A Completed Modeling of Local Binary Pattern Operator for Texture Classification

Zhenhua Guo, Lei Zhang, *Member*, *IEEE*, and David Zhang*, *Fellow*, *IEEE*

***Abstract*—In this paper, a completed modeling of the LBP operator is proposed and an associated completed LBP (CLBP) scheme is developed for texture classification. A local region is represented by its center pixel and a local difference sign-magnitude transform (LDSMT). The center pixels represent the image gray level and they are converted into a binary code, namely CLBP-Center (CLBP_C), by global thresholding. LDSMT decomposes the image local differences into two complementary components: the signs and the magnitudes, and two operators, namely CLBP-Sign (CLBP_S) and CLBP-Magnitude (CLBP_M), are proposed to code them. The traditional LBP is equivalent to the CLBP_S part of CLBP, and we show that CLBP_S preserves more information of the local structure than CLBP_M, which explains why the simple LBP operator can extract the texture features reasonably well. By combining CLBP_S, CLBP_M, and CLBP_C features into joint or hybrid distributions, significant improvement can be made for rotation invariant texture classification.**

***Index Terms*—Local Binary Pattern, Rotation Invariance, Texture Classification**

I. Introduction

Texture classification is an active research topic in computer vision and pattern recognition. Early texture classification methods focus on the statistical analysis of texture images. The representative ones include the co-occurrence matrix method [1] and the filtering based methods [2]. Kashyap and Khotanzad [3] were among the first researchers to study rotation-invariant texture classification by using a circular autoregressive model. In the early stage, many models were explored to study rotation invariance for texture classification, including hidden Markov model [4] and Gaussian Markov random filed [5]. Recently, Varma and Zisserman [6] proposed to learn a rotation invariant texton dictionary from a training set, and then classify the texture image based on its texton distribution. Later, Varma and Zisserman [7-8] proposed another texton based algorithm by using the image local patch to represent features directly. Some works have been recently proposed for scale and affine invariant texture classification by using fractal analysis [9-10] and affine adaption [11-12].

In [13], Ojala *et al* proposed to use the Local Binary Pattern (LBP) histogram for rotation invariant texture classification. LBP is a simple yet efficient operator to describe local image pattern, and it has achieved impressive classification results on representative texture databases [14]. LBP has also been adapted to many other applications, such as face recognition [15], dynamic texture recognition [16] and shape localization [17].

Despite the great success of LBP in computer vision and pattern recognition, its underlying working mechanism still needs more investigation. Before proposing LBP, Ojala *et al* [18] used the Absolute Gray Level Difference (AGLD) between a pixel and its neighbours to generate textons, and used the histogram of them to represent the image. Later, they proposed LBP [13] to use the sign, instead of magnitude, of the difference to represent the local pattern. Ojala *et al* [19] also proposed a multidimensional distribution of Signed Gray Level Difference (SGLD) and regarded LBP as a simplified operator of SGLD by keeping sign patterns only. Ahonen and Pietikäinen [20] analyzed LBP from a viewpoint of operator implementation. Tan and Triggs [21] proposed Local Ternary Pattern (LTP) to quantize the difference between a pixel and its neighbours into three levels. Although some variants of LBP, such as derivative-based LBP [17], dominant LBP [22] and center-symmetric LBP [23], have been proposed recently, there still remain some questions to be better answered for LBP. For example, why the simple LBP code could convey so much discriminant information of the local structure? What kind of information is missed in LBP code, and how to effectively represent the missing information in the LBP style so that better texture classification can be achieved?

This paper attempts to address these questions by proposing a new local feature extractor to generalize and complete LBP, and we name the proposed method completed LBP (CLBP). In CLBP, a local region is represented by its center pixel and a local difference sign-magnitude transform (LDSMT). The center pixel is simply coded by a binary code after global thresholding, and the binary map is named as CLBP_Center (CLBP_C). The LDSMT decomposes the image local structure into two complementary components: the difference signs and the difference magnitudes. Then two operators, CLBP-Sign (CLBP_S) and CLBP-Magnitude (CLBP_M), are proposed to code them. All the three code maps, CLBP_C, CLBP_S and CLBP_M, are in binary format so that they can be readily combined to form the final CLBP histogram. The CLBP could achieve much better rotation invariant texture classification results than conventional LBP based schemes.

Several observations can be made for CLBP. First, LBP is a special case of CLBP by using only CLBP_S. Second, we will show that the sign component preserves more image local structural information than the magnitude component. This explains why the simple LBP (i.e. CLBP_S) operator works much better than CLBP_M for texture classification. Third, the proposed CLBP_S, CLBP_M and CLBP_C code maps have

The work is partially supported by the GRF fund from the HKSAR Government, the central fund from Hong Kong Polytechnic University, and the National Science Foundation of China.

Z. Guo, L. Zhang and D. Zhang are with Department of Computing, the Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China. ({cszguo, cslzhang, csdzhang}@comp.polyu.edu.hk).

* Corresponding author, phone: 852-27667271, fax: 852-27740842

the same format so that they can be readily fused, and the texture classification accuracy can be significantly improved after fusion.

