

## Local pixel patterns

Fangjun Huang<sup>1</sup> (✉), Xiaochao Qu<sup>2</sup>, Hyoung Joong Kim<sup>2</sup>, and Jiwu Huang<sup>3</sup>

© The Author(s) 2015. This article is published with open access at Springerlink.com

**Abstract** In this paper, a new class of image texture operators is proposed. We firstly determine that the number of gray levels in each  $B \times B$  sub-block is a fundamental property of the local image texture. Thus, an occurrence histogram for each  $B \times B$  sub-block can be utilized to describe the texture of the image. Moreover, using a new multi-bit plane strategy, i.e., representing the image texture with the occurrence histogram of the first one or more significant bit-planes of the input image, more powerful operators for describing the image texture can be obtained. The proposed approach is invariant to gray scale variations since the operators are, by definition, invariant under any monotonic transformation of the gray scale, and robust to rotation. They can also be used as supplementary operators to local binary patterns (LBP) to improve their capability to resist illuminance variation, surface transformations, etc.

**Keywords** texture; multi-bit; gray scale; rotation

### 1 Introduction

Image texture is a fundamental research topic in the areas of image processing, pattern recognition, computer vision, etc. It has received considerable attention during the past few decades and numerous approaches have been presented.

Image texture analysis aims to extract textural

features from images. One of the pioneering techniques was presented by Haralick et al. [1], in which the co-occurrence matrix provided textural features for image classification. In Ref. [2], Cross and Jain explored the Markov random field model for the generation and analysis of texture images. Bovik et al. [3] submitted the image to a series of filter banks, followed by use of an energy measure to describe image texture. In Ref. [4], Chang and Kuo focused on different sub-band decompositions for image texture description. All these texture operators proposed in the previous century have received much attention in the past few years. For example, according to Google Scholar search, up to December 2014 the co-occurrence operator [1], Markov operator [2], the filtering operator [3], and wavelet operator [4] had been cited 10472, 1395, 1581, and 1614 times, respectively.

In 2002, Ojala et al. [5] proposed the idea of local binary patterns (LBP). It takes a local neighborhood around each pixel, thresholds the pixels of the neighborhood at the value of the central pixel, and uses the resulting binary-valued patterns as a local image descriptor. LBP provide a simple yet efficient operator to describe local texture, and are invariant under any monotonic gray scale transformations. To achieve rotation invariance, the binary-valued pattern is circularly shifted and the minimum value is kept as the final pattern. Moreover, in order to ensure low computational complexity, Ojala et al. [5] also introduced uniform patterns (a binary-valued pattern is uniform if the number of transitions between zero and one in it is at most two). All non-uniform patterns are merged into one pattern.

LBP have also been extensively studied (according to Google Scholar, cited 5290 times up to December 2014) and have demonstrated their high discriminative power for texture classification. In

1 School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, China. E-mail: huangfj@mail.sysu.edu.cn (✉).

2 Graduate School of Information Security, Korea University, Seoul 136-701, R. O. Korea. E-mail: quxiaochao@gmail.com, khj-@korea.ac.kr.

3 College of Information Engineering, Shenzhen University, Shenzhen 518060, China. E-mail: jwhuang@szu.edu.cn.

Manuscript received: 2015-01-23; accepted: 2015-04-23

order to resolve the sensitivity to noise in near-uniform image regions, Tan and Triggs [6] extended LBP to local ternary patterns (LTP) with a 3-valued coding that includes a threshold around zero when evaluating local gray scale differences. In Ref. [7], Zhou et al. pointed out that the efficiency of LBP could be improved via combining uniform patterns with a few non-uniform patterns. A simple yet effective method for improving the performance of LBP was also presented in Ref. [8] by exploring some new information from the non-uniform patterns. Liao et al. [9] presented dominant LBP (DLBP), in which the 80% most frequently occurring patterns are experimentally chosen from all rotation invariant binary-valued patterns for texture classification. By considering local contrast information, Guo et al. [10] developed LBP variance (LBPV). In Ref. [11], Guo et al. proposed completed LBP (CLBP), in which the operator utilizes both the sign and magnitude information in the difference between the central pixel and its neighborhood pixels. Based on CLBP, a new texture operator, called completed local binary count (CLBC), was proposed [12]. By counting the number of ones in the binary neighbor sets instead of encoding them as in CLBP, comparable classification accuracy rates could also be achieved. In Ref. [13], Khellah presented a new method for texture classification, which combines dominant neighborhood structure (DNS) and traditional LBP features. Most recently, Maani et al. [14] presented a new rotation invariant method for texture classification based on local frequency components of the LBP features. Furthermore, LBP and its improved versions have received considerable attention, finding use in a wide array of fields such as face recognition [6, 15–17], video description [18], eye detection [19], and image retrieval [20, 21].

However, textures in the real world are often not uniform due to variations in orientation, scale, or other visual appearance. No current method can tackle all texture problems. Furthermore, there is no strict definition of image texture even though the topic has been widely studied in the past 50 years, dating back at least to Julesz in 1962 [26]. As a classical and challenging field, new ideas and new techniques are in great request.

In this paper, we present a new class of image texture operators with a new basis which differs

from that of LBP. In our approach, the number of different gray scales in each  $B \times B$  sub-block of the image is utilized to describe texture features. Moreover, a new multi-bit plane strategy is presented for improving the capability of our proposed operators. The extracted texture features are not only by definition invariant under any monotonic transformation of gray scale, but also robust to rotational variations. Though our new proposed may have some limitations in describing illuminance variants and surface transformations, they can be combined with LBP to form a set of more powerful texture operators, as shown by our experiments on various standard texture image databases.

The remainder of this paper is organized as follows. In Section 2 our new texture operator is introduced. Experimental results and a comparative study are given in Section 3. Finally, we conclude in Section 4.

## 2 Proposed approach

In this section, an overview of LBP will be given first and then the main weaknesses of LBP will be analyzed. Based on the analysis, we then introduce our new texture operators.

### 2.1 An overview of LBP

LBP [5] use an operator which characterizes the spatial structure of the local image texture. For a given pixel in the image, an LBP code is computed by comparing it with its neighbors as follows:

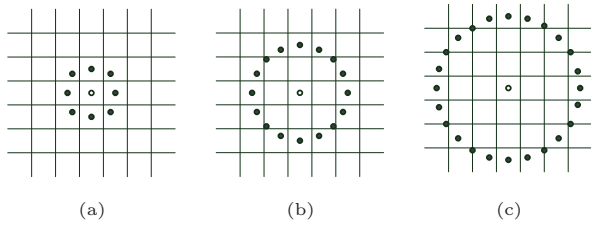
$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (1)$$

where

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2)$$

Here,  $g_c$  is the gray value of the central pixel,  $g_p$  ( $p = 0, \dots, P-1$ ) are the gray values of the neighbors on a circle of radius  $R$ , and  $P$  is the number of neighbors. Figure 1 shows examples of circularly symmetric neighbor sets for different configurations of  $(P, R)$ . As pointed out in Ref. [5], neighbors' values needed off-center of a pixel location are estimated by bilinear interpolation.

