Rotation and scale invariant local binary pattern based on high order directional derivatives for texture classification

Feiniu Yuan

School of Information Technology, Jiangxi University of Finance and Economics, Nanchang 330032, Jiangxi, China

ARTICLE INFO

Article history:
Available online 18 December 2013

Keywords:
Local binary pattern
Directional derivatives
High order
Scale space

ABSTRACT

Local Binary Pattern (LBP) only encodes the first order directional derivatives of a center pixel but it does not consider higher order derivatives. This paper proposes a rotation and scale invariant local binary pattern by jointly taking into account high order directional derivatives, circular shift sub-uniform, and scale space. Each order directional derivatives are independently encoded in a similar way of the first order derivatives to generate a code for the center pixel. Different order derivatives produce different codes that result in several histograms over an image, and then all the histograms multiplied by weights are concatenated together to fully utilize information of different order derivatives. To further improve performance, circular shift sub-uniform and scale space techniques are used to obtain rotation and scale invariant local binary patterns. Extensive experiments show that the high order derivatives based LBP can achieve good performance and obviously outperforms existing methods.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

Texture feature extraction plays an important role in many applications, such as texture retrieval [1], face detection [2], pedestrian detection [3–6], smoke detection [7,8]. There are many methods proposed for texture features, such as local binary patterns [9,10], co-occurrence matrices [11], Gabor filter and wavelet transform methods [12]. These methods can be roughly categorized into several classes, i.e. statistical, structural, and model based methods.

Ojala et al. [10] first proposed Local Binary Pattern (LBP). It is an efficient gray-scale texture operator, which can capture spatial characteristics of images. LBP features already have demonstrated powerful discriminative capability, low computational complexity, and low sensitivity to illumination variation. To further improve the discriminative capability of LBP, there are many variants of LBP that were recently proposed. A Local Ternary Pattern (LTP) [13] quantizes the gray-level differences of the neighboring pixels into a ternary value instead of a binary one. An Elongated Local Binary Pattern (ELBP) [14] uses an elliptic neighborhood instead of a circular one. An Improved Local Binary Pattern (ILBP) [15] compares the neighborhood pixels’ intensity values with the local mean pixel intensity instead of the central pixel intensity. Nanni et al. [16] proposed Elongated Ternary Patterns and Improved Local Ternary Patterns. Guo et al. [17] proposed the Local Binary Pattern Variance (LBPV). This operator can be regarded as the integral projection along the variance axis. Qian et al. [18] proposed to extend local binary pattern to pyramid transform domain (PLBP). Liao et al. [19] proposed dominant local binary patterns for texture classification by regarding the first 80% most frequent patterns as dominant features. Jun et al. [20] proposed a Compact LBP (CLBP) by maximizing mutual information between features and class labels. The CLBP provides better classification performances with smaller number of codes. Zhang et al. [21] proposed Local Derivative Pattern (LDP) for face recognition. The n-th order LDP is generated by encoding the spatial variations of the (n − 1)-th LDP. Because there are a lot of pre-specific directions that can be used to estimate the variations of the (n − 1)-th order local derivatives, LDP may result in too many codes for each pixel. Li et al. [22] proposed scale and rotation invariant local binary pattern by dividing the uniform pattern of original LBP into sub-uniform patterns and then circular aligning the histogram of each sub-uniform patterns according to the dominant bin with the maximum value. Vu and Caplier [23] proposed Patterns of Oriented Edge Magnitudes (POEM). Then they optimized the parameters of POEM and applied the whitened principal–component–analysis dimensionality reduction technique to get a more compact, robust, and discriminative descriptor [24]. Murala et al. [25] proposed local tetra patterns (LTrPs) for content-based image retrieval (CBIR). The proposed method encodes the relationship between the referenced pixel and its neighbors, based on the directions that are calculated using the first order derivatives in vertical and horizontal directions.

In fact, the original LBP only encodes the first order directional derivatives of each pixel, hence important information of higher order derivatives around the pixel is lost. This paper proposes a high order Derivative Local Binary Pattern (DLBP) by encoding the...
sequential binary values of high order directional derivatives. We find that the original LBP is obtained by encoding the first order directional derivatives, which are just forward differences of pixel intensities (0-th order directional derivatives) around a center pixel. Similarly, the $n$-th order directional derivatives are computed from forward differences of the $(n-1)$-th order directional derivatives, and the $n$-th order directional derivatives are similarly encoded to generate a code for the center pixel. Then the weighted histograms of all codes computed from different orders are concatenated together for texture classification.

Once the number of neighbors around a pixel is specified, higher order LBPs are computed using the same encoding scheme as the original LBP. In contrast, the original LBP is a specific case of DLBP with the order equal to one. Compared with LDP [21], the $n$-th order code of DLBP in this paper is generated by directly encoding the real $n$-th order directional derivatives instead of encoding the variations of the $(n - 1)$-th order derivatives in LBP. High order derivatives in LBP are not strict or mathematical high order derivatives, but they are variations of lower order derivatives. Thus DLBP produces more compact codes than LDP, and DLBP is strictly subject to mathematical formulation of directional derivatives.