The rest of the paper is organized as follows. Section II briefly reviews LBP. Section III presents the CLBP scheme. Section IV reports extensive experimental results and Section V concludes the paper.

II. Brief Review of LBP

Given a pixel in the image, an LBP [13] code is computed by comparing it with its neighbours:

$$LBP_{P,R}=\sum_{p=0}^{P-1} s(g_p-g_c)2^p,\ s(x)=\begin{cases}1, x\ge 0\\ 0, x<0\end{cases} \quad (1)$$

where g_c is the gray value of the central pixel, g_p is the value of its neighbors, P is the total number of involved neighbors and R is the radius of the neighborhood. Suppose the coordinate of g_c is (0, 0), then the coordinates of g_p are $(R\cos(2\pi p/P), R\sin(2\pi p/P))$. The gray values of neighbours that are not in the image grids can be estimated by interpolation. Suppose the image is of size I*J. After the LBP pattern of each pixel is identified, a histogram is built to represent the texture image:

$$H(k)=\sum_{i=1}^{I}\sum_{j=1}^{J} f(LBP_{P,R}(i,j),k), k\in[0,K],$$
$$f(x,y)=\begin{cases}1, x=y\\ 0, otherwise\end{cases} \quad (2)$$

where K is the maximal LBP pattern value. The U value of an LBP pattern is defined as the number of spatial transitions (bitwise 0/1 changes) in that pattern

$$U(LBP_{P,R})=\left|s(g_{P-1}-g_c)-s(g_0-g_c)\right|+\sum_{p=1}^{P-1}\left|s(g_p-g_c)-s(g_{p-1}-g_c)\right| \quad (3)$$

The uniform LBP patterns refer to the patterns which have limited transition or discontinuities (U≤2) in the circular binary presentation [13]. In practice, the mapping from $LBP_{P,R}$ to $LBP_{P,R}^{u2}$ (superscript "*u2*" means uniform patterns with U≤2), which has P*(P-1)+3 distinct output values, is implemented with a lookup table of 2^P elements.

To achieve rotation invariance, a locally rotation invariant pattern could be defined as:

$$LBP_{P,R}^{riu2}=\begin{cases}\sum_{p=0}^{P-1} s(g_p-g_c) & if\ \ U(LBP_{P,R})\le 2\\ P+1 & \text{otherwise}\end{cases} \quad (4)$$

The mapping from $LBP_{P,R}$ to $LBP_{P,R}^{riu2}$ (superscript "*riu2*" means rotation invariant "uniform" patterns with U≤2), which has P+2 distinct output values, can be implemented with a lookup table.

III. Completed LBP (CLBP)

A. Local Difference Sign-Magnitude Transform

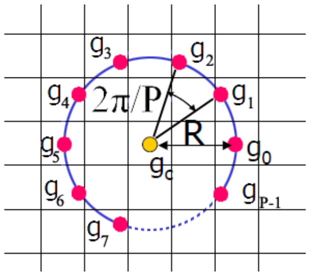

Figure 1: Central pixel and its P circularly and evenly spaced neighbours with radius R.

(a)

9	12	34
10	25	28
99	64	56

(b)

-16	-13	9
-15		3
74	39	31

(c)

-1	-1	1
-1		1
1	1	1

(d)

16	13	9
15		3
74	39	31

Figure 2: (a) A 3×3 sample block; (b) the local differences; (c) the sign and (d) magnitude components.

Referring to Fig. 1, given a central pixel g_c and its P circularly and evenly spaced neighbours g_p, p=0,1,…,P-1, we can simply calculate the difference between g_c and g_p as $d_p=g_p-g_c$. The local difference vector $[d_0,\ldots,d_{P-1}]$ characterizes the image local structure at g_c. Because the central gray level g_c is removed, $[d_0,\ldots,d_{P-1}]$ is robust to illumination changes and they are more efficient than the original image in pattern matching. d_p can be further decomposed into two components:

$$d_p=s_p*m_p \text{ and } \begin{cases}s_p=\text{sign}(d_p)\\ m_p=|d_p|\end{cases} \quad (5)$$

where $s_p=\begin{cases}1, d_p\ge 0\\ -1, d_p<0\end{cases}$ is the sign of d_p and m_p is the magnitude of d_p. With Eq. (5), $[d_0,\ldots,d_{P-1}]$ is transformed into a sign vector $[s_0,\ldots,s_{P-1}]$ and a magnitude vector $[m_0,\ldots,m_{P-1}]$.

We call Eq. (5) the local difference sign-magnitude transform (LDSMT). Obviously, $[s_0,\ldots,s_{P-1}]$ and $[m_0,\ldots,m_{P-1}]$ are complementary and the original difference vector $[d_0,\ldots,d_{P-1}]$ can be perfectly reconstructed from them. Fig. 2 shows an example. Fig. 2a is the original 3×3 local structure with central pixel being 25. The difference vector (Fig. 2b) is [3, 9,-13,-16,-15, 74, 39, 31]. After LDSMT, the sign vector (Fig. 2c) is [1,1,-1,-1,-1,1,1,1] and the magnitude vector (Fig. 2d) is[3, 9, 13, 16, 15, 74, 39, 31]. It is clearly seen that the original LBP uses only the sign vector to code the local pattern as an 8-bit string "11000111" ("-1" is coded as "0").