After the LBP code for each pixel (i.e., the central pixel  $g_c$ ) has been computed, a histogram can be



**Fig. 1** Circularly symmetric neighbor sets for different values of  $(P, R)$ : (a)  $P = 8, R = 1$ ; (b)  $P = 16, R = 2$ ; (c)  $P = 24, R = 3$ .

built to represent the image texture. In order to achieve rotation invariance, Ojala et al. [5] proposed the rotation invariant uniform patterns, which are computed using

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c), & U(LBP_{P,R}) \leq 2 \\ P + 1, & \text{otherwise} \end{cases} \quad (3)$$

where

$$U(LBP_{P,R}) = |s(g_{p-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \quad (4)$$

The uniformity measure  $U$  corresponds to the number of spatial transitions (bitwise 0 to 1 changes or vice versa) in the pattern. Uniform patterns refer to those patterns which have  $U$  values of at most 2, and the remaining non-uniform patterns are all merged into one pattern. After the  $LBP_{P,R}^{riu2}$  code of each pixel has been computed, a histogram is built to represent the image texture.

**2.2 Main weakness of LBP**

The main philosophy behind LBP is that the correlation between the central pixel and its neighboring pixels on the circle can be utilized to describe image texture. This is true and various experimental results have demonstrated the excellent performance of LBP series. However, since only the pixels on the circularly symmetric neighbor sets are considered, some nearer neighboring pixels of the central pixel are neglected instead. For example, when we set  $P = 3$  and  $R = 24$ , the eight nearest pixels to the central pixel are not considered by the LBP operator, as shown by the dashed area in Fig. 2 (note that some pixels outside the dashed area but inside the circle are used to estimate the pixels on the circles via bilinear interpolation, and make some contribution to the operator). Intuitively, LBP may miss some important information about the image

**Fig. 2** Pixels not considered by LBP.

texture.

However, the LBP framework cannot readily consider these missing pixels, especially with increasing  $P$  and  $R$ . According to Eq. (1), when selecting  $P = 24$  and  $R = 3$ , there may be  $2^{24}$  different binary-valued patterns for the central pixel. Because the number of patterns is huge, considering any new pixel may place a heavy burden on the computation complexity. For example, if all eight pixels in the dashed area are considered, there will now be  $2^{32}$  binary-valued patterns.

In addition, in order to reduce the feature dimension for the LBP operator, all the binary-valued patterns are divided into uniform and non-uniform patterns; all non-uniform patterns are directly merged into one pattern. According to a test conducted by Ojala et al. [5], on some image datasets, non-uniform LBP patterns for  $(P, R)$  equal to  $(8, 1)$ ,  $(16, 2)$ , and  $(24, 3)$  on average contribute 12.8%, 33.1%, and 50.7% of the total pattern data. Thus, by merging all these non-uniform patterns into one class, the large amount of texture information that they represent may be discarded. By just considering pixels on the circularly symmetric neighbor sets, some information that can be utilized for describing image texture is lost. For this reason, many improved versions of LBP try to extract as much information as possible from these non-uniform patterns. However, in LBP framework, this problem cannot be resolved easily.

**2.3 Local pixel patterns**

In this section, we introduce our new texture operator. Before doing so, consider the following example. In Fig. 3, the gray values for a  $3 \times 3$  sub-block are illustrated. When selecting  $P = 8$  and  $R = 1$ , since the eight pixels on the circle have greater values than the central pixel, the two sub-blocks have the same LBP code “11111111”. Thus, according to the LBP operator, these two sub-blocks have the same image texture. This is definitely not the expected result.

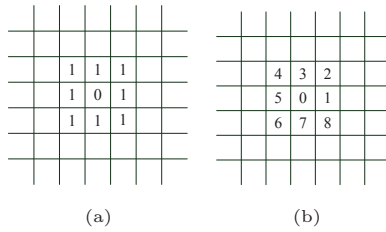


Fig. 3 Two textures with identical LBP.

The  $3 \times 3$  sub-block in Fig. 3(a) has only two different gray values, 0 and 1. However, in the  $3 \times 3$  sub-block in Fig. 3(b), the nine pixels have nine different gray values. In our method, we call the number of different gray values in each sub-block the local pixel pattern (LPP). Thus in Fig. 3(a), the LPP value is 2, whereas in Fig. 3(b) the LPP value is 9. While image texture cannot be precisely defined, there is no question that the texture in Fig. 3(b) is more complex than that in Fig. 3(a).

This phenomenon has inspired us to find a new way to describe image texture using LPPs. In our method, the input image with size  $M \times N$  is divided into overlapping sub-blocks with size  $B \times B$ , as shown by the example in Fig. 4. The input image has size  $5 \times 7$  and the first three overlapping  $3 \times 3$  sub-blocks are shown by the red, blue, and green boxes respectively. An image  $I$  with size  $M \times N$  has  $(M - B + 1) \times (N - B + 1)$  overlapping sub-blocks of size  $B \times B$ .

We assume that one sub-block  $SB_{i,j} (1 \leq i \leq (M - B + 1), 1 \leq j \leq (N - B + 1))$  with  $B \times B$  pixels is represented by

$$SB_{i,j} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,B} \\ \vdots & \ddots & \vdots \\ p_{B,1} & \cdots & p_{B,B} \end{bmatrix} \quad (5)$$

The LPP code for this  $B \times B$  sub-block is defined by

$$LPP_B(i, j) = Unique(SB_{i,j}) \quad (6)$$

where  $Unique(x)$  is a function that returns the number of unique elements in matrix  $x$ .

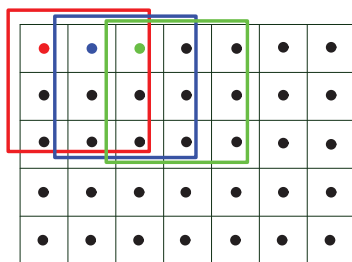


Fig. 4 Overlapping sub-blocks.

After the  $LPP_B$  code for each sub-block has been computed, a histogram is built to describe the texture:

$$H(k) = \sum_{i=1}^{M-B+1} \sum_{j=1}^{N-B+1} f(LPP_B(i, j), k), \quad k \in [1, K] \quad (7)$$

where  $K$  is the maximum value of  $LPP_B$  code and

$$f(x, y) = \begin{cases} 1, & x = y \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

Note that the maximum value of the  $LPP_B$  code is no bigger than  $B \times B$ .

It is clear that the LPP operator is a gray scale invariant, i.e., its output is not affected by any monotonic transformation of the gray scale.

