

Robust Texture Image Representation by Scale Selective Local Binary Patterns

Zhenhua Guo, *Member, IEEE*, Xingzheng Wang, Jie Zhou, *Senior Member, IEEE*, and Jane You, *Member, IEEE*

Abstract—Local binary pattern (LBP) has successfully been used in computer vision and pattern recognition applications, such as texture recognition. It could effectively address grayscale and rotation variation. However, it failed to get desirable performance for texture classification with scale transformation. In this paper, a new method based on dominant LBP in scale space is proposed to address scale variation for texture classification. First, a scale space of a texture image is derived by a Gaussian filter. Then, a histogram of pre-learned dominant LBPs is built for each image in the scale space. Finally, for each pattern, the maximal frequency among different scales is considered as the scale invariant feature. Extensive experiments on five public texture databases (University of Illinois at Urbana-Champaign, Columbia Utrecht Database, Kungliga Tekniska Högskolan-Textures under varying Illumination, Pose and Scale, University of Maryland, and Amsterdam Library of Textures) validate the efficiency of the proposed feature extraction scheme. Coupled with the nearest subspace classifier, the proposed method could yield competitive results, which are 99.36%, 99.51%, 99.39%, 99.46%, and 99.71% for UIUC, CUReT, KTH-TIPS, UMD, and ALOT, respectively. Meanwhile, the proposed method inherits simple and efficient merits of LBP, for example, it could extract scale-robust feature for a 200 × 200 image within 0.24 s, which is applicable for many real-time applications.

Index Terms—Local binary pattern, scale selective, texture classification, nearest subspace classifier.

I. INTRODUCTION

TEXTURE classification is an active research topic in the fields of computer vision and pattern recognition. It has a wide range of applications, such as fabric inspection [1], remote sensing [2], and medical image analysis [3]. Early texture classification methods focus on the statistical

Manuscript received June 18, 2015; revised October 12, 2015; accepted December 4, 2015. Date of publication December 17, 2015; date of current version December 28, 2015. This work was supported in part by the Ministry of Education of China under Grant 20120002110033, in part by the Shenzhen Fundamental Research fund under Grant JCYJ20140509172959961, in part by the National Basic Research Program of China under Grant 2014CB349304, in part by the Shenzhen Oversea High Talent Innovation Fund under Grant KQCX20140521161756231, in part by the Hong Kong General Research Fund under Grant PolyU 152202/14E, in part by the Tsinghua University Initiative Scientific Research Program, and in part by the National Natural Science Foundation of China under Grant 61225008, Grant 61471213, Grant 6152271, and Grant 61527808. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Guoliang Fan.

Z. Guo and X. Wang are with the Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China (e-mail: zhenhua.guo@sz.tsinghua.edu.cn; xingzheng.wang@sz.tsinghua.edu.cn).

J. Zhou is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: jzhou@tsinghua.edu.cn).

J. You is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: csyjia@comp.polyu.edu.hk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2015.2507408

analysis of texture images. The representative ones include the co-occurrence matrix method [4] and the filtering based methods [5].