To obtain scale and rotation invariance, circular shift sub-uniform patterns [22] and scale space theory are used to propose a high order Derivative Local Binary Pattern based on Circular shift sub-uniform and Scale space (DLBPCS). Circular shift sub-uniform patterns are used for improvement of rotation invariance while scale space is used to achieve scale invariant property. Recently, we also find our method is similar to the Local Directional Derivative Pattern (LDDP) proposed by Guo et al. [26]. Compared with LDDP, our method has two different contributions. One is that our method is strictly deduced from mathematical theory of signal Taylor expansion. It is another view to see encoding of higher order directional derivatives. Another is that circular shift sub-uniform and scale space are jointly used to further improve discriminative capability of high order LBPs.

This paper has two main contributions. The first contribution of this paper is to propose a high order Derivative Local Binary Pattern (DLBP) by jointly combining information of first and higher order derivatives together, and then to use circular shift sub-uniform and scale space to obtain rotation and scale invariance. The code of DLBP is strictly generated according to the mathematical definition of directional derivatives. To further improve the discriminative performance of DLBP, circular shift sub-uniform and scale space are used to propose a high order DLBP based on Circular shift sub-uniform and Scale space (DLBPCS). DLBPCS jointly combines information of different order derivatives, sub-uniform and scale space together to achieve rotation and scale invariance. Experiments show that DLBPCS greatly outperforms other methods. The second contribution is to theoretically analyze the original LBP from the Taylor expansion of a signal along a direction and then derive the proposed DLBP and DLBPCS. In fact, the original LBP only encodes the forward difference of first order directional derivatives for a center pixel. Similarly, an $n$-th order directional derivative is computed by the forward difference of its $(n - 1)$-th order directional derivatives. Thus, we have uniform and strict formulation for high order LBPs.

This paper is organized as follows. Section 2 describes local binary patterns and directional derivatives, and analyzes the relationship of them. Section 3 presents the derivative local binary pattern based on circular shift sub-uniform and scale space. Section 4 describes the texture classification method for performance evaluation of the proposed method. Experiments are described in Section 5. Finally, Section 6 concludes this paper.

2. Local binary patterns and directional derivatives

In the following sub-sections, starting from original LBP and the Taylor expansion of a signal, we derive the high order Derivative Local Binary Pattern (DLBP).

2.1. Local binary pattern and mapped patterns

An LBP pixel code is computed by comparing the value of a center pixel $g_c$ with the values of its neighboring pixels $g_i$ (Fig. 1). The LBP code for the center pixel is computed as follows:

$\text{LBP}_{P,R} = \sum_{i=0}^{P-1} s(g_i - g_c) \cdot 2^i$  \hspace{1cm} (1)

where $g_c$ is the gray scale value of the center pixel, $g_i$ is the value of its $i$-th neighbor, $P$ is the number of neighbors, $R$ is the radius of the circular neighborhood that is the Euclidean distance between the center pixel and its neighbors, and $s(x)$ is a binarization function that is defined as follows:

$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases}$  \hspace{1cm} (2)

Ojala et al. [10] defined three mapped patterns, which are uniform, rotation invariant and rotation invariant uniform ones, respectively. The uniform value of an LBP pattern, which is actually the number of circular spatial transition (bitwise 0/1 or 1/0 changes), can be mathematically computed by

$\text{U(LBP}_{P,R}) = \sum_{i=0}^{P-1} |s((g_{(i+1) \mod P} - g_c) - s(g_i - g_c))|$  \hspace{1cm} (3)

where $a \mod b$ denotes the modulo operation, which means that the number $a$ is divided by the number $b$ and the result returns only the remainder.

The uniform pattern is defined as such a pattern that has no more than 2 spatial transitions, i.e. $U \leq 2$. Experiments show that uniform patterns are one of the fundamental patterns within image textures. Uniform patterns have $P \times (P - 1) + 3$ different output values [17]. Mapping from an original pattern $\text{LBP}_{P,R}$ to a uniform pattern $\text{LBPU}_{P,R}$ can be efficiently implemented with a lookup table of $2^P$ elements.

A rotation invariant version of an original pattern, which is called rotation invariant pattern $\text{LBPRI}_{P,R}$, is defined as

$\text{LBPRI}_{P,R} = \min_{0 \leq i < P - 1} \{\text{RO}R(\text{LBP}_{P,R},i)\}$  \hspace{1cm} (4)

where $\text{RO}R(x,i)$ denotes a rotated bit-wise right shift on the number $x$ $i$ times.

Rotation invariant and uniform patterns are combined together to produce rotation invariant uniform pattern $\text{LBPU}^{\text{RIU}}_{P,R}$, which is defined as

$\text{LBPU}^{\text{RIU}}_{P,R} = \begin{cases} \sum_{i=0}^{P-1} s(g_i - g_c) & \text{if } U(\text{LBP}_{P,R}) \leq 2 \\ \frac{P}{P+1} & \text{else} \end{cases}$  \hspace{1cm} (5)
Taylor signal expansion theory, the directional sampled signal of the pixel by the weighted sum of the binary values of the one dimensional signal will simplify computation, but loss of high order information leads to tives. The LBP code of the n-th order code is bina-

\[g_i(u) = g_i(0) + g_i^{(1)}(0) \cdot u + \frac{1}{2!} g_i^{(2)}(0) \cdot u^2 + \cdots\]  

(6)

where \(g_i^{(n)}(0)\) stands for the n-th order derivative evaluated at 0. Obviously, \(g_i(0) = g_c\) and \(g_i(R) = g_i\).

The original LBP obtains illumination invariance by ignoring \(g_i(0)\). But directional derivatives with orders greater than 1 are discarded. Only the first order directional derivative \(g_i^{(1)}(0)\) is bina-

rized and used for encoding. Discarding high order derivatives can simplify computation, but loss of high order information leads to reduction of classification accuracy. Since the first order derivatives cannot fully characterize a complex signal, more order derivatives should be included.

