Rotation Invariant Local Frequency Descriptors for Texture Analysis

Rouzbeh Maani, Sanjay Kalra, Yee-Hong Yang

***Abstract*—This paper presents a novel, simple, yet powerful texture analysis method inspired by the well-known Local Binary Patterns (LBP) method called the Local Frequency Descriptors (LFD). Like LBP, the proposed method is invariant to rotation and linear changes of illumination; however,it does not suffer from the limitations of LBP such as exponential growth of features with an increment in the number of neighbors. The experimental results on the Outex and CUReT datasets show that the proposed LFD method outperforms state-of-the-art texture analysis methods. In addition, LFD is very robust to noise and can improve LBP results up to 50% in extremely noisy conditions. In this paper, we discuss different aspects of the LFD and explain how it addresses the main limitations of LBP and of its variants.**

I. Introduction

TEXTURE analysis is an important topic in image processing and has been used in many applications including automated inspection, image retrieval and medical image analysis. Image textures are defined as visual patterns appearing in images. Texture analysis methods use chromatic and structural characteristics of images to characterize textures. The methods usually consist of five steps:

1) Pre-processing: The images are normalized in this step. The purpose of normalization is to bring the images to a standard range, such that the extracted properties from the images are comparable.
2) Feature extraction: Textural features of images are extracted in this step. Different methods are used to find the textural features (e.g., statistical information, frequency analysis, etc.).
3) Feature selection: In this step, useful features are selected. Sometimes the number of features is huge. As well, some features may not be informative. The goal of this step is to reduce the number of extracted features by selecting those giving important textural information.
4) Classification: In this step, each image is assigned to one of the known texture classes. Basically, there are two sets: a training set and a test set. Classification is performed to assign images in the test set to one of the texture classes learned from the training set. Different classification methods can be used in this step. Some examples are Support Vector Machine (SVM), and Nearest neighbor (NN) classifiers.

All methods referred to as texture analysis methods are used in the second step to extract textural features. The contribution of this paper is to present a new method based on the local frequencies in images. The proposed method is inspired by the successful and popular Local Binary Patterns (LBP) method [1]; however, it addresses the key problems of the LBP. The features defined in the proposed method are (1) invariant to rotation, (2) invariant to linear changes of illumination, (3) robust to noise , (4) tunable, and (5) fewer in number.

In the next sections, this paper reviews relevant previous works (Section II), presents our method (Section III), explains the experimental results (Section IV), and concludes in Section V.

II. Related Works

There are many different methods for texture analysis; however, they can be categorized into four general groups. The first group uses statistical features. The main motivation behind these methods is based on the fact that the human visual system uses statistical features to distinguish textures. The co-occurrence matrix proposed by Haralik and Shanmugam [2] is one of the first known methods using this approach. The co-occurrence matrix represents the relationship between intensity levels for a given direction and distance in the image. Co-occurrence matrices with different directions and distances provide a popular method to distinguish textures. The Run Length Matrices (RLM) [3], [4] method defines a gray level run as consecutive pixels of the same gray level in a given direction, and the length of the runs is used to describe textures. There are other statistical methods such as using higher order statistics [5], [6] and invariant moments of the images [7], [8]. Recently, Local Binary Patterns (LBP) proposed by Ojala et al. [1] has been recognized as one of the most successful statistical methods and has been extended by different research groups [9], [10], [11], [12], [13]. The method represents the relationship of each pixel and its neighbors (located on a circle around the pixel) by a binary pattern and uses the histogram of these patterns for texture classification. The main drawback of LBP, however, is the exponential growth of the number of patterns with respect to the number of neighbors. We will also show in the experimental results that the LBP and its variants are not robust to noise.

The second group of texture analysis methods uses structural features of images. These methods decompose textures into elements known as primitives or texels. The primitives and their spatial arrangements are used to characterize textures. For

R. Maani and Y.H. Yang are with the Department of Computing Science, University of Alberta, Canada.
E-mail: rmaani@ualberta.ca
S. Kalra is with the Departments of Medicine and Biomedical Engineering, University of Alberta, Canada.

example, Hanbury et al. [14] used morphological operations to characterize textures. Song [15] decomposed textures into a set of scale images, found square texels of the same size at each scale and used the histogram of the texels as texture features. Gui et al. [16] extracted the size, position, periodicity, and spatial organization of texels to analyze textures. Khellah [17] used the similarity between pixels and their surrounding neighbors within a predefined window and generated a global map called the dominant neighborhood structure. The features extracted from this map along with the features obtained from the LBP were used for texture classification. The key problem of the structural based methods is how to define texels that represent all different texture structures. In general, the structural-based methods are better suited for textures with large structures (macrostructure) and do not work well on non-structural textures and microtextures [18].

The third class of texture methods defines textures as probability models. Some well-known models are Markov Random Field (MRF) [19], Auto Regressive (AR) model [20], and Gibbs random field [21]. The key issue in these models is how to choose the correct model for a given texture and how to effectively map a texture into the selected probability model [18]. In addition, each model imposes some assumptions that may not be true for all textures. For instance, MRF assumes that the probability of each pixel depends only on its neighbors which may not be correct for all textures.