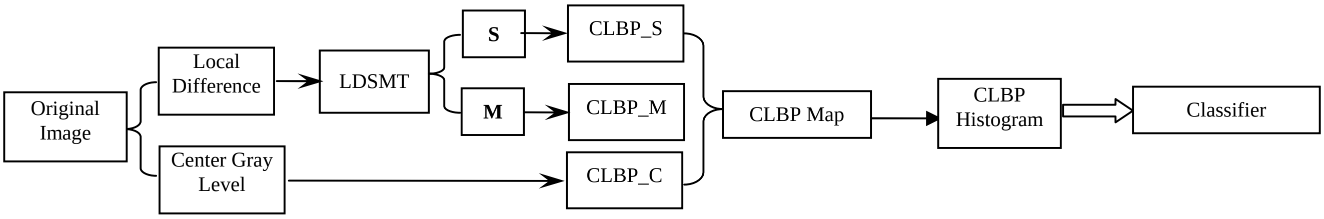

Figure 3: Framework of CLBP.

B. Analysis on Sign & Magnitude Components

Vector $[d_0,...,d_{P-1}]$ characterizes the image local structure. However, texture recognition by direct matching $[d_0,...,d_{P-1}]$ is infeasible because it is sensitive to noise, translation and rotation, etc. Thus we need to extract the distinct and stable features from $[d_0,...,d_{P-1}]$ to robustly recognize texture patterns. In Section III.A, we have seen that LBP actually uses only the sign component of $[d_0,...,d_{P-1}]$ for pattern recognition. Apparently, this may lead to some incorrect matches. For example, the difference vectors [3, 9, -13, -16, -15, 74, 39, 31] and [150, 1, -150, -1, -100, 150, 1, 150] have the same sign vector [-1,-1,-1,1,1,1,1,1]. However, it is hard to say they have similar local structures.

Therefore, several issues need to be addressed for LBP based feature representation. First, why LBP works reasonably well by using only the sign components of the local difference vector? Second, how to exploit the remaining information existed in the magnitude component? Third, can we design a scheme to efficiently and conveniently fuse the sign-magnitude features? In this sub-section we focus on the first issue and the other two issues will be discussed in the next sub-section.

The local difference can be perfectly reconstructed from its sign-magnitude components by $d_p=s_p*m_p$. One intuitive question is that which component, s_p or m_p, is more informative to represent the original local difference d_p? From a viewpoint of signal reconstruction, we can answer this question by reconstructing d_p using only s_p or m_p, and then checking which component can yield a smaller reconstruction error. Since d_p is the multiplication of s_p and m_p, we cannot directly reconstruct d_p by leaving one of s_p and m_p out. It is well accepted that the difference signal d_p can be well modeled by Laplace distribution $Q(x)=\exp\left(-|x|/\lambda\right)/\lambda$ [24], where parameter λ depends on the image content. Here we apply some prior knowledge to the probability distribution of s_p and m_p. It can be observed that the sign component s_p follows a Bernoulli distribution $Q(n)=b^{\frac{n+1}{2}}(1-b)^{\frac{1-n}{2}}$ with b=0.5 and $n\in\{-1,1\}$. Thus the local difference can be reconstructed by using only the magnitude component as

$$\hat{d}_p^m = m_p * s_p^{'} \tag{6}$$

where variable $s_p^{'}$ follows Bernoulli distribution $Q(n)$. Since d_p follows Laplace distribution, its magnitude m_p will follow a single-side Laplace distribution $Q(x)=\exp\left(-x/\lambda\right)/\lambda,\quad x\geq 0$. Thus the local difference can be reconstructed by using only the sign component as

$$\hat{d}_p^s = m_p^{'} * s_p \tag{7}$$

where $m_p^{'}$ can be set as the mean value of the magnitude component m_p.

The local difference reconstruction errors made by using s_p and m_p can be defined as

$$E_s = E\left[\left(d_p-\hat{d}_p^s\right)^2\right],\ E_m = E\left[\left(d_p-\hat{d}_p^m\right)^2\right] \tag{8}$$

After some mathematical calculation, we can derive $E_s=\lambda^2$ and $E_m=4\lambda^2$. Obviously, E_s is only ¼ of E_m. To further validate this conclusion, we calculated E_s and E_m for 864 texture images selected from the Outex database [14]. The average values of E_s and E_m are 98 and 403, respectively. This is exactly identical to the above mathematical derivation.

From the above analysis, we see that d_p can be more accurately approximated by using the sign component s_p than the magnitude component m_p. This implies that s_p will preserve more information of d_p than m_p, and hence it is more likely to result in better pattern recognition performance. This is identical to our experimental result in Section IV, where the texture classification using the sign features achieves much higher accuracy than using the magnitude features. It will also be seen that by coding both the sign features and magnitude features into rotation invariant binary codes and fusing them, much better texture classification results can be obtained than using only one of them.