### 2.4 Multi-bit plane LPP

Without loss of generality, we suppose that the input image  $I$  is an  $M \times N$  grayscale image (note that a color image can be approximately regarded as multiple grayscale images), which is composed of eight 1-bit planes:  $I_1$  to  $I_8$ , ranging from bit-plane 1 for the most significant bit to bit-plane 8 for the least significant bit. Generally, the first bit plane is the coarsest and the last bit plane gives critical detail in the image. The higher the bit plane number, the less is its contribution to the final stage. Since the least one or two bit planes of the input image are insignificant and may be affected by random noise, good results may be obtained if we neglect the least one or two bit planes when applying the LPP operator.

In our method, the first  $k (1 \leq k \leq 8)$  significant bit-planes are combined to give a new image  $P_k$ . Its pixel values  $P_k(m, n) (1 \leq m \leq M, 1 \leq n \leq N)$  may be computed as

$$P_k(m, n) = \sum_{i=1}^k I_i(m, n) 2^{k-i} \quad (9)$$

Thus, we can get eight new images  $P_1, \dots, P_8$ . Note that the new images  $P_1$  and  $P_8$  correspond to the most significant bit-plane  $I_1$  and the input image  $I$ , respectively. For ease of explanation, these newly obtained images  $P_1, \dots, P_8$  are called multi-bit plane images in the following sections. We can apply LPP to these multi-bit plane images with different sub-block sizes to get a series of image texture operators  $LPP_B^{P_k} (k = 1, \dots, 8; B = 3, 4, \dots)$ .

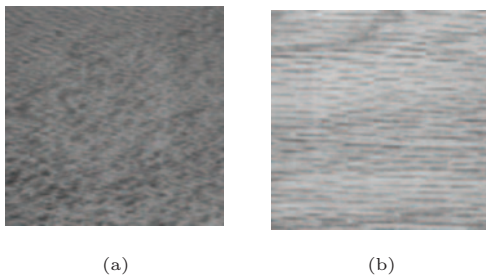
### 3 Experimental results

To evaluate the effectiveness of our proposed LPP operators, we compare them with LBP [5]. As noted, LPP describes the image texture from a different aspect from that of LBP. Firstly, experimental results are given to demonstrate the difference between LPP and LBP via using some specific texture images. Secondly, detailed experiments using four different image texture databases demonstrate the effectiveness of LPP. The experiments also demonstrate that LPP can be used to supplement LBP to improve its capability to describe image texture.

In the following, the LPP operators applied to different multi-bit plane images with different sub-block sizes are represented by  $LPP_B^{P_k}$  ( $B = 3, 4, 5; k = 1, \dots, 8$ ), and the  $LBP_{P,R}^{riu2}$  operators applied on different multi-bit plane images with different  $(P, R)$  values are represented by  $LBP_{P,R}^{P_k}$  ( $(P, R) = (8, 1), (16, 2), (24, 3); k = 1, \dots, 8$ ).

#### 3.1 Difference between LPP and LBP

In order to demonstrate the difference between LPP and LBP, two images (water.090 and wood.090) were selected from the Brodatz database [22]; these are shown in Fig. 5. The two images have the same orientation  $90^\circ$  but belong to different image texture classes. For these two images, the normalized histograms corresponding to  $LPP_3^{P_8}$  and  $LBP_{8,1}^{P_8}$  are shown in Fig. 6 and Fig. 7, respectively. Figure 6 shows that when the  $LPP_3^{P_8}$  operator is used, the difference between the normalized histograms of water.090 and wood.090 may easily be discriminated. However, when the  $LBP_{8,1}^{P_8}$  operator is selected, Fig. 7 shows that the normalized histograms of water.090 and wood.090 are almost the same. Thus these two images (belonging to different image



**Fig. 5** Two input test images: (a) water.090; (b) wood.090.

texture classes) may be classified as belonging to the same image texture class.

We also applied LPP and LBP operators on multi-bit plane images; experimental results are shown in Fig. 8 and Fig. 9, which show that LPP can differentiate these two different image textures more accurately than LBP.

#### 3.2 Comparative results on four image datasets

In this section, a dissimilarity metric is introduced first, and then detailed experiments conducted on four different image texture databases are described which demonstrate the effectiveness of our new LPP operators.

##### 3.2.1 Dissimilarity metric

As in Ref. [11], in the classification phase, we evaluate the dissimilarity of sample and model histograms via a test of goodness-of-fit, measured by a  $\chi$ -squared test. In our study, a test sample  $S$  is assigned to the model  $M$  as follows:

$$L(S, M) = \sum_{t=1}^T (S_t - M_t)^2 / (S_t + M_t) \quad (10)$$

where  $T$  is the number of bins, and  $S_t$  and  $M_t$  correspond to the sample and model probabilities for bin  $t$  respectively.

As in Ref. [5], multi-operator analysis is accomplished using the sum of individual  $\chi$ -squared distances computed from the responses of individual operators:

$$L_N = \sum_{n=1}^N L(S^n, M^n) \quad (11)$$

where  $N$  is the number of operators, and  $S^n$  and  $M^n$  are the sample and model histograms corresponding to operator  $n$  ( $n = 1, \dots, N$ ) respectively.

##### 3.2.2 Image datasets and experimental setup

Four different image datasets were used to evaluate the robustness to rotation variations, illuminance variations, and surface transformations (e.g., view-point changes and non-rigid deformations of the texture surface), etc. The corresponding four image datasets are as follows.

- 1) Brodatz database [22]. The Brodatz image dataset [23] consists of thirteen texture classes digitized at seven different rotation angles ( $0^\circ$ ,  $30^\circ$ ,  $60^\circ$ ,  $90^\circ$ ,  $120^\circ$ ,  $150^\circ$ , and  $200^\circ$ ). For each texture class there are seven  $512 \times 512$  grayscale images and altogether 91 images.

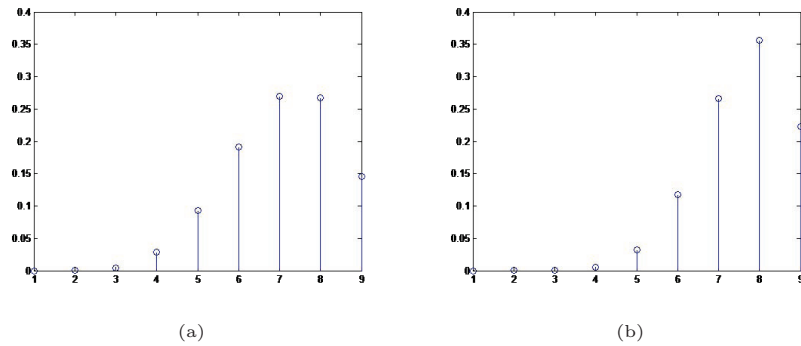


Fig. 6 Normalized histograms using the  $LPP_3^{P_s}$  operator: (a) water.090; (b) wood.090.