The fourth and last approach to analyze textures applies filters on images in spatial domains or analyzes images in frequency domain. For instance, Azencott et al. [22] made use of windowed Fourier filters for texture analysis. Chang and Kuo [23] used tree-structured wavelet transform on textures to extract features. Jafari-Khouzani and Soltanian-Zadeh [24] extracted rotation invariant features by using Radon transform and a translation-invariant wavelet transform. Chu and Chan [25] used tunable Gabor filter banks to define rotation and scale invariant features. The main advantage of these methods that use frequency components is the capability of handling noise. However, the frequency-based methods usually cannot capture local changes in textures. As a result some research studies use spatial domain to define textural features. Leung and Malik [26], Cula and Dana [27], and Varma and Zisserman [28] applied spatial filters on the textures, and computed the frequency histogram of filter response cluster centers as features. Later, Varma and Zisserman [29] substituted the local patches of the original image for the filter responses. Nonetheless, these spatial filter methods are not able to handle noise like their counterparts that use frequency information.

Our proposed method can be categorized in the last group; however, it is inspired by the LBP method (i.e., statistical). The proposed method addresses some important issues of LBP such as reducing the number of patterns and noise sensitivity.

III. The Proposed Method

In this section we explain the proposed method. Since the method is inspired by LBP, we give a brief overview of the LBP approach and its variants in section III-A. The proposed method is then introduced in section III-B.

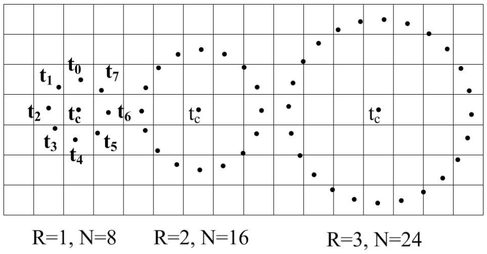

Fig. 1. Three common neighbor settings in LBP.

A. LBP and Its Variants

Traditionally, LBP considers N points on a circle with radius R at center pixel, t_c. These N points $(t_0, t_1, ..., t_{N-1})$ are the neighbors of the center pixel and their gray level values are determined by interpolation if they are not located at the center of pixels. Figure 1 shows three popular configurations with radius of one, two, and three and their corresponding neighboring size of 8, 16 and 24. The eight neighbors in the first neighbor setting are labeled.

The sign of the differences of the gray value of the center pixel with the neighboring pixels are computed. The sign is one if the difference is greater than or equal to zero, and zero otherwise. By assigning a binomial factor 2^n to the sign of differences and summing them together, a binary number (pattern) is created:

$$LBP_{N,R} = \sum_{n=0}^{N-1} s(t_n - t_c).2^n, s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}, \quad (1)$$

where s is the sign function, and N is the number of neighbors. To have rotation invariance, the binary pattern is circularly shifted and the minimum value is kept as the final binary pattern. In other words, the rotation invariant LBP is defined as:

$$LBP^{ri}_{N,R} = min\{ROR(LBP_{N,R}, i) | i = 0, 1, ..., N-1\}, \quad (2)$$

where $ROR(x, i)$ performs i times bitwise circular right shift on the binary number x.

The major problem of LBP is the exponential growth of the number of patterns with respect to the neighborhood size. To address this problem, several methods have been proposed. Ojala et al. [1] observed that some binary patterns are more common than others in some textures. These patterns known as uniform patterns have a common property: the number of spatial transition between zero and one in the binary pattern (i.e., uniformity) is at most two. The uniformity measure is defined as:

$$\begin{aligned} U(LBP_{N,R}) = |s(t_{N-1} - t_c) - s(t_0 - t_c)| + \\ \sum_{n=1}^{N-1} |s(t_n - t_c) - s(t_{n-1} - t_c)|. \end{aligned} \quad (3)$$

Rotation invariant uniform patterns have the uniformity of two or fewer and are defined as:

$$LBP^{riu2}_{N,R} = \begin{cases} \sum_{n=0}^{N-1} s(t_n - t_c).2^n & \text{if } U(LBP_{N,R}) \leq 2 \\ N+1 & \text{otherwise.} \end{cases} \quad (4)$$

To further improve the results, they also introduced $VAR_{N,R}$ operation as follows and used the joint histogram of $LBP^{riu2}_{N,R}/VAR_{N,R}$ for classification [1]:

$$VAR_{N,R} = \frac{1}{N}\sum_{n=0}^{N-1}(t_n-\mu)^2,\ \mu = \frac{1}{N}\sum_{n=0}^{N-1}t_n. \tag{5}$$

Later, Guo et al. [13] argued that the VAR operation produces continuous values and the correct quantization of these values is challenging. They proposed LBP Variance (LBPV) in which the histogram of $LBP_{N,R}$ is weighted by the $VAR_{N,R}$ instead of using the joint histogram of $LBP^{riu2}_{N,R}/VAR_{N,R}$. The histogram of $LBPV_{N,R}$ is computed as:

$$LBPV_{N,R}(k) = \sum_{i=1}^{N}\sum_{j=1}^{M} w(LBP_{N,R}(i,j),k),\ k \in [0,K], \tag{6}$$

where N and M are, respectively, the number of rows and columns of LBP, and the weight is computed as follows:

$$w = \begin{cases} VAR_{N,R}(i,j) & \text{if } LBP_{N,R}(i,j) = k \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Before computing $LBPV_{N,R}$, they use LBP to find the principal orientation of texture and align the binary patterns to that orientation (i.e. global matching).