As the images may be captured under varying illumination and pose conditions, a good texture classification should address gray-scale, rotation and scale variations. In the early stage, many models were explored to get rotation and gray-scale invariant texture classification, such as autoregressive model [6], multi-resolution [7], hidden Markov model [8] and Gaussian Markov random field [9]. Compared to gray-scale and rotation invariance, scale invariance is much more difficult to achieve. Usually, people attempt to get local or global scale invariant features. The former scheme extracts scale invariant features from a pixel or a region first, then computes statistical features or signatures for the whole image; while the latter scheme extracts scale invariant feature from the whole image directly. In other words, local scale invariant features achieve it by normalizing a local region while global scale features achieve scale invariance by normalizing the whole image. Many local scale invariant feature extraction methods have been proposed. For example, Varma and Garg [10] extracted a local fractal vector for each pixel, then computed a statistical histogram; Han and Ma [46] summed all the Gabor filter impulse responses with different scales but along the same orientation for each pixel, then computed the first and second-order statistical features; Liu *et al.* [15] and Liu and Fieguth [16] applied random projection for densely sampled image patches, then extracted histogram or signature feature; instead of using all pixels, Lazebnik *et al.* [12] and Zhang *et al.* [13] detected Harris and Laplacian regions and extracted signatures after normalizing these regions. Recently, global scale invariant feature extraction methods drew attention because local scale normalization is usually slow. Yao and Sun normalized statistical edge feature distribution to resist variation in scale [19]. Pun and Lee used log-polar transform to eliminate the rotation and scale effects in the input image by converting the image into a corresponding log-polar image, after that they extracted wavelet energy features from the transformed image [47]. Hui *et al.* used Laplacian blob to estimate a global scale. After image normalization, they extracted wavelet spectrum feature [11]. Xu *et al.* [17], [18] and Quan *et al.* [45] classified the pixels in the image into multiple point sets by gray intensities or local feature descriptors. Then a global fractal dimension was computed for each point set and a vector was got by concatenating these fractal dimensions. Except extracting scale invariant features, pyramid histograms with shifting matching scheme were proposed by some scholars [14], [34].

However, the extracted features are not scale invariant and shift matching scheme is very time consuming.

In [20], Ojala et al. proposed to use the Local Binary Pattern (LBP) histogram for gray-scale and rotation invariant texture classification. LBP is a simple but efficient operator to describe the local image pattern. The LBP based methods have achieved impressive classification results on representative texture databases [21], and LBP has been deployed to many other applications, such as face recognition [22], dynamic texture recognition [23] and shape localization [24]. And many variants of LBP, including dominant LBP (DLBP) [25], derivative-based LBP [24], center-symmetric LBP [26] and completed LBP (CLBP) [27], have been proposed recently. But these methods could not address the scale variation issue. Recently, two methods based on LBP were proposed for solving this problem. Li et al. [28] proposed to find an optimal scale for each pixel and extracted LBP feature with the optimal scale. However, it failed to extract consistent and accurate scale for all pixels. So, it could not get good performance for complicated databases as shown in our experimental results. Quan et al. [45] proposed to extract global fractal feature based on multiscale LBP. However, global fractal feature are not robust when the image size is small [17]. To our best of knowledge, LBP feature alone could not get good performance for texture database with significant scale variation, such as UIUC [12] and ALOT [48] texture databases.

As shown by Liao et al. [25], statistically dominant local patterns play a critical role in texture classification. These dominant patterns with high frequency alone could provide more discriminant information. When an image is zoomed in or out, these patterns still exist but their feature radius will be enlarged or shrunk. Thus, to achieve scale invariance, it is intuitive to extract these patterns from a scale space of the image. Based on this assumption, we designed a new feature extraction scheme for LBP, scale selective LBP (SSLBP): first, a scale space of a texture image is derived by a Gaussian filter; then, a histogram of pre-learned dominant patterns is built for each image in the scale space; finally, for each pattern, the maximal frequency among different scales is considered as the scale invariant feature. Based on our previous proposed CLBP [27], even with simple Chi-square distance and the nearest neighbourhood classifier (NNC), the proposed feature could get good performance for texture databases with scale variation. If the classifier is replaced by an advanced classifier, such as the nearest subspace classifier (NSC), performance could be improved further and competitive results on UIUC [12], CUReT [29], KTH-TIPS [30], UMD [17] and ALOT [48] texture databases could be achieved.

The main contributions of this paper could be summarized as follows:

- 1) We proposed a simple and effective method to address scale variation issue for texture image classification.
- 2) The proposed feature extraction method takes less than 0.24 seconds for an image with 200×200 pixels, which is fast enough for many real time applications.
- 3) We empirically found that the LBP with scale selection can perform well for difficult texture databases, such as UIUC and ALOT, but traditional LBP cannot.

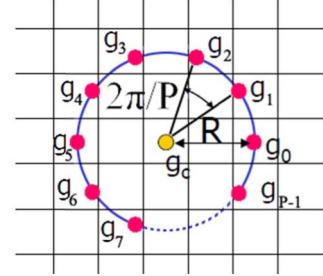


Fig. 1. Central pixel and its P circularly and evenly spaced neighbours with radius R .