C. CLBP_S, CLBP_M, and CLBP_C Operators

In Sub-section III.B, we illustrated that the sign component preserves much the information of local difference. This explains why the simple LBP technique can reasonably represent the image local features. Meanwhile, we see that the magnitude component may contribute additional discriminant information if it is properly used. In addition, the intensity value of the center pixel itself can also contribute useful information [7-8]. In this sub-section, we present a completed LBP (CLBP) framework to explore all the three types of features.

The CLBP framework is illustrated in Fig. 3. We first represent the original image as its center gray level (C) and the local difference. The local difference is then decomposed into the sign (S) and magnitude (M) components by the LDSMT defined in Eq. (5). Consequently, three operators, namely CLBP_C, CLBP_S and CLBP_M, are proposed to code the C, S and M features, respectively. Then, the CLBP_C, CLBP_S and CLBP_M codes are combined to form the CLBP feature map of the original image. Finally, a CLBP histogram can be built, and some classifier, such as the nearest neighbourhood classifier, can be used for texture classification.

The CLBP_S operator is the same as the original LBP operator defined in Eq. (1). Since the M components are of continuous values instead of the binary “1” and “-1” values, they cannot be directly coded as that of S. Inspired by the coding strategy of CLBP_S (i.e. LBP) and in order to code M in a consistent format with that of S, we define the following CLBP_M operator:

$$CLBP\_M_{P,R} = \sum_{p=0}^{P-1} t(m_p, c) 2^p, \quad t(x,c) = \begin{cases} 1, x \ge c \\ 0, x < c \end{cases} \tag{9}$$

where c is a threshold to be determined adaptively. Here we set it as the mean value of m_p from the whole image. Similar to $LBP_{P,R}^{riu2}$, the rotation invariant version of $CLBP\_M_{P,R}$, denoted by $CLBP\_M_{P,R}^{riu2}$, can also be defined to achieve rotation invariant classification.

Both CLBP_S and CLBP_M produce binary strings so that they can be conveniently used together for pattern classification. There are two ways to combine the CLBP_S and CLBP_M codes: in concatenation or jointly. In the first way, we calculate the histograms of the CLBP_S and CLBP_M codes separately, and then concatenate the two histograms together. This CLBP scheme can be represented as "CLBP_S_M". In the second way, we calculate a joint 2D histogram of the CLBP_S and CLBP_M codes. This CLBP scheme is represented as "CLBP_S/M".

The center pixel, which expresses the image local gray level, also has discriminant information. To make it consistent with CLBP_S and CLBP_M, we code it as:

$$CLBP\_C_{P,R} = t(g_c, c_I) \tag{10}$$

where t is defined in Eq. (9) and the threshold c_I is set as the average gray level of the whole image.

The three operators, CLBP_S, CLBP_M and CLBP_C, could be combined in two ways, jointly or hybridly. In the first way, similar to the 2D joint histogram, we can build a 3D joint histogram of them, demoted by "CLBP_S/M/C". In the second way, a 2D joint histogram, "CLBP_S/C" or "CLBP_M/C" is built first, and then the histogram is converted to a 1D histogram, which is then concatenated with CLBP_M or CLBP_S to generate a joint histogram, denoted by "CLBP_M_S/C" or "CLBP_S_M/C".

D.Dissimilarity Metric and Multi-scale CLBP

There are various metrics to evaluate the goodness between two histograms, such as histogram intersection, log-likelihood ratio, and chi-square statistic [13]. In this study, a test sample T is assigned to the class of model L that minimizes the chi-square distance:

$$D(T,L) = \sum_{x=1}^{X} (T_x - L_x)^2 \big/ (T_x + L_x) \tag{11}$$

where X is the number of bins, and T_x and L_x are respectively the values of the sample and the model image at the x^{th} bin. In this paper, the nearest neighborhood classifier with the chi-square distance is used to measure the dissimilarity between two histograms.

Multiresolution analysis could be used to improve the classification accuracy, that is, by employing multiple operators of various (P, R). In this study, we use a straightforward multiresolution analysis that measures the dissimilarity as the sum of chi-square distances from all operators [13]:

$$D_Y = \sum_{y=1}^{Y} D(S^y, Z^y) \tag{12}$$

where Y is the number of operators, and S^y and Z^y are respectively the sample and model histograms extracted from the y^{th} (y=1,2,…,Y) operator.

IV. EXPERIMENTAL RESULTS

To evaluate the effectiveness of the proposed method, we carried out a series of experiments on two large and comprehensive texture databases: the Outex database [14], which includes 24 classes of textures collected under three illuminations and at nine angles, and the Columbia-Utrecht Reflection and Texture (CUReT) database, which contains 61 classes of real-world textures, each imaged under different combinations of illumination and viewing angle [25]. As in [6-8], we chose 92 sufficiently large images for each class with a viewing angle less than 60^0 in the experiments.

A. The methods in comparison

As an LBP based scheme, the proposed CLBP is compared with the representative LBP schemes in [13], and the LTP algorithm in [21]. We also compare CLBP with two state-of-the-art rotation invariant texture classification algorithms, the VZ_MR8 in [6] and the VZ_Joint in [7-8].