The next approach for an efficient reduction of the number of binary patterns was presented by Liao et al. [10]. They showed that the uniform patterns are not necessarily the dominant patterns in all datasets. They suggested choosing the dominant patterns in the textures instead of the uniform ones. Unlike the uniform patterns, the number of patterns for classification is not constant and is determined by choosing the dominant patterns such that they consist of 80% of the whole patterns:

$$k = \underset{k}{\operatorname{argmin}}\left(\frac{\sum_{n=0}^{k-1}H[n]}{\sum_{n=0}^{2^N-1}H[n]} \geq 80\%\right), \tag{8}$$

where N is the neighborhood size in $LBP_{N,R}$, and H the histogram of patterns sorted in a descending order. The final histogram for classification is $H[0...k]$. They also used the Normalized Gabor Filter (NGF) responses of the frequency spectrum to improve the classification rate.

Guo et al. [30] used the Fisher separation criterion to choose patterns. In their approach, the dominant patterns in each image are found. The representative patterns of each texture class are computed as the intersection of patterns that are dominant in all images of the same texture class. Finally, the union of all class representative patterns is used to select the final patterns. The proposed approach tries to maximize the inter-class distance and to minimize the intra-class similarity (i.e., the Fisher separation criterion).

Recently, Guo et al. [12] suggested the completed model of LBP (CLBP) in which they used not only the sign of the difference between the center pixel and its neighbors, but also the magnitude of this difference and the magnitude of the center pixel. They defined three operators: $CLBP\_S$, $CLBP\_M$, and $CLBP\_C$. The first operator is the same as the ordinary $LBP_{N,R}$ and makes a binary pattern based on the sign of the difference of the center pixel and its neighbors. To make a binary pattern from the magnitude of difference $CLBP\_M$ is defined as follows:

$$CLBP\_M_{N,R} = \sum_{n=0}^{N-1} s(t_n - c).2^n, \tag{9}$$

where s is the sign function defined in equation (1) and c an adaptive threshold set to the mean value of t_n. Finally, to take the center pixel's value into account $CLBP\_C$ is defined as:

$$CLBP\_C_{N,R} = s(t_c - c_I), \tag{10}$$

where s is the sign function and c_I a threshold set to the average gray value of all pixels. They used the joint and concatenated frequency histogram of patterns produced by $CLBP\_S$, $CLBP\_M$, and $CLBP\_C$.

B. Local Frequency Descriptors

Our approach is based on the same sampling method of LBP. A circle with radius R and N samples is considered for each pixel $(t_0, t_1, ..., t_{N-1})$. The arrangement of these samples results in a function called $LBP\_FUN_{N,R}$ in this paper. Recall equation (1), we see that LBP uses the value of the center pixel as a threshold to encode $LBP\_FUN_{N,R}$ to a binary number. Similarly, CLBP uses two different thresholds to make two binary patterns from this function using $CLBP\_S$ and $CLBP\_M$ operations.

Our method is based on the fact that the $LBP\_FUN_{N,R}$ function carries important textural features which is the reason that LBP and its variants have become so successful. Nonetheless, thresholding the function by LBP-based methods will remove some important information. To preserve the information we transform the $LBP\_FUN_{N,R}$ function into the frequency domain using the Discrete Fourier Transform (DFT). It is noteworthy that rotation makes a circular shift on the function and does not change the magnitude of its frequency components. The frequency components generated from the $LBP\_FUN_{N,R}$ function are called local frequencies in this paper and used to define texture features.

To formally define the local frequencies, consider N neighbors $t_i (i = 0, ..., N-1)$ of R for each pixel. If a center pixel is located at (x, y) then the coordinates of t_i are given by $(x - R.sin(2i\pi/N), y + R.cos(2i\pi/N))$. The local frequency components f_n are defined as follows:

$$f_n = \sum_{k=0}^{N-1} t_k e^{-2\pi ink/N}. \tag{11}$$

Applying the DFT to a discrete function with N points generates N complex numbers the magnitudes of which are used. Since N/2-1 components have the same magnitude, we have N/2+1 components with different magnitudes $(f_0...f_{N/2+1})$. Therefore, there are N/2+1 frequency channels, in which each channel consists of the absolute value of the n^{th} frequency component of $LBP\_FUN_{N,R}$. Figure 2 shows some sample textures and the magnitude of the first two frequency components computed by (N,R)=(8,1).

The local frequencies capture local texture properties. To analyze these properties we consider each frequency channel

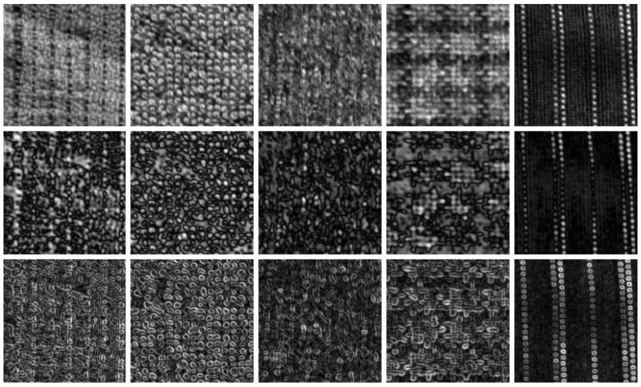

Fig. 2. Five texture samples and their first two frequency channels. The first row shows the five texture samples taken from Outex_TC10 dataset [31]. The second and third rows show the first and second frequency channels of the textures computed by (N,R)=(8,1).

separately. To get the correlation between local frequency components in each channel, we employ the 2D DFT. In other words, we consider each channel as an image and compute the Fourier transform of that channel. The 2D frequency spectrum of each channel CH_n is computed as:

$$CH_n(k,l)=\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}|f_n(x,y)|.e^{-2\pi i(\frac{xk}{X}+\frac{yl}{Y})}, \qquad (12)$$

where X and Y are the number of columns and rows respectively, assuming that the coordinate system ranges from [0,0] to [X-1,Y-1], and $|f_n(x,y)|$ is the magnitude of the n^{th} frequency component of local function $LBP\_FUN_{N,R}$ computed at (x,y). It is noteworthy that although the magnitude of the frequency components of $LBP\_FUN_{N,R}$ does not change by rotation, the location of the functions rotates by rotation.

