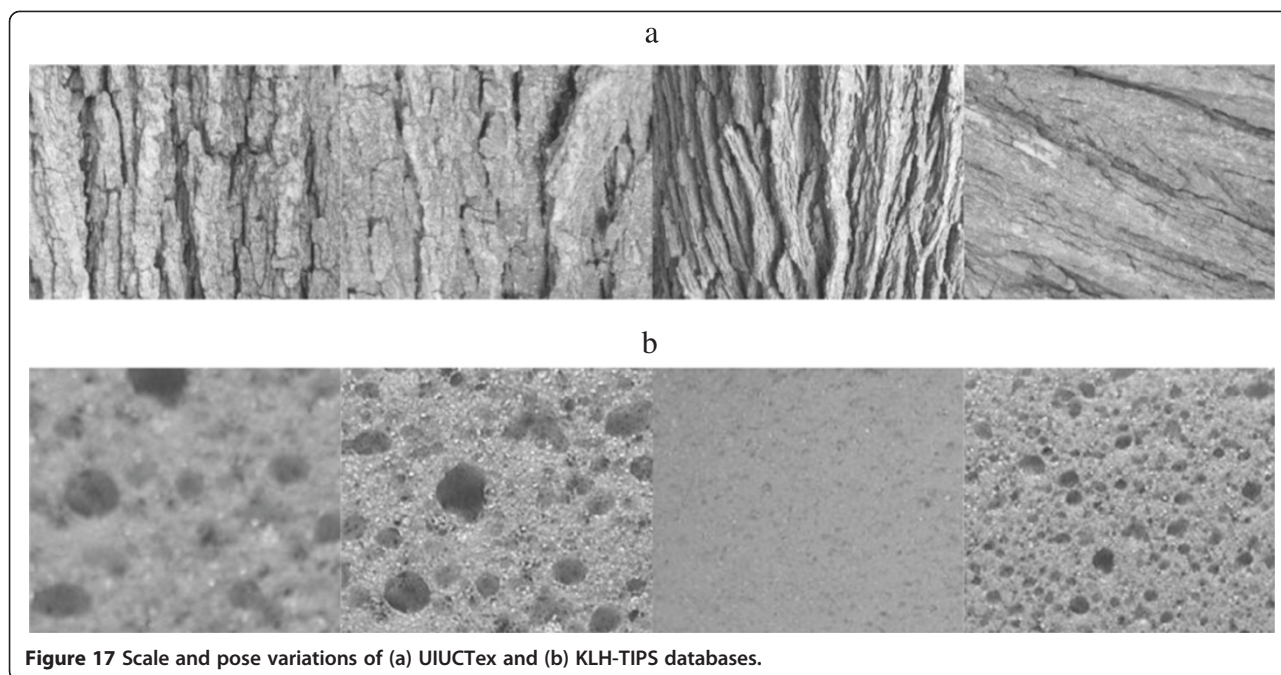


Figure 17 Scale and pose variations of (a) UIUCTex and (b) KTH-TIPS databases.

algorithm of [13] represented by \mathbf{F} [13], wavelet, and rotation-variant features of curvelet in Brodatz database. Since the algorithm of [13] is already better than Gabor and ridgelet transforms and shown in detail in the literature, they are not included in this figure. As can be seen from this figure, the performance of $\mathbf{F}_{\text{KDE}}^{\text{PO}}$ is better than that of the other methods. It should be kept in mind that using $\mathbf{F}_{\text{KDE}}^{\text{PO}}$ instead of $\mathbf{F}_{\mu\sigma}^{\text{PO}}$ increases the complexity due to the increased feature size. Hence, the better performance against $\mathbf{F}_{\mu\sigma}^{\text{PO}}$ comes in the expense of complexity. The results for Outex TC10, TC12-t184, and TC12-horizon are given in Figure 13b,c,d, respectively. It can be observed from these figures that the proposed algorithm with the feature vector $\mathbf{F}_{\text{KDE}}^{\text{PO}}$ provides the best results followed by the feature vector $\mathbf{F}_{\mu\sigma}^{\text{PO}}$. Although the same performance order is preserved for the results of UIUCTex and KTH-TIPS, given in Figure 13e,f, respectively, a lower precision-recall performance is observed compared to that of Outex database. The reason for that both UIUCTex and KTH-TIPS databases include scale and viewpoint variations and the proposed algorithm does not perform as well

under viewpoint and scale variations as it does for rotation variations.