There are three LBP operators in [13]: $LBP_{P,R}^{riu2}$, $VAR_{P,R}$ and the joint $LBP_{P,R}^{riu2}/VAR_{P,R}$. $VAR_{P,R}$ is the local intensity invariant: $VAR_{P,R} = \frac{1}{P}\sum_{p=0}^{P-1}(g_p - u)^2$, where $u = \frac{1}{P}\sum_{p=0}^{P-1} g_p$. However, $VAR_{P,R}$ has continuous values and it needs to be quantized. In our experiments, the quantization levels for $VAR_{P,R}$ and $LBP_{P,R}^{riu2}/VAR_{P,R}$ are respectively set as 128 and 16 bins according to [13].

Different from LBP, LTP quantizes local difference into three levels by a threshold. For computation simplicity, the ternary pattern is spitted into two LBPs, positive LBP and negative LBP. Then two histograms are built and concatenated into one histogram [21].

In VZ_MR8 [6], 40 textons are clustered from each of the n classes using the training samples, and then a histogram based on the n*40 textons is computed for each model and sample image. Similarly, the VZ_Joint algorithm [7-8] learns 40 textons from each texture class and builds a histogram for each image. Here, the 7*7 local patch is used as that in [7-8].

In the experiments we evaluate different combinations of the three operators proposed, including $CLBP\_S_{P,R}^{riu2}$ (it is the same as $LBP_{P,R}^{riu2}$), $CLBP\_M_{P,R}^{riu2}$, $CLBP\_M_{P,R}^{riu2}/C$, $CLBP\_S_{P,R}^{riu2}\_M_{P,R}^{riu2}/C$, $CLBP\_S_{P,R}^{riu2}/M_{P,R}^{riu2}$, and $CLBP\_S_{P,R}^{riu2}/M_{P,R}^{riu2}/C$ (refer to Section III.B for the notation of CLBP).

Please note that VZ_MR8, VZ_Joint, $VAR_{P,R}$ and $LBP_{P,R}^{riu2}/VAR_{P,R}$ are training based methods (in feature extraction), while LTP and the proposed CLBP are training free. In the following experiments, except for VZ_MR8 and VZ_Joint, each texture sample was normalized to have an average intensity 128 and a standard deviation 20 [13]. For the VZ_MR8 and VZ_Joint methods, the image sample was normalized to have an average intensity of 0 and a standard deviation of 1 [6-8]. This is to remove global intensity and contrast [6-8, 13]. The Chi-square dissimilarity defined in Section III.D and the nearest neighbourhood classifier were used for all methods here. The source codes of the proposed algorithm can be downloaded from http://www.comp.polyu.edu.hk/~cslzhang/code/CLBP.rar.

Table 1. Classification rate (%) on TC10 and TC12 using different schemes.

	$(P,R)=(8,1)$				$(P,R)=(16,2)$				$(P,R)=(24,3)$			
	TC10	TC12		Average	TC10	TC12		Average	TC10	TC12		Average
		"t"	"h"			"t"	"h"			"t"	"h"	
LTP	76.06	62.56	63.42	67.34	96.11	85.20	85.87	89.06	98.64	92.59	91.52	94.25
$VAR_{P,R}$	90.00	62.93	64.35	72.42	86.71	63.47	67.26	72.48	81.66	58.98	65.18	68.60
$LBP_{P,R}^{riu2}/VAR_{P,R}$	96.56	79.31	78.08	84.65	97.84	85.76	84.54	89.38	98.15	87.13	87.08	90.79
$CLBP\_S_{P,R}^{riu2}$	84.81	65.46	63.68	71.31	89.40	82.26	75.20	82.28	95.07	85.04	80.78	86.96
$CLBP\_M_{P,R}^{riu2}$	81.74	59.30	62.77	67.93	93.67	73.79	72.40	79.95	95.52	81.18	78.65	85.11
$CLBP\_M_{P,R}^{riu2}/C$	90.36	72.38	76.66	79.80	97.44	86.94	90.97	91.78	98.02	90.74	90.69	93.15
$CLBP\_S_{P,R}^{riu2}\_M_{P,R}^{riu2}/C$	94.53	81.87	82.52	86.30	98.02	90.99	91.08	93.36	98.33	94.05	92.40	94.92
$CLBP\_S_{P,R}^{riu2}/M_{P,R}^{riu2}$	94.66	82.75	83.14	86.85	97.89	90.55	91.11	93.18	99.32	93.58	93.35	95.41
$CLBP\_S_{P,R}^{riu2}/M_{P,R}^{riu2}/C$	96.56	90.30	92.29	93.05	98.72	93.54	93.91	95.39	98.93	95.32	94.53	96.26
VZ_MR8	93.59 (TC10), 92.55 (TC12, "t"), 92.82 (TC12, "h") (Average 92.99)											
VZ_Joint	92.00 (TC10), 91.41 (TC12, "t"), 92.06 (TC12, "h") (Average 91.82)											