To have rotation invariant features we define circular disks filters on the spectrum of frequency channels CH_n. These filters are defined as:

$$D_{r_1,r_2}(x,y)=\begin{cases} 1 & \text{if } r_1\leq\sqrt{x^2+y^2}\leq r_2 \\ 0 & \text{otherwise,}\end{cases} \qquad (13)$$

where r_1 and r_2 are the radii representing the inner and outer boundaries of the disk. We may note that before applying these circular disks, we circularly shift the spectrum to translate the frequency component (0,0) to the center of the spectrum. We define the rotation invariant features by applying these circular band-pass filters to the magnitude of spectrum of frequency channels CH_n. We call these features LFD_C (C stands for Circular filter used for computing the features):

$$LFD\_C(r_1,r_2,n)=\frac{\sum\limits_{k=-K/2}^{K/2-1}\sum\limits_{l=-L/2}^{L/2-1}|CH_n(k,l)|.D_{r_1,r_2}(k,l)}{\sum\limits_{k=-K/2}^{K/2-1}\sum\limits_{l=-L/2}^{L/2-1}D_{r_1,r_2}(k,l)}, \qquad (14)$$

where $|CH_n|$ is the magnitude of shifted spectrum of frequency channels computed by (12) and K and L are the number of columns and rows, respectively. The factor in the denominator averages the response of the disk filter and removes the effect of the disk's size.

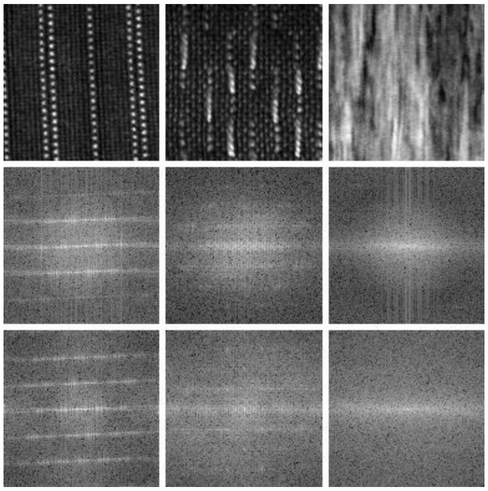

Fig. 3. Directionality of textures. The first row shows three texture samples taken from Outex_TC10 dataset [31]. The second and third rows show the logarithm of the magnitude of CH_1 and CH_2 computed by (N,R)=(8,1).

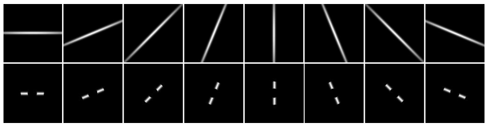

Fig. 4. Directional filters equally distributed in eight directions. First row, directional Gaussian filters. Second row band-pass directional Gaussian filters.

Although using disks makes the features rotation invariant, it removes the directionality information of the spectrum. In fact, some textures have directionality and capturing that information can lead to better discrimination. Figure 3 shows an example in which the texture has directional information appeared in the spectrum of frequency channels.

To capture the directionality information, we use directional filters inspired by Varma and Zisserman [28]. We define the directional filter-bank by means of Gaussian filters at multiple orientations:

$$G_{\theta,\sigma_1,\sigma_2}(x,y)=\frac{1}{2\pi\sigma_1\sigma_2}e^{-(\frac{(x\cos(\theta)-y\sin(\theta))^2}{2\sigma_1^2}+\frac{(x\sin(\theta)+y\cos(\theta))^2}{2\sigma_2^2})} \qquad (15)$$

where θ is the direction of the filter and σ_1 and σ_2 control the width and length of the filter, respectively. We combine these filters with the previously mentioned disk filters to have directional band-pass filters. The combination of the disk and Gaussian filters (called DG in this paper) are formally defined as the multiplication of the two filters:

$$DG_{\theta,\sigma_1,\sigma_2,r_1,r_2}(x,y)=D_{r_1,r_2}(x,y)\times G_{\theta,\sigma_1,\sigma_2}(x,y). \qquad (16)$$

Figure 4 shows the directional and band-pass directional Gaussian filters. The first row consists of directional Gaussian filters with $\sigma_1=64$ and $\sigma_2=2$ equally distributed on eight directions. The second row shows the band-pass version of the same filters.