We now provide a more depth analysis based on the precision-recall curve for a particular image taken from Brodatz database. Figure 14 shows the precision-recall curve of D1 query image of Brodatz database given in Figure 9. As it can be followed from Figure 14, the proposed feature vector $\mathbf{F}_{\mu\sigma}^{\text{PO}}$ with rotation invariance provides much better precision-recall curve on this particular image. Figure 15 includes intermediate results and gives the mixed classes that are not relevant with the query image and the point where they are included in the precision-recall curve. Figure 15 shows that the first irrelevant image comes at the 26% recall and 100% precision point. It means that 50 relevant images (192×0.26) are retrieved before an irrelevant image is retrieved. This break point can also be seen on the blue line in Figure 14. Similarly, Figure 16 provides intermediate results for the algorithm of [13]. The first irrelevant image is retrieved at 7% recall and 100%. It means that 13 relevant images (192×0.07) are retrieved before an irrelevant one is retrieved.

Table 2 Comparison of classification rates with KLD of [15] for number of training images of 60 and 70

Classification	Train 60		Train 70	
	KTH-TIPS	KTH-TIPS	KTH-TIPS	KTH-TIPS
$\mathbf{F}_{\text{KDE}}^{\text{PO}}$	92.84	98.40	98.40	98.40
$\mathbf{F}_{\mu\sigma}^{\text{PO}}$	91.52	98.22	98.22	98.22
KLD of [15]	83.60	86.90	86.90	86.90

6.2. Classification rates

In this section, classification rates are provided. If the query image is classified to its own class, then this classification is marked as true, if not, then it is marked as false. The percentage of correctly marked ones gives the classification rate. The training images are selected randomly from each class, and then, the remaining images are used as queries to get the classification rate. This

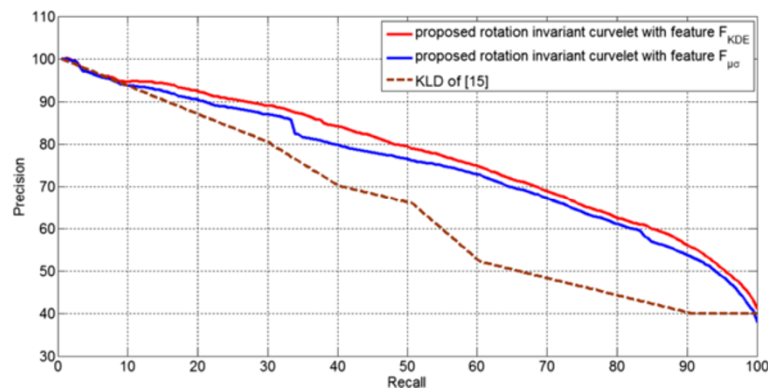


Figure 18 Precision-recall comparison with KLD of [15] for Brodatz database.

process is repeated 100 times, and the average results are reported in Table 1 where classification rates of the proposed feature vectors F_{KDE}^{PO} and $F_{\mu\sigma}^{PO}$ and nonscaled feature vectors $F_{\mu\sigma}^{\prime,PO}$ and $F^{[13]}$ of [13] are included. As can be seen from the table, F_{KDE}^{PO} has the superior performance followed by $F_{\mu\sigma}^{PO}$. Brodatz database provides the highest performance in terms of classification since it only contains the rotated replica of the cropped images. Outex database provides the next best results followed by TC12-horizon, TC10, and TC12-t184 with slight differences. The differences among the subclasses of Outex database are not much, and overall performance for this database is good since it also includes only rotations and does not have scale or viewpoint variations. The UIUCTex and KTH-TIPS databases are the ones with the worst results among all. This is due to the fact that both databases include scale and pose variations as can be seen from Figure 17. The proposed feature vectors perform well for these databases as well, as can be seen from Table 1.