B. Experimental results on the Outex Database

The Outex database includes two test suites: Outex_TC_00010 (TC10) and Outex_TC_00012 (TC12). The two test suites contain the same 24 classes of textures, which were collected under 3 different illuminants ("horizon", "inca", and "t184") and 9 different rotation angles (0^0, 5^0, 10^0, 15^0, 30^0, 45^0, 60^0, 75^0 and 90^0). There are 20 non-overlapping 128*128 texture samples for each class under each situation. The experiment setups are as follows:

1. For TC10, samples of illuminant "inca" and angle 0^0 in each class were used for classifier training and the other 8 rotation angles with the same illuminant were used for testing. Hence, there are 480 (24*20) models and 3,840 (24*8*20) validation samples.
2. For TC12, the classifier was trained with the same training samples as TC10, and it was tested with all samples captured under illuminant "t184" or "horizon". Hence, there are 480 (24*20) models and 4,320 (24*20*9) validation samples for each illuminant.

Table 1 lists the experimental results by different schemes. Under TC12, "t" represents the test setup of illuminant "t184" and "h" represents "horizon". We could make the following findings.

First, CLBP_S achieves much better result than CLBP_M in most cases. It is in accordance with our analysis in Section III.B that the sign component is more informative than the magnitude component. Second, the center pixel, which represents the gray level of the local patch, contains additional discriminant information as CLBP_M/C could get much better results than CLBP_M, and CLBP_S/M/C gets better results than CLBP_S/M. Third, because CLBP_S and CLBP_M/C contain complementary features, the classification accuracy could be much improved by fusing them either in concatenation or jointly. Between the two types of fusing methods, "CLBP_S/M/C" has better results than "CLBP_S_M/C". However, the latter has one advantage over the former: its feature dimension is much smaller. Take *P*=24 as an example, the feature sizes of "$CLBP\_S_{P,R}^{riu2}/M_{P,R}^{riu2}/C$" and "$CLBP\_S_{P,R}^{riu2}\_M_{P,R}^{riu2}/C$" are 1352 (26*26*2) and 78 (26+26*2), respectively. How to reduce the feature dimension of "CLBP_S/M/C" will be explored in our future research. Fortunately, a feature size of 1352 is not a big problem for implementation. Fourth, LTP has better results than CLBP_S as it is more robust to noise [21]. However, it could not compete with the fusion of CLBP_S and CLBP_M. One reason is that the local difference is quantized into three levels in LTP while it is quantized into four levels in CLBP. The other reason is that LTP is decomposed into one positive LBP and one negative LBP, which are however not totally independent of each other. On the contrary, the proposed CLBP_S and CLBP_M are independent and contain complementary information. Fifth, $CLBP\_M_{P,R}^{riu2}$ works better than $VAR_{P,R}$. This is because $VAR_{P,R}$ only measures the local intensity variance but neglects the local spatial structure, which is useful for texture discrimination. In contrast, the proposed CLBP_M operator exploits such information using binary coding.

Sixth, $CLBP\_S_{P,R}^{riu2}/M_{P,R}^{riu2}/C$ works better than VZ_Joint. For example, $CLBP\_S_{24,3}^{riu2}/M_{24,3}^{riu2}C$ can get 4% improvement over VZ_Joint. VZ-MRF, a variation of VZ_Joint, could improve a little the recognition accuracy but with much more cost on feature extraction and matching [7-8]. Both VZ_Joint and VZ-MRF use original local patch as the feature to learn the texton. They are simple to implement; however, it is time consuming to build the feature histogram. For example, VZ_Joint will spend more than 3.5 seconds on a 128*128 image by 7*7 patch (49 dimensions) to build the feature histogram (using a PC with Intel Dual Core E6550, 4G RAM, Windows XP and Matlab 7.6). The proposed CLBP scheme is much more efficient. It spends less than 0.1 second on a 128*128 image with (P, R)=(24,3), which is more than 35 times faster than VZ_Joint. Furthermore, CLBP is training free on feature extraction, and can still get good results when the training samples are limited.

Finally, $CLBP\_S_{P,R}^{riu2}/M_{P,R}^{riu2}/C$ achieves better and more robust results than the state-of-the-art methods $LBP_{P,R}^{riu2}/VAR_{P,R}$ and VZ_MR8. Particularly, it can have about 5% improvement over $LBP_{P,R}^{riu2}/VAR_{P,R}$ and 3% improvement over VZ_MR8 with (P,R)=(24,3). The improvement for TC12 is more obvious because $LBP_{P,R}^{riu2}/VAR_{P,R}$ is sensitive to illumination changes. When the illuminations for training and test sets are different, the quantization of VAR based on training set could not represent test set well, so the accuracy of $LBP_{P,R}^{riu2}/VAR_{P,R}$ drops quickly, by 11% for $LBP_{24,3}^{riu2}/VAR_{24,3}$ from TC10 to TC12. CLBP is training free and it is robust to illumination changes. For example, $CLBP\_S_{24,3}^{riu2}/M_{24,3}^{riu2}/C$ only drops 5% in classification accuracy from TC10 to TC12. This is a very important advantage in real applications because it is common for illumination variations.