These directional band-pass filters are applied to the spectrum of frequency channels CH_n. However, to have rotation invariant features, we keep the maximum, the minimum, and the median responses. The idea is similar to the work of Varma and Zisserman [28]; however, they keep only the maximum response. The directional features ($LFD\_D(\theta, \sigma_1, \sigma_2, r_1, r_2)$) produced by directional band-pass Gaussian filters are formally defined as (we drop the subscript variables $r_1, r_2, \theta, \sigma_1$, and σ_2 from the notation of DG and LFD_D for brevity):

$$LFD\_D_{min} = \min_{\theta\in\Theta} \sum_{k=-K/2}^{K/2-1} \sum_{l=-L/2}^{L/2-1} |CH_n(k,l)|.DG(k,l) \tag{17}$$

$$LFD\_D_{max} = \max_{\theta\in\Theta} \sum_{k=-K/2}^{K/2-1} \sum_{l=-L/2}^{L/2-1} |CH_n(k,l)|.DG(k,l) \tag{18}$$

$$LFD\_D_{med} = \operatorname*{median}_{\theta\in\Theta} \sum_{k=-K/2}^{K/2-1} \sum_{l=-L/2}^{L/2-1} |CH_n(k,l)|.DG(k,l), \tag{19}$$

where Θ is the set of directions (e.g., for a set of eight directions $\Theta = \{0, 2\pi/8, 4\pi/8, ..., 14\pi/8\}$).

The third set of features in this paper is obtained from the energy of the local frequency patterns. This energy is computed from the local frequency components. First, we find the local frequency components by (11) and then we compute the energy at each pixel as follows:

$$E(x,y) = \sum_{n=1}^{N/2} |f_n(x,y)|^2, \tag{20}$$

where N is the number of samples in circles with radius R (i.e., the number of all frequency channels).

Finally we use the global frequencies of the textures as the last set of features by applying the 2D Fourier transform to the image. The last two sets of features are defined by applying the disk filters (defined in (13)) to the energy and the global frequencies. These two sets are called LFD_E and LFD_G and defined formally as follows:

$$LFD\_E(r_1,r_2) = \frac{\sum\limits_{k=-K/2}^{K/2-1} \sum\limits_{l=-L/2}^{L/2-1} E(k,l).D_{r_1,r_2}(k,l)}{\sum\limits_{k=-K/2}^{K/2-1} \sum\limits_{l=-L/2}^{L/2-1} D_{r_1,r_2}(k,l)} \tag{21}$$

$$LFD\_G(r_1,r_2) = \frac{\sum\limits_{k=-K/2}^{K/2-1} \sum\limits_{l=-L/2}^{L/2-1} |G(k,l)|.D_{r_1,r_2}(k,l)}{\sum\limits_{k=-K/2}^{K/2-1} \sum\limits_{l=-L/2}^{L/2-1} D_{r_1,r_2}(k,l)}, \tag{22}$$

where G is the 2D Fourier transform of the texture image, and the coordinates range from $[-K/2, -L/2]$ to $(K/2, L/2)$.

The final feature vector consists of these four sets of features: $LFD\_C(r_1, r_2, n)$, $LFD\_D(r_1, r_2, n)$, $LFD\_E(r_1, r_2)$, and $LFD\_G(r_1, r_2)$.

To classify data the nearest neighborhood (NN) classifier with χ^2 distance is used. The chi-square distance between two feature vectors v_1 and v_2 is computed as follows:

$$D(v_1, v_2) = \sum_{n=1}^{N} \frac{(v_1(n) - v_2(n))^2}{v_1(n) + v_2(n)}, \tag{23}$$

where $v(n)$ is the n^{th} feature in the vector and N is the total number of features. The experimental results are discussed in the next section.

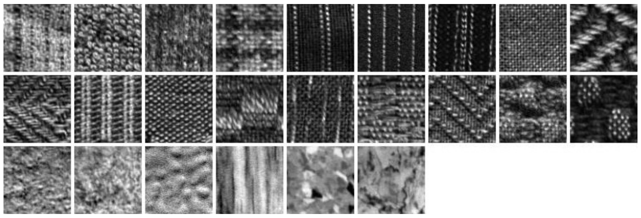

Fig. 5. The Outex dataset includes 24 different texture classes.

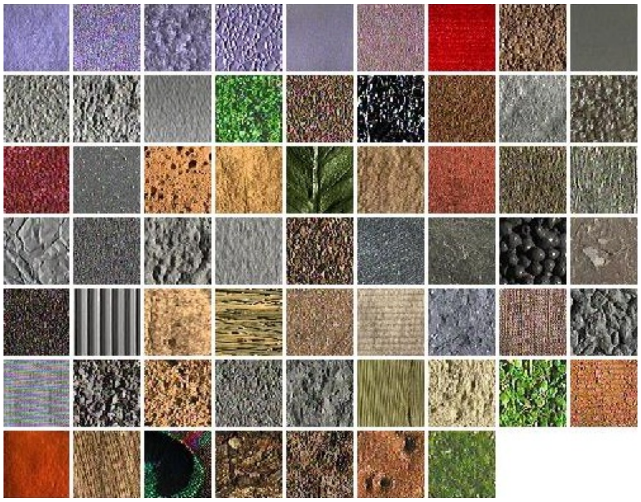

Fig. 6. The CUReT dataset includes 61 different texture classes.

IV. Experimental Results

To assess our proposed method, we use two well-known comprehensive datasets designed for evaluation of rotation and illumination changes: the Outex [31] and the Columbia-Utrecht Reflectance (CUReT) [32] datasets. The Outex dataset includes 24 texture classes shown in Figure 5.

The CUReT dataset consists of 61 real-world textures acquired under different viewing angles and illumination conditions. Figure 6 shows the texture classes in the CUReT dataset. The experimental results on each of these datasets are demonstrated in the next subsections.