2.3. High order directional derivative based LBP

We encode the number P of the n-th order directional derivatives \(g_i^{(n)}(0)\), \(i = 0, \ldots, P - 1\), in the same way of the first derivatives. The LBP code of the n-th order directional derivatives for a pixel can be computed as follows:

\[\text{LBP}^{n}_{P,R} = \sum_{i=0}^{P-1} s(g_i^{(n)}(0)) \cdot 2^i\]  

(7)

When \(n = 1\), \(\text{LBP}^{1}_{P,R}\) is just the same as the original LBP defined in Eq. (1). If we want to compute N number of orders, there are N codes for the pixel, i.e. \(\text{LBP}^{1}_{P,R}, \text{LBP}^{2}_{P,R}, \ldots, \text{LBP}^{N}_{P,R}\). All these codes can be also be further mapped into uniform, rotation invariant and

rotation invariant uniform patterns, which are denoted as \(\text{LBP}^{n,ui}_{P,R}\), \(\text{LBP}^{n,ri}_{P,R}\) and \(\text{LBP}^{n,riu}_{P,R}\), respectively. Each code of order \(n\) can produce a histogram over an image. N order codes produce N histograms. To combine the information of different order codes, we concatenate the N histograms together to form a robust feature vector for texture classification. Thus we can reduce histogram bins.

2.4. Discretization of directional derivatives

Comparing \(g_i^{(1)}(0)\) to the definition of LBP in Eq. (1), we find that the directional derivative \(g_i^{(n)}(0)\) is just the same as the forward difference of pixel intensity at 0 with the resampling interval \(R\). Similarly, the n-th order derivatives can also be computed by the forward difference of \((n-1)\)-th order derivatives with the interval \(R\).

As shown in Fig. 2, we take the case of the second order directional derivative as an example to demonstrate how to compute directional derivatives. The number of neighbors and the radius of the circular neighborhood are \(P\) and \(R\), respectively. Obviously, \(P\) is also the number of directions. For each direction \(i\) \((i = 0, \ldots, P-1)\), we create a local one dimensional coordinate system and the system origin is located at the center pixel \(g_c\), the pixel intensity along the direction \(i\) can be expressed as an one dimensional signal \(g_i(u)\). The first order directional derivatives at 0 and \(R\) are computed by the forward difference of the intensity:

\[g_i^{(1)}(0) = g_i(R) - g_i(0)\]  

(8)

\[g_i^{(1)}(R) = g_i(2R) - g_i(R)\]  

(9)

Similarly, the second derivative at 0 is also computed by the forward difference of the first order derivatives:

\[g_i^{(2)}(0) = g_i^{(1)}(R) - g_i^{(1)}(0) = g_i(0) + g_i(2R) - 2 g_i(R)\]  

(10)

As we can see, the computation of the second derivative requires two circular neighboring sets, whose radii are \(R\) and \(2R\), respectively. Therefore, \(n\)-order derivatives need \(n\) circular neighboring sets whose radii are \(R, 2R, \ldots, nR\), respectively.

The sign changes of the first order derivatives of a signal represent the existence of a stationary point. The binary value of first order derivatives can capture the ascending or descending property of a signal. So original LBP codes reflect information about stationary points and monotonicity of pixel intensities. The sign changes of the second derivatives of a signal are used to find an inflection point, which is the changing point of convexity and concavity of a signal. So the binary value of second order derivatives can capture the information about the convex and concave property of the signal.

The circular neighborhood requires interpolation of pixel intensity. To improve computation efficiency, we use rectangular neighbor-

hood instead in our implementation. As shown in Fig. 3, the neighbor number of each rectangular set is 8 and the radius of the

![Fig. 2. Circular neighborhood for LBP patterns with second order directional deriva-
tives.](image2)

![Fig. 3. Rectangular neighborhood for LBP patterns with second order directional derivatives \((P = 8, R = 1)\).](image3)
minimum rectangle is 1, i.e., \( P = 8 \) and \( R = 1 \). Given 2 rectangular sets, according to Eqs. (7), (8) and (10), we can compute two LBP codes with orders less than 2, i.e. \( \text{LBP}^1_{P,R} \) and \( \text{LBP}^2_{P,R} \), as follows:

\[
\text{LBP}^1_{P,R} = \sum_{i=0}^{P-1} s(g_i - g_c) \cdot 2^i
\]

\[
\text{LBP}^2_{P,R} = \sum_{i=0}^{P-1} s(g_i + g_{i+2} - 2g_{i+1}) \cdot 2^i
\]

As we can see, the intervals of different directions are variable but the interval along each direction is always fixed. For example, the interval of a direction \( i = 1 \) is \( \sqrt{2} \) while the interval along another direction \( i = 2 \) is 1. We encode the binary value of derivatives, so varying intervals do not influence final results much.

Fig. 4 shows an example of high order derivative LBP patterns. The center pixel value \( g_c \) is equal to 86, the number of neighbors in each set is 8 and the minimum radius of the neighborhood is 1. So the first and second order LBP pattern codes are 00001111 and 00110000, respectively. In addition, we can also compute the corresponding uniform, rotation invariant and rotation invariant uniform codes according to the method of [10].