By applying the multi-scale scheme defined in Section III.D, better results could be obtained. For example, $CLBP\_S/M/C_{8,1+16,2+24,3}$ could achieve 99.14%, 95.18% and 95.55% for TC10, TC12 "t" and TC12 "h" respectively. The proposed multi-scale scheme could be regarded as a simple sum fusion. Better performance can be expected if more advanced fusion techniques are used [26].

The proposed multi-scale CLBP is simple and fast to build the feature histogram; however, its feature size is a little higher than that of VZ_MR8. For example, the dimension of multi-scale $CLBP\_S/M/C_{8,1+16,2+24,3}$, which is the maximal dimension of CLBP in this section, is 2200 (26*26*2+18*18*2+10*10*2), while the size of VZ_MR8 is 960 (24*40). There are methods to reduce the number of models of each texture class, such as the greedy algorithm [6]. Usually, it is possible to get better accuracy by removing some outlier models [6].

Table 2. Classification rate (%) on CUReT using different schemes.

N	(P, R) = (8,1)				(P, R) = (16,3)				(P, R) = (24,5)			
	46	23	12	6	46	23	12	6	46	23	12	6
LTP	85.13	79.25	72.04	63.09	92.66	87.30	80.22	70.50	91.81	85.78	77.88	67.77
$VAR_{P,R}$	68.60	60.99	52.78	43.50	61.87	54.40	46.61	38.39	58.17	50.73	43.49	35.83
$LBP_{P,R}^{riu2}/VAR_{P,R}$	93.87	88.76	81.59	71.03	94.20	89.12	81.64	71.81	91.87	85.58	77.13	66.04
$CLBP\_S_{P,R}^{riu2}$	80.63	74.81	67.84	58.70	86.37	81.05	74.62	66.17	86.37	81.21	74.71	66.55
$CLBP\_M_{P,R}^{riu2}$	75.20	67.96	60.27	51.49	85.48	79.01	71.24	61.59	82.16	76.23	69.22	60.45
$CLBP\_M_{P,R}^{riu2}/C$	83.26	75.58	66.91	56.45	91.42	85.73	78.05	68.14	89.48	83.54	75.96	66.41
$CLBP\_S_{P,R}^{riu2}\_M_{P,R}^{riu2}/C$	90.34	84.52	76.42	66.31	93.87	89.05	82.46	72.51	93.22	88.37	81.44	72.01
$CLBP\_S_{P,R}^{riu2}/M_{P,R}^{riu2}$	93.52	88.67	81.95	72.30	94.45	90.40	84.17	75.39	93.63	89.14	82.47	73.26
$CLBP\_S_{P,R}^{riu2}/M_{P,R}^{riu2}/C$	95.59	91.35	84.92	74.80	95.86	92.13	86.15	77.04	94.74	90.33	83.82	74.46
VZ_MR8	97.79 (46), 95.03 (23), 90.48 (12), 82.90 (6)											
VZ_Joint	97.66 (46), 94.58 (23), 89.40 (12), 81.06 (6)											

C. Experimental results on the CUReT Database

The CUReT database contains 61 texture classes, each having 205 images acquired at different viewpoints and illumination orientations. There are 118 images shot from a viewing angle of less than 60^0. Of these 118 images, we selected 92 images, from which a sufficiently large region could be cropped (200*200) across all texture classes [6]. We converted all the cropped regions to grey scale.

To get statistically significant experimental results [10-11], N training images were randomly chosen from each class while the remaining 92-N images per class were used as the test set. The partition is implemented 1000 times independently. The average accuracy over 1000 randomly splits is listed in Table 2. The first 23 images of each class were used to learn the cut values for VAR, and were used to learn the texton dictionary for VZ_MR8 and VZ_Joint.

Similar conclusions to those in Section IV.B can be made from the experimental results on the CUReT dataset. The proposed $CLBP\_S_{P,R}^{riu2}/M_{P,R}^{riu2}/C$ gets better results than $LBP_{P,R}^{riu2}/VAR_{P,R}$ for all cases. Especially, when the number of training samples is small, bigger improvement is achieved. For example, $CLBP\_S_{24,5}^{riu2}/M_{24,5}^{riu2}/C$ has 3% higher accuracy than $LBP_{24,5}^{riu2}/VAR_{24,5}$ when N=46, while the difference is more than 8% when N=6. Again, the multi-scale scheme could improve the CLBP accuracy. For example, $CLBP\_S/M/C_{8,1+16,3+24,5}$ can get 97.39%, 94.19%, 88.72% and 79.88% for 46, 23, 12 and 6 training samples respectively. However, the performance of CLBP is not better than that of VZ_MR8 and VZ_Joint on this database. This is mainly because there are scale and affine variations in the CUReT database, while CLBP is an operator proposed for rotation and gray level invariance and it has limited capability to address scale and affine invariance. Meanwhile, VZ_MR8 and VZ_Joint are learning based algorithms, and large amount of training samples are necessary to construct the representative texton dictionary. When the training samples are not enough, the accuracy will drop. Table 3 shows the classification rate of VZ_MR8 under different number of training samples. We see that when the training samples are not enough, its performance is worse than the proposed CLBP operator. Similar finding could be found for the VZ_Joint algorithm.