Table 3 Comparison of classification rates with LBP variants of [20] for training number of 20

Classification	Feature dimension	TC10	TC12-horizon	TC12-t184
F_{KDE}^{PO}	840	94.87	95.66	93.59
$F_{\mu\sigma}^{PO}$	84	94.80	95.15	93.32
F_{KDE}^{PO} out of class	840	94.87	87.35	86.25
$F_{\mu\sigma}^{PO}$ out of class	84	94.80	86.73	85.51
$LBP_{24,3}^{ru2}/VAR_{24,3}$ [20]	416	98.15	87.03	87.15
$LBP_{8,1}^{ru2}/VAR_{8,1}$ [20]	160	96.66	77.98	79.25
$LBP_{24,3}^{u2}_{GMES}$ [20]	13251	97.76	95.57	95.39
$LBP_{8,1}^{u2}_{GMES}$ [20]	451	73.64	76.57	72.47
$LBPV_{24,3}^{u2}_{GMPD2}$ [20]	2211	97.55	94.18	94.23
$LBPV_{8,1}^{u2}_{GMPD2}$ [20]	227	72.99	76.15	72.19

7. Discussion

In this section, a broader comparison with the most recent and successful works in the literature is provided. The proposed rotation-invariant texture retrieval algorithm is evaluated by using the proposed PO-aligned feature vectors F_{KDE}^{PO} and $F_{\mu\sigma}^{PO}$ and observed that they really perform well even though the feature dimensions are considerably low compared to those of the literature. In [15], following an energy-based cycle shift based on only one level, GGD estimations of curvelet coefficients are used with Kullback-Leibler distance (KLD) measure. Although the size of the feature dimensions is not elaborated in [15], we presume that it is close to size of F_{KDE}^{PO} which is 840 in our case. As can be seen from Table 2, both proposed methods outperform KLD in KTH-TIPS database. The precision recall curve is also provided in Figure 18 for comparison in the Brodatz database. The superior performance of the proposed methods over KLD can be observed from this figure.

In [20], LBP variance features provide really promising results. We compare our results with the results of [20] in Table 3. The classification results for F_{KDE}^{PO} and $F_{\mu\sigma}^{PO}$ reflect in-class training. That is, the training images and the query images belong to the same class. However, the authors of [20] used in-class training for TC10 while they use out of class training for TC12-horizon and TC12-t184 for which they choose 20 images from TC10 database and use it for the queries of the other databases. Hence, we have run the simulations for these settings as well. ' F_{KDE}^{PO} out of class' and ' $F_{\mu\sigma}^{PO}$ out of class' reflect the results of these simulations. As can be seen from Table 3, variants of LBP with low feature sizes perform worse than the proposed algorithm, but for high feature sizes, they outperform our algorithm especially in out of class classifications. The main reason for this outcome is that our algorithm is computationally efficient with its small feature size, and good results of LBP come in the expense of increased computational complexity.

Table 4 Comparison of classification rates with [16] for indicated training numbers

Classification	Feature size	UIUCTex	KTH-TIPS	Brodatz
F_{KDE}^{PO}	840	71.41	90.83	92.00
$F_{\mu\sigma}^{PO}$	84	70.36	89.42	90.78
HSR + LSR SIFT [16]	$40 \times 128 = 5,120$	98.00	92.70	94.00
HSR + LSR RIFT [16]	$40 \times 100 = 4,000$	96.00	86.70	89.60

Number of training images for UIUCTex, 20; KTH-TIPS, 40; Brodatz, 3.

The authors of [16] use Laplace and Harris detectors for salient region detection. SIFT is also used for scale invariance, and RIFT is used for rotation invariance. Although the results are good, feature vector dimensions are considerably large, 5,120 (40×128) for SIFT descriptor with EMD. It should also be noted that support vector machine (SVM) classification, a strong classifier requiring learning effort, is used in [16]. Since we are not using SVM, we are not exactly able to tell how much of the better performance is obtained due to SVM. It is worth noting that using rotation-invariant technique RIFT and decreasing the feature size in their work also cause decrease in performance, and this effect can be seen from HSR + LSR of RIFT [16] in Table 4 where our proposed algorithm has better performance in KTH-TIPS and Brodatz databases.