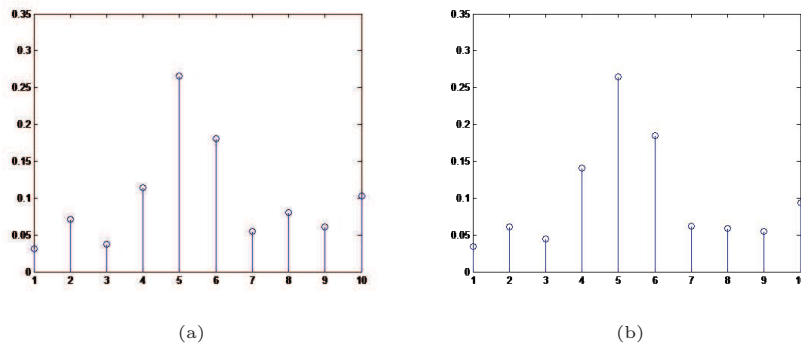


Fig. 7 Normalized histograms using the  $LBP_{8,1}^{P_s}$  operator: (a) water.090; (b) wood.090.

Those images with a rotation angle of  $0^\circ$  for each class are shown in Fig. 10, and as an example, for the first texture class, “bark”, the images with different rotation angles are shown in Fig. 11.

- 2) Outex\_TC\_00010 test suite [24]. See <http://www.outex.oulu.fi/>. This test suite contains 24 classes of texture images and altogether 4320 images of size  $128 \times 128$ . Each intensity image is individually normalized to an average intensity of 128 and a standard deviation of 20. Samples of illuminant “inca” and angle  $0^\circ$  in each class were used for classifier training, and the other eight rotation angles ( $5^\circ$ ,  $10^\circ$ ,  $15^\circ$ ,  $30^\circ$ ,  $45^\circ$ ,  $60^\circ$ ,  $75^\circ$ , and  $90^\circ$ ) with the same illuminant were used for testing. Hence, there are 480 ( $24 \times 20$ ) models and 3840 ( $24 \times 20 \times 8$ ) validation samples.
- 3) Outex\_TC\_00012 test suite [24]. Again see <http://www.outex.oulu.fi/>. There are 24 classes of texture images and altogether 9120 images with size  $128 \times 128$  in this test suite, again normalized as above. The classifier is trained with the same training samples as for the Outex\_TC\_00010 test suite. Testing used

all samples captured under illuminants “t184” and “horizon”. There are 480 ( $24 \times 20$ ) models and 4320 ( $24 \times 20 \times 9$ ) validation samples for each illuminant.

- 4) UIUC image dataset [25]. This texture database consists of 1000 uncalibrated, unregistered images of size  $640 \times 480$ ; it has 40 samples each of 25 different textures. The database includes surfaces whose texture is due mainly to albedo variations (e.g., wood and marble), 3D shape (e.g., gravel and fur), or a mixture of both (e.g., carpet and brick). Significant viewpoint changes and scale differences are present within each class, and illuminant conditions are uncontrolled. For each class, the original image without any variation is shown in Fig. 12, and as an example, for the first class “T01”, the images with different variations are shown in Fig. 13.

For the Outex\_TC\_00010 and Outex\_TC\_00012 test suites, the training and testing sets were selected according to their default settings. For the Brodatz and UIUC test datasets, we conducted all our experiments under a challenging condition. Any one sample could have been used as the model and the

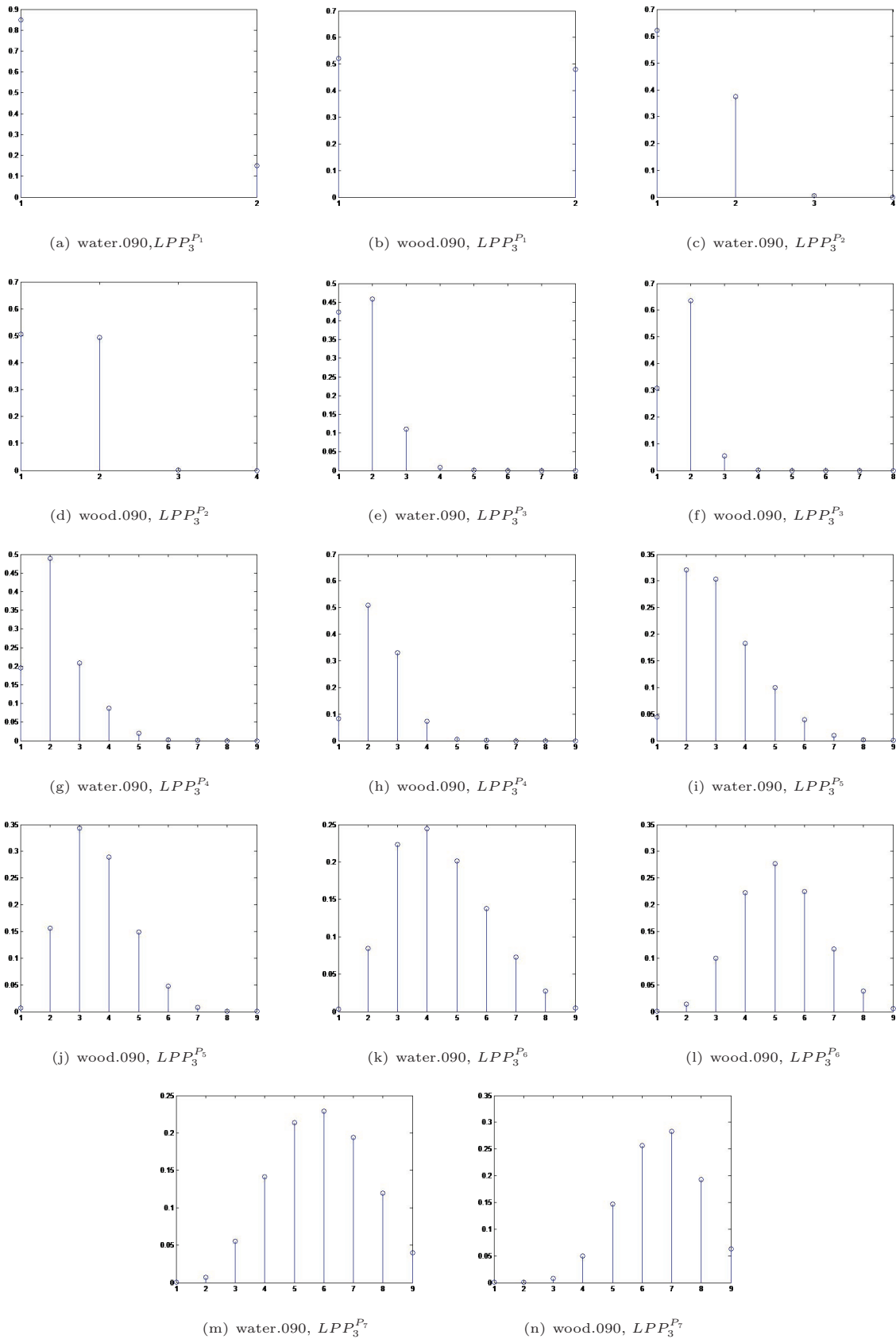


Fig. 8 Normalized histograms of  $LPP_3^{P_k}$  ( $k = 1, \dots, 7$ ).