A. Outex Dataset

The Outex dataset is one of the well-known datasets used for evaluation. The images are produced by rotating textures by nine different angles ($0°, 5°, 10°, 15°, 30°, 45°, 60°, 75°$, and $90°$) and under three different illuminations conditions (“horizon”, “inca”, and “t184”). There are 20 non-overlapping 128×128 gray-level images for each class produced under each condition. The dataset consists of two test suites: Outex_TC10 and Outex_TC12.

TABLE I
CLASSIFICATION RATE ON THE OUTEX DATASET

Method	TC10	TC12-“t184”	TC12-“horizon”
VZ_Joint	92.00	91.41	91.82
VZ_MR8	93.59	92.55	92.99
$LBP^{riu2}_{24,3}$	95.07	85.04	80.78
$LBP^{riu2}_{24,3}/VAR_{24,3}$	98.15	87.13	87.08
$DLBP_{16,2}+NFG$ [10]	99.1	93.2	90.4
$DNS+LBP_{24,3}$ [17]	99.27	94.40	92.85
$CLBP\_S^{riu2}_{24,3}/M^{riu2}_{24,3}$	99.32	93.58	93.35
$CLBP\_S^{riu2}_{24,3}/M^{riu2}_{24,3}/C$	98.93	95.32	94.53
$LBPV^{u2}_{24,3}GM_{ES}$	97.76	95.39	95.57
$LFD_{24,3}$	**99.84**	**97.69**	**98.47**

The Outex_TC10 has been designed for rotation invariant analysis. In this test suite, the training set consists of images acquired under illumination condition “inca” and angle 0° (i.e., 24×1×1×20=480 samples). The test set includes images produced by the same illumination condition, “inca”, but rotated by eight different angles, resulting in 3840 (24×1×8×20) images.

The Outex_TC12 test suite aims to analyze both rotation and illumination invariance. The training set is the same as the one in Outex_TC10. However, the test set consists of two datasets: the first test set includes the images produced under “t184” illumination condition and the second test set consists of images acquired under “horizon” illumination condition. Each set includes all rotation conditions which makes each has 4320 (24×1×8×20) samples.

Table I compares classification rate for the proposed method and some state-of-the-art methods on the Outex dataset. In each LBP-based method the results are shown for the (N,R) setting which has the highest classification rate. For $CLBP$, the 2D joint histogram of $CLBP\_S$ and $CLBP\_M$ ($CLBP\_S^{riu2}_{N,R}/M^{riu2}_{N,R}$) and the 3D joint histogram of $CLBP\_S$, $CLBP\_M$ and $CLBP\_C$ ($CLBP\_S^{riu2}_{N,R}/M^{riu2}_{N,R}/C$) are shown which produce the highest classification rate in the $CLBP$ method. Sixteen bins were used for $LBP^{riu2}_{24,3}/VAR_{24,3}$ quantization according to [1]. $LBPV^{u2}_{N,R}GM_{ES}$ uses the histogram of LBP weighted by variance and an exhaustive search for global matching [13]. The proposed $LFD_{24,3}$ uses four frequency channels.

As one can see, the LFD method outperforms in all test suites of the Outex dataset. The $CLBP$ operators are next in TC10 while $LBPV\_GM$ stands higher than $CLBP$ in TC12 test suites. The non-LBP methods, VZ_Joint and VZ_MR8, have the lowest performance for TC10, but are better than the original LBP method, $LBP^{riu2}_{24,3}$ and $LBP^{riu2}_{24,3}/VAR_{24,3}$, for TC12. It can be observed that the accuracy of LFD is remarkably better than the other methods particularly for Outex_TC12. For instance, the method is about 2% and 3% more accurate than the second top method in TC12-“t184” and TC12-“horizon” and the results are better than that of the original LBP by more than 12% and 17% on TC12-“t184” and TC12-“horizon”, respectively.

TABLE II
CLASSIFICATION RATE ON THE CUReT DATASET

Method	Accuracy
VZ_Joint	97.15
VZ_MR8	97.51
$LBP^{riu2}_{24,3}$	87.53
$LBP^{riu2}_{24,3}/VAR_{24,3}$	92.23
$CLBP\_S^{riu2}_{24,3}/M^{riu2}_{24,3}$	93.83
$CLBP\_S^{riu2}_{24,3}/M^{riu2}_{24,3}/C$	96.12
$LBPV^{u2}_{24,3}GM_{ES}$	94.44
$LFD_{24,3}$	**97.93**

B. CUReT Dataset

The CUReT dataset is the other famous dataset that we used for evaluation. The dataset includes 205 images from 61 texture types. This dataset is very challenging because some of the samples are visually very similar. Similar to [28] and [12], we select 92 images that are large enough to be cropped to an area of 200×200 pixels. The selected images are converted to gray-level before analysis. These 92 images are selected alternatively for training and testing, that is, the odd numbered images are chosen for the training set and the even numbered images are used as the test set. As a result, each training and test set includes 46 images. Similar to [12] we use the first 23 samples in each class for learning texton dictionary in the VZ_Joint and VZ_MR8 methods and for computing the cutting value of VAR operation. In this dataset, we used nine frequency channels to compute the $LFD_{24,3}$ features. Table II compares the classification rate achieved by different methods on the CUReT dataset.