Table 5 is included for easy comparison with the literature in terms of computational load and performance. The proposed algorithms, especially mean and standard deviation feature vector, have small feature dimensions. This is important as execution of the distance calculation at each comparison is proportional to the feature size. The computational complexity based on feature sizes are depicted as low, medium, and high in Table 5,

and the table is arranged in an increasing complexity manner. That is, top rows have lower complexity and bottom rows have higher complexity. Since the computational complexity of SVM is much higher than that of NN, SVM-based algorithms are placed at the bottom of Table 5. The proposed $F_{\mu\sigma}^{PO}$ feature vector has quite low dimension, 84, and it provides really good results. The other proposed vector F_{KDE}^{PO} also provides good results with 840 feature dimension.

Table 5 gives a comparison of the proposed algorithm with the rest of the literature in terms of performance, dimension, and complexity in related databases. The algorithms in the top three rows are based on curvelet transformation, which are shown to outperform earlier multiscale-based texture classification methods. It is clear from the table that although they have similar feature sizes and complexities with the proposed algorithms, the proposed algorithms outperform all of them. The variants of LBP proposed in [20] are given in the table as well. It is seen that for dimension size of 160, $LBP_{8,1}^{riu2}/VAR_{8,1}$ algorithm provides worse results than the proposed algorithms in TC12-horizon and TC12-t184 databases but better result in TC-10 database. The performance of the proposed algorithms are better than $V_{8,1}^{u2}GM_{PD2}$, whose dimension size is 227, in all of the compared databases. $LBPV_{8,1}^{u2}GM_{PD2}$ and $LBP_{24,3}^{u2}GM_{ES}$ whose dimensions are 2,211 and 13,251, respectively, provide better results than the proposed algorithms at a high cost of increased feature size. The algorithms of [16] are provided in the 10th and 11th rows. It should be noted that their classification algorithm is based on SVM, an algorithm with higher computational complexity. Moreover, their feature vectors have higher dimensions

Table 5 Performance comparison of the proposed algorithms with the literature

	Algorithm	Feature size	Comput. complexity of feature	Class. method	Better than the proposed algorithms	Comparable with the proposed algorithms	Worse than the proposed algorithms
1	$F_{\mu\sigma}^i$	84	Low	NN	-	-	<i>a,b,c,d,e,f</i>
2	$F^{[13]}$	84	Low	NN	-	-	<i>a,b,c,d,e,f</i>
3	$LBP_{8,1}^{riu2}/VAR_{8,1}$ [20]	160	Medium	NN	b	-	<i>c,d</i>
4	$LBPV_{8,1}^{u2}GM_{PD2}$ [20]	227	Medium	NN	-	-	<i>b,c,d</i>
5	$LBP_{24,3}^{riu2}/VAR_{24,3}$ [20]	416	Medium	NN	b	c,d	-
6	$LBP_{8,1}^{u2}GM_{ES}$ [20]	451	Medium	NN	-	-	<i>b,c,d</i>
7	KLD [15]	840	Medium	NN	-	-	<i>a,e</i>
8	$LBPV_{24,3}^{u2}GM_{PD2}$ [20]	2,211	High	NN	b,c,d	-	-
9	$LBP_{24,3}^{u2}GM_{ES}$ [20]	13,251	High	NN	b,c,d	-	-
10	HSR + LSR [16] RIFT	4,000	High	SVM	f	-	<i>a,e</i>
11	HSR + LSR [16] SIFT	5,120	High	SVM	a,e,f	-	-

The table does not include the proposed algorithms and is ordered in the increasing complexity and feature size order (top to bottom). The databases are represented with as follows: a, Brodatz; b, TC10; c, TC12-horizon; d, TC12-t184; e, KTH-TIPS; f, UIUCTex. The italicized letters indicate the databases where the proposed methods are superior.