As a completed operator of original LBP, the proposed framework could be applied to different LBP variants, such as derivative-based LBP [17], Dominant LBP [22], and center-symmetric LBP [23]. For example, apart from extracting dominant CLBP_S, dominant CLBP_M could be extracted as complementary information and concatenated with the dominant CLBP_S for classification.

Table 3. Classification rate (%) of VZ_MR8 using different number of training samples.

N	46	23	12	6
n=61	97.79	95.03	90.48	82.90
n=20	96.65	93.45	88.40	80.35
n=10	95.82	92.32	86.92	78.65

V. CONCLUSION

In this paper, we analyzed LBP from a viewpoint of local difference sign-magnitude transform (LDSMT), and consequently a new scheme, namely completed LBP (CLBP), was proposed. Three operators, CLBP_C, CLBP_S and CLBP_M, were defined to extract the image local gray level, the sign and magnitude features of local difference, respectively. We demonstrated that the sign component is more important than the magnitude component in preserving the local difference information, which can explain why the CLBP_S (i.e. conventional LBP) features are more effective than the CLBP_M features. Finally, by fusing the CLBP_S, CLBP_M and CLBP_C codes, all of which are in binary string format, either in a joint or in a hybrid way, much better texture classification accuracy than the state-of-the-arts LBP algorithms were obtained.

ACKNOWLEDGMENT

The authors sincerely thank MVG and VGG for sharing the source codes of LBP and VZ_MR8.

REFERENCES

[1] R.M. Haralik, K. Shanmugam, and I. Dinstein, "Texture features for image classification," *IEEE Trans. on Systems, Man, and Cybertics*, vol. 3, no. 6, pp. 610-621, 1973.

[2] T. Randen, and J.H. Husy, "Filtering for texture classification: a comparative study," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 291-310, 1999.

[3] R.L. Kashyap, and A. Khotanzed, "A model-based method for rotation invariant texture classification," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, no. 4, pp. 472-481, 1986.

[4] J.L. Chen, and A. Kundu, "Rotation and gray scale transform invariant texture identification using wavelet decomposition and hidden Markov model," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no. 2, pp. 208-214, 1994.

[5] H. Deng, and D.A. Clausi, "Gaussian VZ-MRF rotation-invariant features for image classification," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 951-955, 2004.

[6] M. Varma, and A. Zisserman, "A statistical approach to texture classification from single images," *International Journal of Computer Vision*, vol. 62, no. 1-2, pp. 61-81, 2005.

[7] M. Varma and A. Zisserman, "Texture classification: are filter banks necessary?" in *Proc. International Conference on Computer Vision and Pattern Recognition*, 2003, pp. 691-698.

[8] M. Varma, and A. Zisserrman, "A statistical approach to material classification using image patch examplars," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 2032-2047, 2009.

[9] Y. Xu, H. Ji, and C. Fermuller, "A projective invariant for texture," in *Proc. International Conference on Computer Vision and Pattern Recognition*, 2005, pp. 1932-1939.

[10] M. Varma, and R. Garg, "Locally invariant fractal features for statistical texture classification," in *Proc. International Conference on Computer Vision*, 2007, pp.1-8.

[11] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1265-1278, 2005.

[12] J. Zhang, M. Marszalek, S. Lazebnik and C. Schmid, "Local features and kernels for classification of texture and object categories: a comprehensive study," *International Journal of Computer Vision*, vol. 73, no. 2, pp. 213-238, 2007.

[13] T. Ojala, M. Pietikäinen, and T. T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with Local Binary Pattern," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, 2002.

[14] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, and S. Huovinen, "Outex – new framework for empirical evaluation of texture analysis algorithm," in *Proc. International Conference on Pattern Recognition*, 2002, pp. 701-706.

[15] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with Local Binary Patterns: application to face recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037-2041, 2006.

[16] G. Zhao, and M. Pietikäinen, "Dynamic texture recognition using Local Binary Patterns with an application to facial expressions," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 915-928, 2007.

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

[18] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51-59, 1996.

[19] T. Ojala, K. Valkealahti, E. Oja, and M. Pietikäinen, "Texture discrimination with multidimensional distributions of signed gray level differences," *Pattern Recognition*, vol. 34, no. 3, pp. 727-739, 2001.

[20] T Ahonen, and M. Pietikäinen, "Image description using joint distribution of filter bank responses," *Pattern Recognition Letters*, vol. 30, no. 4, pp. 368-376, 2009.

[21] X. Tan, and B. Triggs, "Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions," in *Proc. International Workshop on Analysis and Modeling of Faces and Gestures*, 2007, pp. 168-182.

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

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

[24] M. J. Weinberger, J. J. Rissanen, and R. B. Arps, "Applications of universal context modeling to lossless compression of gray-scale images," *IEEE Trans. on Image Processing*, vol. 5, no. 4, pp. 575-586, 1996.

[25] 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, 1999.

[26] K. Woods, W. P. Kegelmeyer Jr., and K. Bowyer, "Combination of Multiple Classifiers Using Local Accuracy Estimates," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 405-410, 1997.