We can see that the LFD method has the highest classification rate. The method improves the accuracy of the original LBP by more than 10%. Next to the LFD, the VZ_MR8 and VZ_Joint stand. This is interesting that these two methods are among the lowest performing methods on the Outex dataset.

C. Noise Robustness

In many applications we need to deal with noisy images. As a result, robustness to noise is considered as one of the most important factors to assess texture methods. The three test suites of the Outex dataset are used for the noise robustness experiments. In each experiment, a random Gaussian noise with a specific Signal to Noise Ratio (SNR) is added. To reduce variability of the randomness, each experiment is repeated five times. Table III shows the average and standard deviation of the classification rate for the TC10 test suite.

We can see from Table III that the LFD method outperforms other methods in all levels of noise. The main reason is that the proposed LFD is a filter-based method and has a great advantage of handling noise in comparison with the other LBP-based methods. Noise usually affects high frequency components. The disk filters (D_{r_1,r_2} defined in (13)) are band-pass filters and therefore remove the unwanted noisy information. We can see that not only LFD is robust to noise level of up to SNR=5, but also it outperforms in situations with higher levels of noise.

TABLE III
CLASSIFICATION RATE ON TC10 WITH DIFFERENT SIGNAL TO NOISE RATIO.

	SNR=30	SNR=15	SNR=10	SNR=5	SNR=4	SNR=3
$LBP^{riu2}_{24,3}$	93.76±0.07	91.97±0.27	85.46±0.13	59.27±0.23	45.98±0.22	33.36±0.10
$LBP^{riu2}_{24,3}/VAR_{24,3}$	97.25±0.09	96.98±0.12	96.01±0.11	73.82±0.28	60.71±0.29	43.12±0.16
$CLBP\_S^{riu2}_{24,3}/M^{riu2}_{24,3}$	98.90±0.07	98.86±0.09	98.32±0.11	72.79±0.21	53.49±0.16	36.81±0.25
$CLBP\_S^{riu2}_{24,3}/M^{riu2}_{24,3}/C$	99.01±0.08	98.89±0.05	98.55±0.10	82.67±0.21	61.73±0.19	40.38±0.17
$LBPV^{u2}_{24,3}GM_{ES}$	98.08±0.13	97.86±0.10	97.06±0.11	91.10±0.12	86.44±0.11	68.65±0.26
$LFD_{24,3}$	**99.84±0.00**	**99.82±0.02**	**99.80±0.04**	**98.79±0.04**	**95.64±0.15**	**83.48±0.15**

The interesting observation is the remarkably huge gap in performance between LFD and other methods in extremely noisy conditions (SNR=4,3) where the LFD can improve the accuracy of the original LBP by about 50%. The second best method in high levels of noise is $LBPV$ which is about 7%, 9%, and 15% less accurate than LFD when SNR is equal to 5, 4, and 3, respectively. It is noteworthy that the $LBPV$ is inferior to $CLBP$ methods in low levels of noise. It can also be observed that the original LBP is the most noise sensitive method in this dataset.

D. Discussion

As shown by the experimental results, the proposed LFD method is an accurate and robust method. In this section we discuss some advantages and challenges of the method. The first and the most important advantage of the proposed method is accuracy. The state-of-the-art methods investigated in this paper usually outperform on one test suite. For instance, if we disregard LFD then the best methods for TC10, TC12, and CUReT are $CLBP\_S/M$, $LBPV$, and VZ_MR8 respectively. These methods do not outperform on all test suites and in fact they sometimes perform very poorly on the other test suites (e.g., VZ_MR8 on TC10 and TC12). Nonetheless, the LFD method outperforms consistently on all these test suites.

Another advantage of the method is that it can provide good results in extremely noisy situations. The main reason is that the LFD is a filter-based method and can handle noise effectively. This advantage can be used to get even higher accuracy by tuning the disk filters to remove higher frequency components in noisy situations.

V. CONCLUSIONS

This paper presents a new, simple and powerful method called local frequency descriptors (LFD). The method uses local frequency components extracted from the LBP function. Similar to LBP, the proposed method provides features invariant to rotation and linear changes of illumination. However, the proposed method does not suffer from the limitations of LBP and its variants. For example, the proposed LFD method does not have the exponential growth of patterns problem that is common among all LBP-based methods. Indeed, LFD uses the frequency representation which is compact and more informative than the binary patterns. The binary pattern representation is also sensitive to noise because only one outlier can change the pattern; however, the frequency representation in LFD can address the problem by removing the higher frequency components that are usually susceptible to noise. The experimental results show that it outperforms state-of-the-art texture analysis methods on the Outex and CUReT datasets.

REFERENCES

[1] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[2] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.

[3] M. M. Galloway, "Texture analysis using gray level run lengths," *Computer Graphics and Image Processing*, vol. 4, no. 2, pp. 172–179, 1975.

[4] X. Tang, "Texture information in run-length matrices," *IEEE Transactions on Image Processing*, vol. 7, no. 11, pp. 1602–1609, 1998.

[5] V. Murino, C. Ottonello, and S. Pagnan, "Noisy texture classification: A higher-order statistics approach," *Pattern Recognition*, vol. 31, no. 4, pp. 383–393, 4 1998.

[6] M. K. Tsatsanis and G. B. Giannakis, "Object and texture classification using higher order statistics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 7, pp. 733–750, 1992.

[7] L. Wang and G. Healey, "Using zernike moments for the illumination and geometry invariant classification of multispectral texture," *IEEE Transactions on Image Processing*, vol. 7, no. 2, pp. 196–203, 1998.