3. Derivative local binary pattern based on circular shift sub-uniform and scale space

Li et al. [22] proposed the concept of circular shift sub-uniform pattern and used the circular shift of histogram to obtain rotation invariance. They also computed sub-uniform LBP in scale space to obtain scale invariance. To further improve DLBP, we also used sub-uniform and scale space techniques to propose a high order Derivative Local Binary Pattern on Circular shift sub-uniform (DLBPC) and a high order Derivative Local Binary Pattern based on Circular shift sub-uniform and Scale space (DLBPCS). Experiments validated that DLBPCS is more robust than existing methods.

3.1. DLBP on circular shift sub-uniform

A sub-uniform pattern of an \( n \)-th order DLBP \( \text{LBP}^n_{P,R} \) is defined as follows:

\[
\text{LBP}^n_{P,R} = \begin{cases} 
M & \text{if } U(\text{LBP}^2_{P,R}) = 0 \\
(M, W) & \text{if } U(\text{LBP}^2_{P,R}) = 2 \\
P + 1 & \text{otherwise}
\end{cases}
\]

where \( n \) is the order of derivatives to be encoded, \( su2 \) denotes the sub-uniform patterns with spatial transition (\( U \leq 2 \)) and \( M \) is the uniform pattern index, which is just the number of bit “1” contained in the code \( \text{LBP}^2_{P,R} \). \( M \) is defined as follows:

\[
M = \sum_{i=0}^{P-1} s(g_i^{(n)}(0))
\]

\( W \) actually denotes the angular position index of a sub-uniform pattern, which is defined as the index of 0/1 transition. \( W \) can be mathematically defined as follows.

\[
W = \arg \max_{i=0, \ldots, P-1} \left\{ b_i \left( (\text{LBP}^2_{P,R}) \cdot b_i \left( \text{LBP}^n_{P,R} \right) \right) \right\}
\]

where \( b_i(c) \) is the \( i \)-th bit of a binary code \( c \) and \( b_i(c) \) denotes the inverse value of the \( i \)-th bit of \( c \).

Given parameters \( P \) and \( R \), the sub-uniform pattern index is within the range (\( M = 1, \ldots, P - 1 \)) and the position index is within the range (\( W = 0, \ldots, P - 1 \)). So there are \( P - 1 \) sub-uniform patterns, resulting in \( P - 1 \) sub-histograms, each of which has \( P \) bins. For each sub-histogram, the sub-uniform pattern \( M \) with the maximum statistical value is defined as the dominant orientation sub-uniform pattern. Then the dominant bin of the sub-histogram is circularly shifted to the 0-th bin of the sub-histogram. Thus the circular shift sub-uniform histogram is generated to obtain rotation invariance. In this way, we obtain the DLBP based on circular shift sub-uniform (DLBPC).

3.2. DLBP based on circular shift sub-uniform and scale space

Scale space theory [27] is a framework for multi-scale signal analysis developed by computer vision, image processing and signal processing communities. The scale space representation \( L \) of a 2D image \( f \) can be obtained by convoluting \( f \) with a Gaussian kernel \( k \):

\[
L(x, y, t) = k(x, y, t) * f(x, y)
\]

where

\[
k(x, y, t) = \frac{1}{2\pi \sigma^2} e^{-(x^2+y^2)/(2\sigma^2)}
\]

in which \( \sigma \) is the variance parameter of the Gaussian kernel \( k \).

As \( \sigma \) increases, \( L(x, y, t) \) is a more smoothing version of \( f(x, y) \), so more details of the function are removed. When the standard deviation \( \sigma \) is large, the width of the Gaussian template \( W_C = (\sqrt{2} \times 2.5 \times 2 + 1) \) becomes large too, so the size \( W_C \times W_C \) of the Gaussian template becomes very large. Convoluting an image with the large Gaussian template is time-consuming. Alternatively, we can use 2D Fast Fourier Transform (2D FFT) to transform the image and the Gaussian kernel into the Fourier domain, and multiply their frequency components pixel by pixel, and then transform the multiplied result back into the spatial domain. Thus we can obtain the smoothed image in constant time for any standard deviation \( \sqrt{2} \). Nowadays, consumer-level GPUs have very powerful computation compared with CPUs. For example, a GPU of Nvidia Gefore GTX 650 has 384 kernels while a CPU usually has less than 8 kernels. To further accelerate the construction of the image scale space, GPU can totally take over time-consuming computation of 2D FFT from CPU, and thus the construction speed is extremely improved.

Fig. 5 shows the scale space representation of an image. The image is down loaded from the Brodatz texture database [28], and cropped to the size of 128 \( \times \) 128.

The normalized Laplacian operator is applied to the scale space representation \( L(x, y, t) \) of the image \( f(x, y) \). For each pixel, we find the scale at which the local maximum response is reached over scale, and the scale is regarded as the optimal scale \( r(x, y) \) of the pixel [22]. The radius of the neighborhood is set to the optimal scale \( r(x, y) \), and then we compute the DLBP based on circular shift sub-uniform and scale space with \( P \) and \( r(x, y) \). In other words, the radius \( r(x, y) \) of LBP is varying across the image while the neighbor number \( P \) is always fixed for all pixels. Thus we generate a high order Derivative Local Binary Pattern based on Circular shift sub-uniform and Scale space (DLBPCS).
LBP codes based on the i-derivative derivatives, then shift sub-uniform and scale space, resulting in a histogram over Q to measure the dissimilarity of the histograms between a query and a target texture. To evaluate the feature discriminative capability of derivative learning procedure, in addition, the complex learning procedure classification performance, but these methods require a complex learning procedure. In summary, we have eleven LBP methods to perform extensive experiments. Table 1 summarizes all the LBP methods with detailed description, references and feature dimensions. Our method can assign a weight to each histogram and then concatenate all the histograms together. The classification process can be expressed by the following equation:

\[ J = \arg\min_{n=0,1,\ldots,N-1} \left\{ D(Q, T_n) \right\} \]

where \( B \) is the bin number of a feature histogram, and \( Q^i \) and \( T_n^i \) are the values at the i-th bins of the histograms of the query image \( Q \) and the target image \( T_n \), respectively.