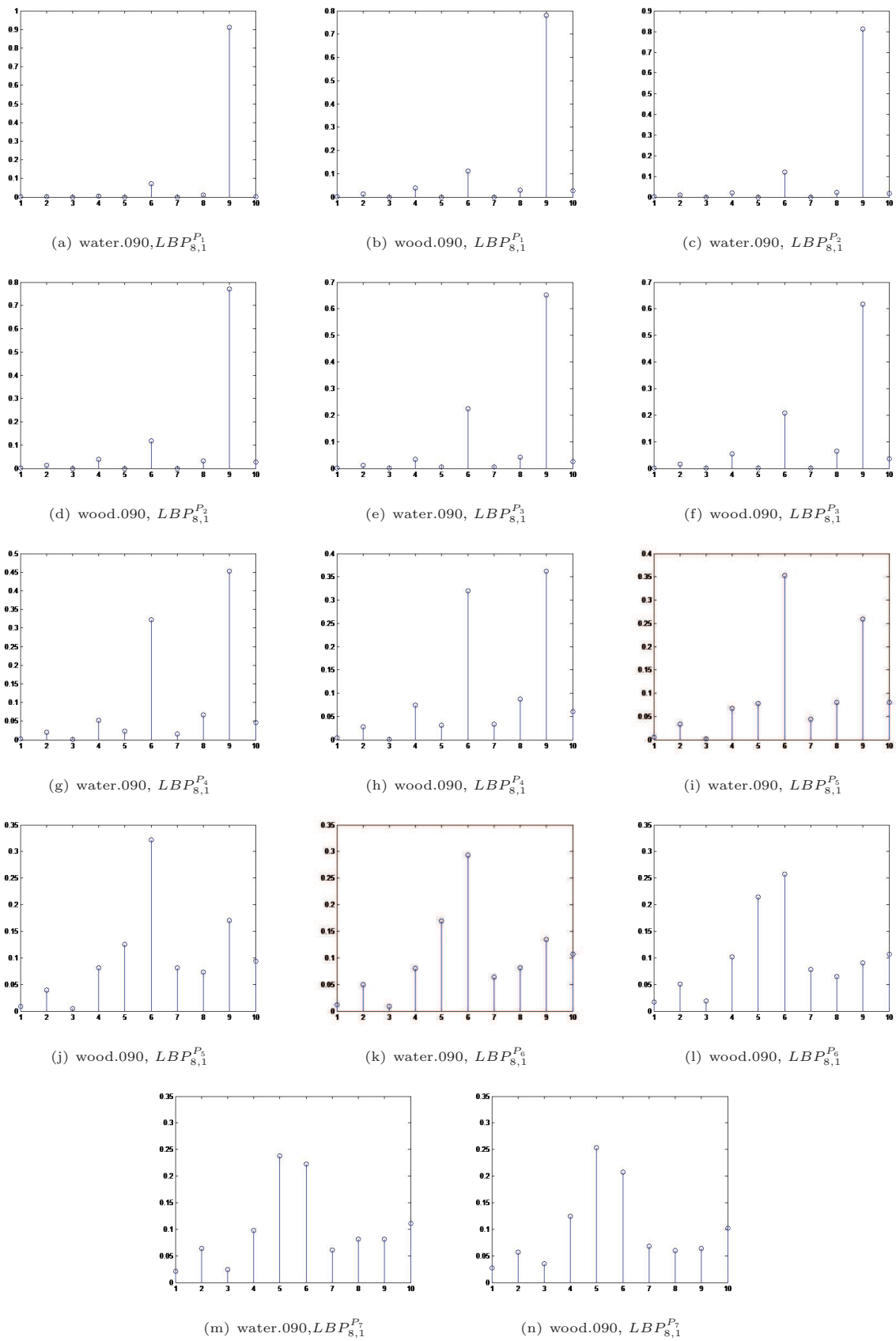


Fig. 9 Normalized histograms of  $LBP_{8,1}^{P_k}$  ( $k = 1, \dots, 7$ ).



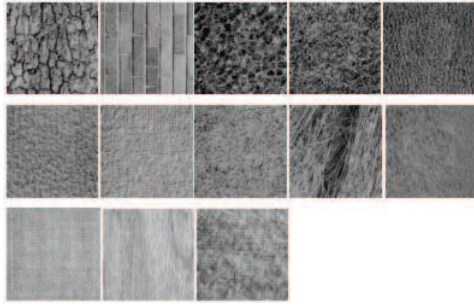


Fig. 10 Image classes in the Brodatz database.

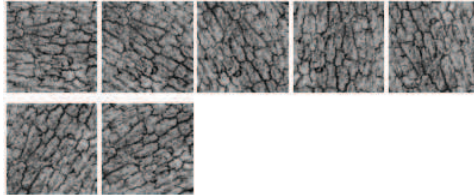


Fig. 11 Seven images with different rotation angles for texture class “bark” in the Brodatz database.

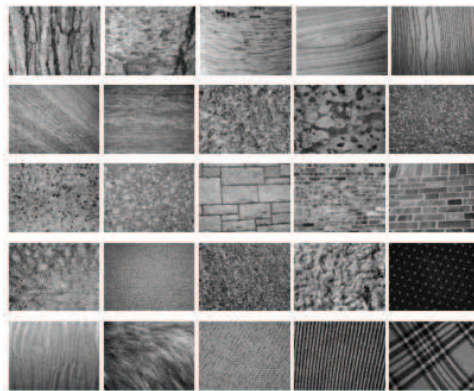


Fig. 12 Images classes in the UIUC database.

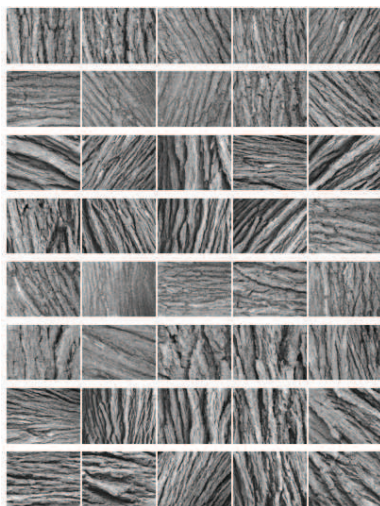


Fig. 13 Forty images with different variations for texture class “T01” in the UIUC database.

rest samples in the image dataset were utilized for testing. Suppose that the number of texture classes was  $U$ , and each texture class had  $V$  samples with different variations or surface transformations. The testing process was as follows. Firstly, one of texture image  $D_{i,j}$  ( $1 \leq i \leq U, 1 \leq j \leq V$ ) was utilized as the model. Then all the images in the image dataset except  $D_{i,j}$  were used for testing. If those images  $\{D_{i,1}, \dots, D_{i,j-1}, D_{i,j+1}, \dots, D_{i,V}\}$  in the same class of  $D_{i,j}$  were all among the nearest  $V - 1$  images to  $D_{i,j}$  according to Eq. (10), it means that all the samples were correctly classified. Otherwise, if some of the test images (there are  $v$  images) in  $\{D_{i,1}, \dots, D_{i,j-1}, D_{i,j+1}, \dots, D_{i,V}\}$  were not among the nearest  $V - 1$  images to  $D_{i,j}$ , the classification accuracy rate corresponding to the model image  $D_{i,j}$  would have been  $\frac{V - 1 - v}{V - 1}$ . After testing all images in the dataset, the average accuracy rate was reported as the testing result.