[8] D.-G. Sim, H.-K. Kim, and R.-H. Park, "Invariant texture retrieval using modified zernike moments," *Image and Vision Computing*, vol. 22, no. 4, pp. 331–342, 4/1 2004.

[9] X. Huang, S. Z. Li, and Y. Wang, "Shape localization based on statistical method using extended local binary pattern," in *Image and Graphics, 2004. Proceedings. Third International Conference on*, 2004, pp. 184–187.

[10] S. Liao, M. W. K. Law, and A. C. S. Chung, "Dominant local binary patterns for texture classification," *IEEE Transactions on Image Processing*, vol. 18, no. 5, pp. 1107–1118, 2009.

[11] M. Heikkila, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognition*, vol. 42, no. 3, pp. 425–436, 3 2009.

[12] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657–1663, 2010.

[13] ——, "Rotation invariant texture classification using lbp variance (lbpv) with global matching," *Pattern Recognition*, vol. 43, no. 3, pp. 706–719, 3 2010.

[14] A. Hanbury, U. Kandaswamy, and D. Adjeroh, "Illumination-invariant morphological texture classification," in *Mathematical Morphology: 40 Years On*, ser. Computational Imaging and Vision, C. Ronse, L. Najman, and E. Decenciere, Eds., vol. 30, Vienna University of Technology PRIP group, Institute of Computer-Aided Automation Favoritenstrae 9/1832 A-1040 Vienna Austria. Springer Netherlands, 2005, pp. 377–386.

[15] Q. Song, *A New Multi-scale Texture Analysis with Structural Texel*, ser. Advanced Research on Computer Science and Information Engineering. Springer Berlin Heidelberg, 2011, vol. 152, pp. 61–66.

[16] Y. Gui, M. Chen, L. Ma, and Z. Chen, "Texel based regular and near-regular texture characterization," in *International Conference on Multimedia and Signal Processing (CMSP)*, vol. 1, 2011, pp. 266–270.

[17] F. M. Khellah, “Texture classification using dominant neighborhood structure,” *IEEE Transactions on Image Processing*, vol. 20, no. 11, pp. 3270–3279, 2011.

[18] J. Zhang and T. Tan, “Brief review of invariant texture analysis methods,” *Pattern Recognition*, vol. 35, no. 3, pp. 735–747, 2002.

[19] S. Yousefi and N. Kehtarnavaz, “A new stochastic image model based on markov random fields and its application to texture modeling,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 1285–1288.

[20] N. Abbadeni, “Texture representation and retrieval using the causal autoregressive model,” *Journal of Visual Communication and Image Representation*, vol. 21, no. 7, pp. 651–664, 10 2010.

[21] I. M. Elfadel and R. W. Picard, “Gibbs random fields, cooccurrences, and texture modeling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 24–37, 1994.

[22] R. Azencott, J.-P. Wang, and L. Younes, “Texture classification using windowed fourier filters,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 148–153, 1997.

[23] T. Chang and C. Kuo, “Texture analysis and classification with tree-structured wavelet transform,” *IEEE Transactions on Image Processing*, vol. 2, no. 4, pp. 429–441, 1993.

[24] K. Jafari-Khouzani and H. Soltanian-Zadeh, “Rotation-invariant multiresolution texture analysis using radon and wavelet transforms,” *IEEE Transactions on Image Processing*, vol. 14, no. 6, pp. 783–795, 2005.

[25] X. Chu and K. Chan, “Rotation and scale invariant texture analysis with tunable gabor filter banks,” in *Advances in Image and Video Technology*, ser. Lecture Notes in Computer Science, T. Wada, F. Huang, and S. Lin, Eds., vol. 5414, Nanyang Technological University School of Electrical and Electronics Engineering Singapore 639798. Springer Berlin / Heidelberg, 2009, pp. 83–93.

[26] T. Leung and J. Malik, “Representing and recognizing the visual appearance of materials using three-dimensional textons,” *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, 2001.

[27] O. G. Cula and K. J. Dana, “3d texture recognition using bidirectional feature histograms,” *International Journal of Computer Vision*, vol. 59, no. 1, pp. 33–60, 2004.

[28] M. Varma and A. Zisserman, “A statistical approach to texture classification from single images,” *International Journal of Computer Vision*, vol. 62, no. 1, pp. 61–81, 2005.

[29] ——, “A statistical approach to material classification using image patch exemplars,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 2032–2047, 2009.

[30] Y. Guo, G. Zhao, M. Pietikäinen, and Z. Xu, “Descriptor learning based on fisher separation criterion for texture classification,” in *Computer Vision ACCV 2010*, ser. Lecture Notes in Computer Science, R. Kimmel, R. Klette, and A. Sugimoto, Eds., vol. 6494, Machine Vision Group, Department of Electrical and Information Engineering, University of Oulu, Finland. Springer Berlin / Heidelberg, 8-12 November 2011, pp. 185–198.

[31] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllonen, and S. Huovinen, “Outex - new framework for empirical evaluation of texture analysis algorithms,” in *Proceedings. 16th International Conference on Pattern Recognition*, vol. 1, 2002, pp. 701–706 vol.1.

[32] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink, “Reflectance and texture of real-world surfaces,” *ACM Transactions on Graphics*, vol. 18, no. 1, pp. 1–34, January 1999.