Given a target texture set \( \{T_0, T_1, \ldots, T_{N-1}\} \), a query image \( Q \) is classified into the \( j \)-th target image \( T_j \) that has the minimum dissimilarity to \( Q \). The classification process can be expressed by the following equation:

\[ D(Q, T_n) = \frac{1}{B} \sum_{i=0}^{B-1} |Q^i - T_n^i| \]

4. Histogram concatenation and classification method

4.1. Weighted concatenation of histograms

Each order derivative produces an LBP code based on circular shift sub-uniform and scale space, resulting in a histogram over an image. Hence, if we want to extract LBP codes with \( M \)-th order derivatives, then \( M \) histograms are produced. To obtain a final histogram, we can assign a weight to each histogram and then concatenate them together. So the weighted concatenation of \( M \) histograms can be formulated as follows:

\[ H = \{\alpha_i H_1, \ldots, \alpha_M H_M\} \quad (18) \]

where \( H_i \) (\( i = 1, \ldots, M \)) is the i-th histogram calculated from the LBP codes based on the i-th order derivatives, \( \alpha_i \) denotes the weighting coefficient for the i-th histogram, and \( M \) is the maximum order of derivatives. For the sake of simplicity, we can set all the weighting coefficients to 1, and then the weighted concatenation of histograms become the normal concatenation of histograms. We also notice that higher order derivatives are more sensitive to noise, so a small weight can be assigned to higher order histograms.

4.2. Classification method

Some classification methods, such as artificial neural network [9], SVM [29] and AdaBoost [2], can achieve outstanding classification performance, but these methods require a complex learning procedure. In addition, the complex learning procedure may influence analysis of discriminative capabilities of features. To evaluate the feature discriminative capability of derivative LBPs, a simple classification method is used to minimize the influence of classification methods. So we use Mean Absolute Distance (MAD) to measure the dissimilarity of the histograms between a query texture \( Q \) and a target texture \( T_n \), defined as follows:

\[ D(Q, T_n) = \frac{1}{B} \sum_{i=0}^{B-1} |Q^i - T_n^i| \]

where \( B \) is the bin number of a feature histogram, and \( Q^i \) and \( T_n^i \) are the values at the i-th bins of the histograms of the query image \( Q \) and the target image \( T_n \), respectively.

Given a target texture set \( \{T_0, T_1, \ldots, T_{N-1}\} \), a query image \( Q \) is classified into the \( j \)-th target image \( T_j \) that has the minimum dissimilarity to \( Q \). The classification process can be expressed by the following equation:

\[ J = \arg\min_{n=0,1,\ldots,N-1} \left\{ D(Q, T_n) \right\} \]

5. Experiments

5.1. Implementation

We used C++ to implement the high order Derivative Local Binary Pattern based on Circular shift sub-uniform (DLBPC), and the Derivative Local Binary Pattern based on Circular shift sub-uniform and Scale space (DLBPCS). For comparison purposes, we implemented original LBP methods and several variants of LBP. The original LBP with uniform, rotation invariant and rotation invariant uniform patterns [10] are respectively denoted as LBP U2, LBP RI and LBP RIU2. We implemented an LBP method based on circular shift sub-uniform, and one based on circular shift sub-uniform and scale space [22], denoted as LBPC and LBPCS, respectively. Also, the local derivative pattern (LDP) [21] was implemented and the second order derivatives along the four directions (0°, 45°, 90° and 135°) were computed to produce four LDP codes with second order derivatives, so there are 32 bits for each pixel. To reduce histogram bins, we individually computed the uniform histogram of each code and then concatenated all the histograms together. The method is denoted as LDP4 U2. So far, we have eight LBP methods for experiments.

To compare our method with recently proposed methods, three recently proposed methods were selected and implemented using visual C++ for comparisons. The Patterns of Oriented Edge Magnitudes (POEM) [24] was implemented and optimized parameters were used for comparison experiments. The cell size, block size, orientation number, and neighbor number of POEM are set to 7, 10, 3 and 6, respectively. We also implemented Local Tetra Patterns (LTrP) presented in [25]. There are 13 binary pattern codes generated for each pixel. In addition, we also implemented the local directional derivative pattern (LDDP) proposed by Guo et al. [26]. Only first and second order LBPs are used and their histograms are concatenated together for comparison.

In summary, we have eleven LBP methods to perform extensive experiments. Table 1 summarizes all the LBP methods with detailed description, references and feature dimensions.
extracts 118 dimensional feature vectors. LTrP generates longest dimensions of features. To clearly see comparison results, the three recent LBP methods, i.e. LTrP, POEM and LDDP, will be independently compared with our DLBPCS, while other eight LBP methods, i.e. DLBPC, DLBPCS, LBPC, LBPCS, LBP U2, LBP RI, LBP U2RI, and LDP4 U2, are compared.