than those of the proposed algorithms. Even though their algorithm provides better results for HSR + LSR + SIFT, the proposed algorithms outperform HSR + LSR + RIFT in Brodatz and KTH_TIPS databases. This result is suspected to arise from the deduced feature size and less satisfactory performance of RIFT in scale-variant database (KTH-TIPS). In general, the proposed algorithms of this study outperform all of the multiscale-based texture classification algorithms. They also outperform LBP variants of [20] with small dimensions. The performance of the algorithm of [20] with high dimensions is better as expected. Proposed algorithms also outperform the algorithms of [16] in smaller dimensions especially in rotation-variant databases.

Finally, we mention one of the latest works in the rotation and scale-invariant texture retrieval published by Li et al. [21]. They provide scale invariance by finding optimal scale of each pixel. They modify Outex and Brodatz databases to include enough scale and rotation variations and report their results on these databases. For scale and rotation invariance feature, they report average precision rates around 69% for Brodatz and 60% for Outex database. Since they use a modified database, including this database will extend the scope of this study considerably, and we are leaving the scale invariance and the comparison with their database as our future work.

8. Conclusions

Low-dimensional and rotation-invariant curvelet features for multiscale texture retrieval are proposed through two feature vectors in this study. This study is important since it provides the best results for multiscale texture retrieval in the literature to the best of our knowledge. Moreover, the results are comparable with the state-of-the-art techniques in low and medium feature dimension sizes. Rotation invariance is provided by using the cross energies of curvelet blocks at adjacent orientations. The orientations with maximum cross energy are defined as the principle orientation of an image, which is the least affected location by rotation. The corresponding location is selected as the reference point for the image, and the feature vector is cycle-shifted based on this reference point. The feature vector $F_{\mu\sigma}^{PO}$ has 84 elements. The other proposed feature vector F_{KDE}^{PO} uses KDE, and it has 840 elements. It provides better results than $F_{\mu\sigma}^{PO}$ in the expense of increased complexity. The texture retrieval results of the proposed method are better than earlier works which make use of other rotation-invariant curvelet features and are comparable with the state-of-the-art works in the literature, especially in the low and medium feature dimension ranges. As a result, we provide a novel rotation invariance method for curvelets and two separate feature vectors for texture retrieval in this study. The proposed methods suggest

highly effective discriminative power for texture retrieval. The comparisons with the literature show the effectiveness of the proposed algorithms since they provide good performances with low complexity. Addition of scale invariance for curvelet features may provide better results. Thus, we plan to extend this study for scale-invariant features of curvelet transform as our future work.

Competing interests

The author declares that he has no competing interests.