### 3.3 Experimental results on Brodatz

The experimental results on the Brodatz data are illustrated in Tables 1 and 2. Table 1 shows that our proposed LPP operators are more robust to rotational variations than LBP in general. For example, the best discrimination results obtained by LPP operators with sub-block sizes of 3, 4, and 5 are 98.4 ( $k = 3$ ), 99.1 ( $k = 4, 5$ ), and 99.6 ( $k = 6$ ), while the best discrimination results obtained by LBP operators with  $(P, R)$  values of  $(8, 1)$ ,  $(16, 2)$ , and  $(24, 3)$  are 89.0 ( $k = 6$ ), 94.9 ( $k = 8$ ), and 98.7 ( $k = 3$ ), respectively.

Moreover, by combining LPP with LBP, more powerful operators can easily be obtained. In our experiments, we combined  $LBP_{8,1}^{P_8}$  (i.e., the original LBP operator  $LBP_{8,1}^{riu2}$ ) with  $LPP_3^{P_k}$  ( $k = 1, \dots, 8$ ),  $LBP_{16,2}^{P_8}$  with  $LPP_4^{P_k}$  ( $k = 1, \dots, 8$ ), and  $LBP_{24,3}^{P_8}$  with  $LPP_5^{P_k}$  ( $k = 1, \dots, 8$ ) for a demonstration. The results are shown in Table 2. It is clear that these combined operators can achieve higher discrimination accuracy rates than the original LBP operators. Note that for our proposed operators  $LPP_B^{P_1}$ ,  $LPP_B^{P_2}$ , and  $LPP_B^{P_3}$  with any block size  $B$ , the dimensions of the feature vectors are 2, 4, and 8, respectively. The experimental results demonstrate that there is a high complementarity between LBP and our proposed LPP operators.

**Table 1** Comparison with LBP on the Brodatz test data

$LBP_{P,R}^{P_k}$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$
$(P, R) = (8, 1)$	78.4	82.4	81.3	81.3	87.8	<b>89.0</b>	87.0	86.3
$(P, R) = (16, 2)$	86.1	90.8	85.7	87.9	91.8	93.6	94.3	<b>94.9</b>
$(P, R) = (24, 3)$	96.2	94.7	<b>98.7</b>	92.3	96.2	96.7	97.1	98.0
$LPP_B^{P_k}$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$
$B = 3$	47.1	88.3	<b>98.4</b>	96.5	93.4	91.6	88.5	87.7
$B = 4$	51.1	94.3	98.7	<b>99.1</b>	<b>99.1</b>	97.3	94.0	92.3
$B = 5$	50.6	95.8	98.9	99.5	99.5	<b>99.6</b>	96.2	95.4

**Table 2** Combining LBP and LPP on the Brodatz test data

$LBP_{P,R}^{P_s} + LPP_B^{P_k}$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$
$(P, R)=(8, 1), B=3$	92.7	97.1	93.0	99.1	99.6	<b>99.8</b>	97.8	98.9
$(P, R)=(16, 2), B=4$	<b>100</b>	<b>100</b>	98.9	<b>100</b>	<b>100</b>	99.3	99.8	99.1
$(P, R)=(24, 3), B=5$	99.8	99.8	99.5	99.8	99.6	<b>100</b>	99.8	99.1

### 3.4 Experimental results on Outex\_TC\_00010

The experimental results on Outex\_TC\_00010 are illustrated in Tables 3 and 4. Table 3 shows that both LPP and LBP operators are robust to rotational variations as well as to some pre-processing operations such as average and deviation normalization of the pixel values. For example, the best discrimination results obtained by LPP operators with sub-block sizes of 3, 4, and 5 are 85.7 ( $k = 4$ ), 86.3 ( $k = 4$ ), and 86.2 ( $k = 5$ ), and the best discrimination results obtained by LBP operators with  $(P, R)$  values of (8, 1), (16, 2), and (24, 3) are 85.6 ( $k = 7$ ), 89.4 ( $k = 7, 8$ ), and 95.2 ( $k = 6, 8$ ), respectively.

Moreover, by combining LPP with LBP, more powerful operators can easily be obtained, as shown

in Table 4, with higher discrimination accuracy rates than the original LBP operators. The experimental results also demonstrate that there is a high complementarity between LBP and our proposed LPP operators.

### 3.5 Experimental results on Outex\_TC\_00012

Two problems were investigated using the Outex\_TC\_00012 test suite. In problem 000, samples of illuminant “inca” with angle  $0^\circ$  in each class were used for training and all samples captured under illuminant “t184” with nine rotation angles were used for testing. In problem 001, samples of illuminant “inca” with angle  $0^\circ$  in each class were used for training and all samples captured under illuminant “horizon” with nine rotation angles were used for testing. The experimental results are shown

**Table 3** Comparison with LBP on the Outex\_TC\_00010 test data

$LBP_{P,R}^{P_k}$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$
$(P, R) = (8, 1)$	72.2	74.9	83.4	83.9	84.7	84.0	<b>85.6</b>	84.9
$(P, R) = (16, 2)$	80.6	82.0	89.3	89.3	88.9	89.1	<b>89.4</b>	<b>89.4</b>
$(P, R) = (24, 3)$	87.9	88.3	92.3	92.0	92.5	<b>95.2</b>	94.8	<b>95.2</b>
$LPP_B^{P_k}$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$
$B = 3$	25.2	56.4	82.4	<b>85.7</b>	82.2	76.1	71.0	67.7
$B = 4$	22.4	58.3	82.5	<b>86.3</b>	85.2	82.4	77.7	79.1
$B = 5$	21.6	58.4	80.8	85.4	<b>86.2</b>	83.8	79.6	82.9

**Table 4** Combining LBP and LPP on the Outex\_TC\_00010 test data

$LBP_{P,R}^{P_s} + LPP_B^{P_k}$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$
$(P, R)=(8, 1), B=3$	91.3	94.3	<b>95.6</b>	95.1	95.1	94.1	92.9	91.0
$(P, R)=(16, 2), B=4$	93.8	95.4	96.5	96.2	<b>97.2</b>	95.9	95.9	94.8
$(P, R)=(24, 3), B=5$	96.3	96.9	96.8	96.4	<b>97.2</b>	97.0	96.1	95.8

in Tables 5 and 6.

Table 5 shows that with similar dimensions, LBP operators have generally better discrimination accuracy rates than our proposed LPP operators: the robustness to illuminance variations of our LPP operators is lower than that of LBP operators.