For fair comparisons, the eleven implemented LBP methods are set to same parameters, i.e. the number \( P \) of neighbors and the radius \( R \) of the neighborhood are respectively set to 8 and 1, except for POEM. As for two scale space based methods, the standard deviation \( \sqrt{t} \) of the Gaussian kernel is set to 1, 2, ..., 16. For simplicity, the highest order of directional derivatives is set to 2. All the experiments were conducted on a PC equipped with an AMD Phenom X4 955 CPU, 3.2 GHz and 8 G RAM for evaluation of computation efficiency.

5.2. Brodatz texture database

We randomly selected thirty texture images from the Brodatz texture database [28]. One of the selected thirty images has the size of 643 \( \times \) 643, and others 640 \( \times \) 640. A patch with the size of 128 \( \times \) 128 is extracted from each selected image and regarded as the target texture of the image. Thus we generated a Brodatz target set with 30 textures, as shown in Fig. 6. As we can see, the Brodatz target set has more periodic textures than non-periodic ones.

All the selected textures were rotated in a step of 15° from 0° to 180°, so there are 13 rotated versions of each image. Then four non-overlapping patches with the size of 128 \( \times \) 128 were extracted from each rotated image. The four patches are used as query textures of the image. So we have generated a rotated Brodatz query set with \( 30 \times 13 \times 4 = 1560 \) images. Fig. 7 shows 13 rotated images of a patch.

We tested the eight LBP methods and the three recent LBP methods on the rotated Brodatz query set. Fig. 8 illustrates the chart of detection rates of the eight LBP methods with respect to rotation angles. We do not put all the curves of the eleven LBP methods together in order to clearly see the curve of each method.

As we can see, high detection rates are obtained around the rotation angles 0°, 90° and 180° for all the LBP methods. This is because the image distortion at the three angles is minimized. From Fig. 8, we can easily see that our DLBPCS has the best classification performance among the eight methods for most of rotation angles.

Table 2 lists average detection rates and overall computation time of the eleven LBP methods without considering rotation angles.
angles. Our DLBPCS and DLBPC achieved average detection rates of 72.0% and 40.6%, which are ranked in the first and third position, respectively. Due to usage of sub-uniform pattern and scale space, LBPCS also obtained an average rate of 51.2%. Since we have 30 classes, so the average guess rate is $\frac{1}{30} = 3.3\%$. The highest detection rate of 72.0% is far greater than 3.3%. So our DLBPCS performs excellently on the rotated Brodatz query set. Although LDP4 U2 adopted four kinds of second order derivatives, the detection rate is lower than others. The reason is that high order derivatives are sensitive to noise. So only usage of higher order derivatives can not obtain good performance. Therefore, our methods utilize first and second order derivatives at the same time.

We also wrote code to record the timings of texture classification on the rotated Brodatz set. The time includes reading the 1560 images from the hard disk, extracting LBP features and classifying each image. As we can see from Table 2, both DLBPCS and LBPCS consume about 100 s, and the two methods are two of most time-consuming methods due to time-consuming process of scale space. But our DLBPCS greatly outperforms LBPCS. The original LBP is fastest among them. The scale space based methods are almost 20 times slower than others. Better performance may sacrifice computation efficiency, so we need to make a trade off between efficiency and accuracy in time-critical applications, e.g. video smoke detection [8].

To validate classification performance on scale, all the selected textures were scaled at coefficients from 0.5 to 1.7 with a step of 0.1, so there are also 13 scaled versions of each image. Similarly, four non-overlapping patches with the size of $128 \times 128$ were also extracted from each scaled image. So we obtained a scaled Brodatz query set with $30 \times 13 \times 4 = 1560$ images. The reason of the same texture number as the rotated set is to facilitate comparisons and programming. Fig. 9 shows 13 scaled images of a patch.

We tested the eight LBP methods and the recent three LBP methods on the scaled Brodatz query set. For the same reason, Fig. 10 shows detection rates of the eight LBP methods only with respect to scale coefficients. The highest detection rate is achieved at the scale coefficient 1.0 for all the LBP methods because the image scaled at the coefficient 1.0 is mostly similar to the target texture. From Fig. 10, we can easily see that our DLBPCS has the best classification performance among the eight methods due to the combination of first and second order information, sub-uniform and scale space.

Table 2 lists average correct classification rates of the eleven LBP methods without considering scale coefficients. Our DLBPCS also have the best correct classification rate of 75.2% among them. Due to usage of scale space, LBPCS has a good average detection rate of 58.7%. The original LBP with uniform also has good detection rate of 44.6%. The timings of the eleven LBP methods on the scaled set is almost the same as ones on the rotated set, so we do not list timings again.