Received: 17 August 2013 Accepted: 1 April 2014

Published: 11 April 2014

References

1. HH Arsenault, YN Hsu, K Chalasinskamacukow, Rotation-invariant pattern-recognition. *Opt. Eng.* **23**, 705–709 (1984)
2. RL Kashyap, A Khotanzad, A model-based method for rotation invariant texture classification. *IEEE T. Pattern Anal.* **8**, 472–481 (1986)
3. SG Mallat, A theory for multiresolution signal decomposition - the wavelet representation. *IEEE T. Pattern Anal.* **11**, 674–693 (1989)
4. BS Manjunath, WY Ma, Texture features for browsing and retrieval of image data. *IEEE T. Pattern Anal.* **18**, 837–842 (1996)
5. MN Do, M Vetterli, The finite ridgelet transform for image representation. *IEEE T. Image Process.* **12**, 16–28 (2003)
6. EJ Candes, DL Donoho, Curvelets, multiresolution representation, and scaling laws. *Wavelet Appl Signal Image ProcessViii Pts 1 and 2* **4119**, 1–12 (2000)
7. EJ Candes, DL Donoho, New tight frames of curvelets and optimal representations of objects with piecewise C-2 singularities. *Commun. Pur. Appl. Math.* **57**, 219–266 (2004)
8. IJ Sumana, M Islam, DS Zhang, GJ Lu, *Content based image retrieval using curvelet transform, vol. 1 and 2 (2008 IEEE 10th Workshop on Multimedia Signal Processing, 2008)* (Queensland, Australia, 2008), pp. 11–16
9. GM Haley, BS Manjunath, Rotation-invariant texture classification using a complete space-frequency model. *IEEE T. Image Process.* **8**, 255–269 (1999)
10. G Tzagkarakis, B Beyerull-Lozano, P Tsakalides, Rotation-invariant texture retrieval with Gaussianized steerable pyramids. *IEEE T. Image Process.* **15**, 2702–2718 (2006)
11. M Kokare, PK Biswas, BN Chatterji, Rotation-invariant texture image retrieval using rotated complex wavelet filters. *IEEE T. Syst. Man. Cy. B* **36**, 1273–1282 (2006)
12. VR Rallabandi, VPS Rallabandi, Rotation-invariant texture retrieval using wavelet-based hidden Markov trees. *Signal Process* **88**, 2593–2598 (2008)
13. DS Zhang, MM Islam, GJ Lu, IJ Sumana, Rotation invariant curvelet features for region based image retrieval. *Int. J. Comput. Vision* **98**, 187–201 (2012)
14. MM Islam, DS Zhang, GJ Lu, *Rotation invariant curvelet features for texture image retrieval, Presented at the Icmc, vol. 1–3 (2009 IEEE International Conference on Multimedia and Expo (New York, NY, USA, 2009)*
15. F Gomez, E Romero, Rotation invariant texture characterization using a curvelet based descriptor. *Pattern Recogn Lett* **32**, 2178–2186 (2011)
16. J Zhang, M Marszalek, S Lazebnik, C Schmid, Local features and kernels for classification of texture and object categories: a comprehensive study. *Int. J. Comput. Vision* **73**, 213–238 (2007)
17. K Mikolajczyk, C Schmid, Scale & affine invariant interest point detectors. *Int. J. Comput. Vision* **60**, 63–86 (2004)
18. D Lowe, Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision* **60**, 91–110 (2004)
19. S Lazebnik, C Schmid, J Ponce, A sparse texture representation using local affine regions. *IEEE T. Pattern Anal. Mach. Intel.* **27**, 1265–1278 (2005)
20. ZH Guo, L Zhang, D Zhang, Rotation invariant texture classification using LBP variance (LBVP) with global matching. *Pattern Recogn* **43**, 706–719 (2010)
21. Z Li, GZ Liu, Y Yang, JY You, Scale- and rotation-invariant local binary pattern using scale-adaptive texton and subuniform-based circular shift. *IEEE T. Image Process.* **21**, 2130–2140 (2012)
22. S Lazebnik, C Schmid, J Ponce, *Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in Computer Vision and Pattern Recognition (IEEE Computer Society Conference on, 2006 (New York, NY, USA, 2006)*, pp. 2169–2178

23. B Silverman, *Density Estimation for Statistics and Data Analysis* (Chapman & Hall, London, 1986)
24. P Brodatz, *Textures: A Photographic Album for Artists and Designers* (Dover, NewYork, 1966)
25. T Ojala, T Maenpaa, M Pietikainen, J Viertola, J Kyllonen, S Huovinen, *Outex - new framework for empirical evaluation of texture analysis algorithms, in Pattern Recognition, 2002, vol. 1 (Proceedings. 16th International Conference on, 2002)* (Quebec City, QC, Canada), pp. 701–706
26. E Hayman, B Caputo, M Fritz, J-O Eklundh, On the significance of real-world conditions for material classification, in *Computer Vision*, ed. by T Pajdla, J Matas (Springer, Berlin Heidelberg, 2004), pp. 253–266

doi:10.1186/1687-5281-2014-22

Cite this article as: Cavusoglu: Multiscale texture retrieval based on low-dimensional and rotation-invariant features of curvelet transform. *EURASIP Journal on Image and Video Processing* 2014 **2014**:22.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Immediate publication on acceptance
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ springeropen.com