However, our LPP can be combined with LBP operators to achieve more powerful operators as before. Experimental results are shown in Table 6: in combination with LPP operators, the discrimination capability of LBP operators can be greatly improved. For example, in problem 000, the discrimination accuracy rate of  $LBP_{8,1}^{P_8}$  is 65.2. By combining it with  $LPP_3^{P_2}$  (note that the dimension of the  $LPP_3^{P_2}$  feature vector is 4), the obtained accuracy rate can be improved to 85.2. In problem 001, the obtained accuracy rate of  $LBP_{8,1}^{P_8}$  is 63.6. By combining it with  $LPP_3^{P_2}$ , the discrimination accuracy rate can be improved to 82.2.

### 3.6 Experimental results on UIUC

Because of the wide range of surface transformations

such as viewpoint changes and nonrigid deformations, UIUC is a challenging dataset for almost all texture descriptors. Experimental results corresponding to LBP and LPP operators are shown in Tables 7 and 8.

Table 7 shows that both LBP and LPP operators poorly discriminate the texture images in this image dataset. Nevertheless, the conclusion can be made from Table 8 that by combining LBP and LPP, the discrimination accuracy rate can be greatly improved. For example, by combining  $LPP_3^{P_3}$  the discrimination accuracy rate of  $LBP_{8,1}^{P_8}$  can be improved from 24.0 to 30.5, by combining  $LPP_4^{P_3}$  the discrimination accuracy rate of  $LBP_{16,2}^{P_8}$  can be improved from 23.8 to 33.5, and by combining  $LPP_5^{P_3}$  the discrimination accuracy rate of  $LBP_{24,3}^{P_8}$  can be improved from 25.8 to 34.5.

Another observation that can be made from Table 7 is that our proposed multi-bit plane strategy can also greatly improve the discrimination capability of LBP. For example, by using our multi-

**Table 5** Comparison with LBP on the Outex\_TC\_00012 test data

Problem	$LBP_{P,R}^{P_k}$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$
000	$(P, R)=(8, 1)$	64.4	66.9	<b>73.1</b>	70.9	70.7	70.7	67.4	65.2
	$(P, R)=(16, 2)$	78.8	79.1	79.5	79.0	80.9	<b>83.1</b>	82.8	82.5
	$(P, R)=(24, 3)$	82.5	80.8	80.5	78.7	84.8	<b>86.2</b>	84.4	85.1
001	$(P, R)=(8, 1)$	65.3	<b>72.7</b>	69.2	65.9	64.2	65.8	64.0	63.6
	$(P, R)=(16, 2)$	78.4	<b>80.5</b>	78.1	77.9	77.4	76.3	75.3	77.9
	$(P, R)=(24, 3)$	80.7	79.7	76.7	<b>82.5</b>	82.1	80.6	80.9	81.8
Problem	$LPP_B^{P_k}$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$
000	$B = 3$	18.6	41.6	<b>62.5</b>	60.3	56.9	52.2	37.8	32.5
	$B = 4$	18.5	41.4	<b>63.6</b>	58.6	60.9	58.2	43.7	36.8
	$B = 5$	15.8	39.9	<b>62.1</b>	58.4	60.9	60.5	46.1	37.6
001	$B = 3$	18.3	40.0	60.9	<b>62.1</b>	60.1	56.1	42.8	38.9
	$B = 4$	16.6	41.0	<b>64.1</b>	62.6	62.0	61.8	49.2	43.5
	$B = 5$	15.1	40.5	63.2	63.7	63.8	<b>64.5</b>	52.4	44.3

**Table 6** Combining LBP and LPP on the Outex\_TC\_00012 test data

Problem 000									
$LBP_{P,R}^{P_8} + LPP_B^{P_k}$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$	
$(P, R)=(8, 1), B=3$	76.6	<b>85.2</b>	83.4	78.2	77.6	79.0	64.8	51.7	
$(P, R)=(16, 2), B=4$	87.1	<b>87.7</b>	<b>87.7</b>	82.1	82.5	85.2	73.1	55.3	
$(P, R)=(24, 3), B=5$	88.0	85.9	<b>88.7</b>	83.2	82.8	82.6	69.5	51.9	
Problem 001									
$LBP_{P,R}^{P_8} + LPP_B^{P_k}$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$	
$(P, R)=(8, 1), B=3$	72.1	<b>82.2</b>	79.4	76.4	75.8	75.6	67.1	61.0	
$(P, R)=(16, 2), B=4$	77.5	82.8	<b>84.2</b>	82.0	81.3	82.0	75.6	62.2	
$(P, R)=(24, 3), B=5$	83.0	82.7	<b>84.8</b>	81.5	81.8	82.4	75.6	59.4	

**Table 7** Comparison with LBP on the UIUC dataset

$LBP_{P,R}^{P_k}$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$
$(P, R)=(8, 1)$	23.1	23.9	25.5	<b>29.1</b>	28.8	27.4	25.0	24.0
$(P, R)=(16, 2)$	30.3	30.5	30.8	<b>33.3</b>	30.2	26.3	24.3	23.8
$(P, R)=(24, 3)$	32.4	34.7	<b>37.8</b>	34.4	30.5	26.8	25.9	25.8
$LPP_B^{P_k}$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$
$B = 3$	11.2	17.9	19.9	<b>20.3</b>	19.4	18.9	17.2	16.2
$B = 4$	11.4	19.7	22.7	<b>23.2</b>	22.4	22.0	20.8	19.9
$B = 5$	11.7	20.7	24.1	<b>25.0</b>	24.5	24.2	23.5	22.9

**Table 8** Combining LBP and LPP on the UIUC dataset

$LBP_{P,R}^{P_k} + LPP_B^{P_k}$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$
$(P, R)=(8, 1), B=3$	25.3	28.9	<b>30.5</b>	30.0	29.1	28.4	27.3	26.5
$(P, R)=(16, 2), B=4$	26.5	31.6	<b>33.5</b>	33.2	32.7	32.4	31.6	31.3
$(P, R)=(24, 3), B=5$	26.8	32.5	<b>34.5</b>	<b>34.5</b>	34.2	34.0	33.8	33.6

bit plane strategy, the discrimination accuracy rates of  $LBP_{P,R}^{P_k}$  with  $(P, R)$  values of  $(8, 1)$ ,  $(16, 2)$ , and  $(24, 3)$  can be improved from 24.0 to 29.1 ( $k = 4$ ), 23.8 to 33.3 ( $k = 4$ ), and 25.8 to 37.8 ( $k = 3$ ), respectively.

## 4 Conclusions

Image texture analysis has been the topic of extensive research in the past few decades. However, no texture analysis method is perfect and able to tackle all texture problems. As a classical and challenging topic, new ideas and new techniques are called for. In this paper, we have proposed a novel class of texture operators. The main contributions of this paper are as follows.

- 1) We firstly show that the number of gray scales in a sub-block is a fundamental property that can be utilized to describe local image texture.
- 2) A new multi-bit plane strategy has been proposed which may also have other applications to texture analysis.
- 3) The LPP operators have low feature dimension and are complementary to LBP operators; they can easily be combined with LBP operators to form more powerful image texture operators.

## Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (61173147, 61332012, and U1135001), the National Basic Research Program of China (2011CB302204),

the Fundamental Research Funds for Central Universities (12lgpy31), the Korea Foundation for Advanced Studies' International Scholar Exchange Fellowship for the academic year of 2013–2014, and the MKE (The Ministry of Knowledge Economy), R. O. Korea, under the ITRC support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2010-C1090-1001-0004).

**Open Access** This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

## References

- [1] Haralick, R. M.; Shanmugan, K.; Dinstein, I. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics* Vol. SMC-3, No. 6, 610–621, 1973.
- [2] Cross, G. R.; Jain, A. K. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. PAMI-5, No. 1, 25–39, 1983.
- [3] Bovik, A. C.; Clark, M.; Geisler, W. S. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 12, No. 1, 55–73, 1990.
- [4] Chang, T.; Kuo, C.-C. J. Texture analysis and classification with tree-structured wavelet transform. *IEEE Transactions on Image Processing* Vol. 2, No. 4, 429–441, 1993.
- [5] Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 24, No. 7, 971–987, 2002.
- [6] Tan, X.; Triggs, B. Enhanced local texture feature sets

- for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing* Vol. 19, No. 6, 1635–1650, 2010.
- [7] Zhou, H.; Wang, R.; Wang, C. A novel extended local-binary-pattern operator for texture analysis. *Information Sciences* Vol. 178, No. 22, 4314–4325, 2008.
- [8] Nanni, L.; Brahnam, S.; Lumini, A. A simple method for improving local binary patterns by considering non-uniform patterns. *Pattern Recognition* Vol. 45, No. 10, 3844–3852, 2012.
- [9] Liao, S.; Law, M. W. K.; Chung, A. C. S. Dominant local binary patterns for texture classification *IEEE Transactions on Image Processing* Vol. 18, No. 5, 1107–1118, 2009.
- [10] Guo, Z.; Zhang, L.; Zhang, D. Rotation invariant texture classification using LBP variance (LBPV) with global matching. *Pattern Recognition* Vol. 43, No. 3, 706–719, 2010.
- [11] Guo, Z.; Zhang, D.; Zhang, D. A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing* Vol. 19, No. 6, 1657–1663, 2010.
- [12] Zhao, Y.; Huang, D.-S.; Jia, W. Completed local binary count for rotation invariant texture classification. *IEEE Transactions on Image Processing* Vol. 21, No. 10, 4492–4497, 2012.
- [13] Khellah, F. M. Texture classification using dominant neighborhood structure. *IEEE Transactions on Image Processing* Vol. 20, No. 11, 3270–3279, 2011.
- [14] Maani, R.; Kalra, S.; Yang, Y.-H. Rotation invariant local frequency descriptors for texture classification. *IEEE Transactions on Image Processing* Vol. 22, No. 6, 2409–2419, 2013.
- [15] Ahonen, T.; Hadid, A.; Pietikainen M. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 28, No. 12, 2037–2041, 2006.
- [16] Zhang, B.; Gao, Y.; Zhao, S.; Liu, J. Local derivative pattern versus local binary pattern: Face recognition with high-order local pattern descriptor. *IEEE Transactions on Image Processing* Vol. 19, No. 2, 533–544, 2010.
- [17] Huang, D.; Shan, C.; Ardabilian, M.; Wang, Y. Local binary patterns and its application to facial image analysis: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* Vol. 41, No. 6, 765–781, 2011.
- [18] Zhao, G.; Ahonen, T.; Matas, J.; Pietikainen, M. Rotation-invariant image and video description with local binary pattern features. *IEEE Transactions on Image Processing* Vol. 21, No. 4, 1456–1477, 2012.
- [19] Choi, I.; Kim, D. Generalized binary pattern for eye detection. *IEEE Signal Processing Letters* Vol. 20, No. 4, 343–346, 2013.
- [20] Murala, S.; Maheshwari, R. P.; Balasubramanian, R. Local tetra patterns: A new feature descriptor for content-based image retrieval. *IEEE Transactions on Image Processing* Vol. 21, No. 5, 2874–2886, 2012.
- [21] Lin, C.-H.; Liu, C.-W.; Chen, H.-Y. Image retrieval and classification using adaptive local binary patterns based on texture features. *IET Image Processing* Vol. 6, No. 7, 822–830, 2012.
- [22] Brodatz, P. *Textures: A Photographic Album for Artists and Designers*. New York, NY, USA: Dover Publications, 1966.
- [23] Information on <http://sipi.usc.edu/database/database.php?volume=textures>.
- [24] Information on <http://www.outex.oulu.fi/>.
- [25] Lazebnik, S.; Schmid, C.; Ponce, J. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 27, No. 8, 1265–1278, 2005.
- [26] Julesz, B. Visual pattern discrimination. *IRE Transactions on Information Theory* Vol. 8, No. 2, 84–92, 1962.



**Fangjun Huang** received the B.S. degree from Nanjing University of Science and Technology, China, in 1995, and the M.S. and Ph.D. degrees from Huazhong University of Science and Technology, China, in 2002 and 2005, respectively. From June 2009 to June 2010, he was a postdoctoral researcher in the Department of Electrical and Computer Engineering, New Jersey Institute of Technology. From September 2013 to August 2014, He was a Korea Foundation for Advanced Studies (KFAS) scholar in Korea University. He is currently an associate professor with the School of Information Science and Technology, Sun Yat-sen University, China. His research interests include multimedia security and image processing.



**Xiaochao Qu** received his B.S. degree in computer science and technology from Harbin Institute of Technology, China, in 2009. He is currently pursuing a Ph.D. degree in Korea University. His research interests are reversible watermarking, image processing, and pattern recognition.



**Hyoung Joong Kim** got the B.S. in electrical engineering in 1978, and M.S. and Ph.D. degrees in control and instrumentation engineering in 1986 and 1989, respectively, from Seoul National University. He was a visiting scholar from 1992 to 1993 at the University of Southern California, USA. He was a professor of Control and Instrumentation Engineering Department of Kangwon National University, Chuncheon, R. O. Korea, from 1989 to 2006. He is currently a professor of Korea University, R. O. Korea. His research interests include distributed computing, machine learning, and multimedia security.



**Jiwu Huang** received his B.S. degree from Xidian University, China, in 1982, M.S. degree from Tsinghua University, China, in 1987, and Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, China, in 1998. He is currently a professor with the College of Information Engineering (also

Shenzhen Key Laboratory of Media Security), Shenzhen University, China. His current research interests include multimedia forensics and security. He serves as an associate editor for *IEEE Transactions on Information Forensics and Security*, and he also serves as a member of the IEEE SPS Information Forensics and Security Technical Committee.

Other papers from this open access journal are available at no cost from <http://www.springer.com/journal/41095>. To submit a manuscript, please go to <https://www.editorialmanager.com/cvmj>.