### Table 2

<table>
<thead>
<tr>
<th>Methods</th>
<th>Average detection rate (%)</th>
<th>Average computation time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Brodatz texture database</td>
<td>Outex texture database</td>
</tr>
<tr>
<td></td>
<td>Rotate</td>
<td>Scale</td>
</tr>
<tr>
<td>DLBPC</td>
<td>5.0 s</td>
<td>40.6%</td>
</tr>
<tr>
<td>DLBPCS</td>
<td>108.8 s</td>
<td>72.0%</td>
</tr>
<tr>
<td>LBPCS</td>
<td>3.9 s</td>
<td>35.0%</td>
</tr>
<tr>
<td>LPCS</td>
<td>101.7 s</td>
<td>51.2%</td>
</tr>
<tr>
<td>LBP U2</td>
<td>3.6 s</td>
<td>27.4%</td>
</tr>
<tr>
<td>LBP RI</td>
<td>3.6 s</td>
<td>27.8%</td>
</tr>
<tr>
<td>LBP U2RI</td>
<td>3.6 s</td>
<td>27.3%</td>
</tr>
<tr>
<td>LDP4 U2</td>
<td>8.7 s</td>
<td>24.6%</td>
</tr>
<tr>
<td>POEM</td>
<td>16.0 s</td>
<td>34.6%</td>
</tr>
<tr>
<td>LDDP</td>
<td>9.5 s</td>
<td>33.6%</td>
</tr>
<tr>
<td>LTrP</td>
<td>7.1 s</td>
<td>19.0%</td>
</tr>
</tbody>
</table>

![Fig. 8. Texture classification using the eight LBP methods on the rotated Brodatz query set.](image8)

![Fig. 9. Scaled textures from the Brodatz texture database. Scale coefficients are 0.5, 0.6, ..., 1.7, respectively.](image9)

![Fig. 10. Texture classification using eight variants of LBP on the scaled Brodatz query set.](image10)
Fig. 11. Texture classification using DLBPCS, POEM, LDDP and LTrP on the rotated and scaled Brodatz query sets.

Fig. 11(a) and Fig. 11(b) show experimental results of the three recent LBP methods and our DLBPCS on the rotated and scaled Brodatz query sets. As shown in Fig. 11(a), our DLBPCS greatly outperforms LTrP, POEM and LDDP for most of rotated samples, but LDDP obtained the highest detection rate at the rotation angle 0°. Similarly, our DLBPCS obviously outperforms LTrP, POEM and LDDP for most of scale coefficients, LDDP also obtained the highest detection rate at the scale coefficient 1.0, and LTrP has the lowest detection rate for the two sets, as shown in Fig. 11(b). These experiments show that our DLBPCS has better discriminative capability than LTrP, POEM and LDDP with respect to rotation and scale transforms.

5.3. Outex texture database

We randomly selected thirty texture images from the Outex texture database [30] at the parameters of an illumination of “Inca”, a resolution of 300 dpi and a rotation angle of 0°. The size of all the selected thirty images is 746 × 538. A patch with the size of 128 × 128 is extracted from each selected image and the patch is regarded as the target texture of the image. Thus we generated
an Outex target set with 30 textures, as shown in Fig. 12. We can see that there are more non-periodic textures in the selected Outex images than ones in the selected Brodatz images.

We generated rotated query textures in a similar way described in the above section. All the selected textures were rotated in a step of 15° from 0° to 180°, and four non-overlapping patches with the size of 128×128 were extracted from each rotated image, so we generated a rotated Outex query set with 30×13×4 = 1560 images.

Similarly, all the selected textures were scaled at coefficients from 0.5 to 1.7 with a step of 0.1, and four non-overlapping patches with the size of 128×128 were also extracted from each scaled image, so we obtained a scaled Outex query set with 30×13×4×4 = 1560 images.

First, we tested the eight LBP methods and the three recent LBP methods on the rotated Outex query set. Fig. 13 shows the chart of correct classification rate with respect to rotation angles for clear comparison. As we can see, high detection rates are also obtained at the rotation angles 0°, 90° and 180° for all the LBP methods because the image distortion at the three angles is minimal. From Fig. 13, we can see that our DLBPCS has the highest correct classification rate among the eight methods due to the combined usage of high order derivatives, sub-uniform and scale space.

Second, we tested the eleven LBP feature extraction methods on the scaled Outex query set. Fig. 14 shows the chart of classification rate with respect to scale coefficients. Also, the highest detection rate is achieved at the scale coefficient 1.0 for all the LBP methods because the image scaled at the coefficient 1.0 is mostly similar to the target texture. From Fig. 14, we can easily see that our DLBPCS is the best classification method among the eight methods due to the integration of high order derivative, sub-uniform and scale space.

Table 2 lists the average detection rate of the eleven LBP methods without considering rotation angles and scale coefficients. For the rotated Outex query set, our DLBPCS achieves the best correct classification rate of 58.4% among them. As for the scaled Outex query set, our DLBPCS obtains the best detection rate of 53.3%, which is also highest among all the methods. LDP4 U2 achieved the correct rate of 30.3% that is close to the rate of 30.9% by DLBPC, but LDP4 U2 uses 4 codes while DLBPC only has 2 codes for each pixel. Of course, higher order derivatives are more sensitive to noise, but higher order derivatives are important. Experiments show that our method outperforms existing methods. All the above experimental results validate that higher order derivatives can improve classification accuracy.

Fig. 15 shows experimental results of the three recent LBP methods and our method on the rotated and scaled Outex query sets. As shown in Fig. 15(a), our DLBPCS greatly outperforms LTrP, POEM and LDDP for most of rotated samples, but LDDP also obtained the highest detection rate at the rotation angle 0°. Similarly, our DLBPCS obviously outperforms LTrP, POEM and LDDP for most of scaled samples, but LDDP obtained the highest detection rate at the scale coefficient 1.0, as shown in Fig. 15(b). We can see that LTrP also has the lowest detection rate for the two Outex query sets. These experiments show that our DLBPCS has better discriminative capability than LTrP, POEM and LDDP with respect to rotation and scale transforms. So our DLBPCS is suitable to detect textures when samples are obtained after performing unknown rotation and scale transforms.
Fig. 16. Texture classification using DLBPCS, POEM, LDDP and LTrP on the rotated and scaled PSU query sets.

Table 3

<table>
<thead>
<tr>
<th>Methods</th>
<th>DLBP U2</th>
<th>LBP U2</th>
<th>DLBP RI</th>
<th>LBP RI</th>
<th>DLBP RI U2</th>
<th>LBP RI U2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotated Brodatz set</td>
<td>33.6</td>
<td>27.4</td>
<td>38.8</td>
<td>27.8</td>
<td>35.6</td>
<td>27.3</td>
</tr>
<tr>
<td>Scaled Brodatz set</td>
<td>48.3</td>
<td>44.6</td>
<td>32.8</td>
<td>24.3</td>
<td>26.5</td>
<td>20.6</td>
</tr>
<tr>
<td>Rotated Outex set</td>
<td>24.4</td>
<td>19.2</td>
<td>37.2</td>
<td>23.3</td>
<td>34.2</td>
<td>22.9</td>
</tr>
<tr>
<td>Scaled Outex set</td>
<td>36.0</td>
<td>34.6</td>
<td>25.5</td>
<td>21.4</td>
<td>23.4</td>
<td>19.8</td>
</tr>
</tbody>
</table>

5.4. PSU Near-Regular Texture Database

The Near-Regular Texture Database [31] covers the spectrum of textures from completely regular to near-regular to irregular. It also includes video of near-regular textures in motion. We selected 30 texture images from the PSU Near-Regular Texture Database. A similar way of generating query sets from Brodatz and Outex databases was used to generate a PSU target set with 30 textures, a rotated PSU query set with 1560 images, and a scaled Brodatz query set with 1560 images.

Fig. 16(a) shows the experimental results of the three recent methods and our method on the scaled PSU set. On the whole, DLBPCS and LDDP have nearly equivalent performance on the rotated PSU set, POEM ranks slightly after the above two methods, and LTrP has the worst performance. As shown in Fig. 16(b), both DLBPCS and POEM have better performance than LDDP and LTrP on the scaled PSU set. DLBPCS is slightly better than POEM. These experiments show that POEM is not suitable for rotation while POEM has powerful discriminative capability over scale.

From Table 2, we can see that our DLBPCS has a detection rate of 40.2% ranked in the third place for the rotated PSU set. For the scale PSU set, our DLBPCS has the highest detection rate of 74.7%, and POEM has the second highest rate of 68.5%. The results show that our DLBPCS always achieves the best performance on scaled sets. This is because our DLBPCS includes scale information.

5.5. Comparison of mapped patterns of LBP and DLBP

Our Derivative Local Binary Patterns (DLBP) can also be mapped to the uniform, rotation invariant and rotation invariant uniform patterns, which are denoted as DLBP U2, DLBP RI and DLBP RI U2, respectively. To validate the importance of high order directional derivatives, we also compared these mapped patterns with the mapped patterns of original LBP, i.e. LBP U2, LBP RI and LBP RI U2.

Experiments were performed on the four sets, i.e. the rotated Brodatz query set, the scaled Brodatz query set, the rotated Outex query set and the scaled Outex query set. The results are listed in Table 3. As we can see, for any texture query set, any mapped pattern of DLBP always achieves higher detection rate than the same mapped pattern of LBP. For example, DLBP U2 has a detection rate of 33.6% while LBP U2 achieves only 27.4% on the rotated Brodatz set. For the Rotated Outex set, a detection rate of 37.2% with DLBP RI is far higher than a detection rate of 23.3%, so we obtain 59.7% improvement of detection rate by using the second order derivatives. The experimental results validate that the higher order directional derivatives can obviously improve detection rate. So it is important to take into account the first and higher order directional derivatives for improvement of accuracy.

6. Conclusions

Because original local binary patterns (LBP) only encode the first order directional derivatives of each pixel, important higher order derivative information is lost. So this paper proposes a novel local binary pattern by taking into account first and higher order directional derivatives at the same time. Directional derivatives of different orders are independently encoded to generate several codes for each pixel. Thus several histograms can be produced for an image. Then all the histograms of different orders are concatenated together for texture classification. To further improve performance, circular shift sub-uniform pattern is used to obtain rotation invariance, and scale space is used to achieve scale invariance. Each code is divided into sub-uniform patterns and the histogram of the sub-uniform patterns is circularly aligned to the maximum entry. Optimal scales are estimated for each pixel. Thus we propose high order derivative local binary patterns based on sub-uniform and scale space. Experiments show that the proposed method obviously outperforms existing methods of local binary patterns and it is suitable to detect objects with unknown rotation and scale transforms.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61363038, 61063034) and Science Technology Application Project of Jiangxi province (KJLD12066). Special thanks are given to anonymous reviewers for their helpful grammar corrections and suggestions.
References


Feiniu Yuan received the B.Eng. and M.E. degrees in mechanical engineering from the Hefei University of Technology, Hefei, China, in 1998 and 2001, respectively, and the Ph.D. degree in pattern recognition and intelligence system from the University of Science and Technology of China (USTC), Hefei, in 2004. He was a post-doctor with USTC and a Senior Research Fellow with Singapore Bioimaging Consortium, Agency for Science, Technology and Research, Singapore. He is currently a Professor with Jiangxi University of Finance and Economics. His research interests include 3D modeling, image processing and pattern recognition.