The rest of the paper is organized as follows. Section II briefly reviews CLBP which is the fundamental operator for the proposed method. Section III presents the proposed scale selective feature extraction scheme in detail. Section IV reports extensive experimental results and Section V concludes the paper.

II. BRIEF REVIEW OF CLBP

The proposed method uses CLBP to extract scale sensitive features first, and then applies a scale selective scheme to get SSLBP. Thus, the fundamental CLBP is briefly introduced.

CLBP [27] is a gray-scale texture operator that characterizes the local spatial structure of the image texture. Referring to Fig. 1, given a central pixel g_c and its P circularly and evenly spaced neighbours g_p with radius R , $p = 0, 1, \dots, P - 1$, we can simply calculate the difference between g_c and g_p as $d_p = g_p - g_c$. d_p can be further decomposed into two components:

$$d_p = s_p \times m_p \text{ and } \begin{cases} s_p = \text{sign}(d_p) \\ m_p = |d_p| \end{cases} \quad (1)$$

where $s_p = \begin{cases} 1, d_p \geq 0 \\ -1, d_p < 0 \end{cases}$ is the sign and m_p is the magnitude of d_p .

Thus, three operators are defined as:

$$CLBP_{SP,R} = \sum_{p=0}^{P-1} t(s_p, 0) 2^p \quad (2)$$

$$CLBP_{MP,R} = \sum_{p=0}^{P-1} t(m_p, c_m) 2^p \quad (3)$$

$$CLBP_{CP,R} = t(g_c, c_I) \quad (4)$$

where t is a threshold function $t(x, c) = \begin{cases} 1, x \geq c \\ 0, x < c \end{cases}$. c_m is the mean value of m_p for the image. c_I is the average gray level of the image. It is obvious that both $CLBP_{SP,R}$ and $CLBP_{MP,R}$ can have 2^P different values. As $CLBP_C$ contains only one bit, it can have 2 kinds of values and it is rotation invariant.

To remove the effect of rotation, i.e. to assign a unique identifier to each rotation invariant pattern, rotation invariant $CLBP_S$ and $CLBP_M$ are defined as:

$$\begin{aligned} CLBP_{SP,R}^{ri} &= \min\{ROR(CLBP_{SP,R}, i)\} \\ &\quad i = 0, 1, \dots, P - 1 \\ CLBP_{MP,R}^{ri} &= \min\{ROR(CLBP_{MP,R}, i)\} \\ &\quad i = 0, 1, \dots, P - 1 \end{aligned} \quad (5)$$

Algorithm 1 Determining the Dominant Patterns Across Different Scales for $CLBP_S_{P,R}^{ri}/C$

Input: The training image set, $T = \{f_i | i = 1, 2, \dots, N\}$. f_i is a training image and N is the size of the training set. K is the number of dominant patterns to be learned for $CLBP_S/C$. L is the size of the scale space for a given image. g_σ is the 2D Gaussian filter with a standard deviation σ to build the scale space.

Output: K dominant patterns for $CLBP_S/C$.

Procedure:

Step 1. Initialize one pattern histogram for the training set, $H^T_{CLBP_S/C}[0, 1, \dots, HS-1] = 0$, where HS is the size of the rotation invariant histogram (its value is 36×2 , 4116×2 and 699252×2 when $P=8, 16$ and 24 respectively);

Step 2. Derive a scale space for image f_i based on g_σ :

$$s_l = \begin{cases} f_i & , l=1 \\ s_{l-1} * g_\sigma, & 1 < l \leq L, \text{ ``*'' is the convolution operator} \end{cases}$$

Step 3. For s_l , build one pattern histogram, $H^{s_l}_{CLBP_S/C}$ based on $CLBP_S/C$;

Step 4. One new histograms for image f_i is built:

$$H^{f_i}_{CLBP_S/C}(k) = \max(H^{s_1}_{CLBP_S/C}(k), H^{s_2}_{CLBP_S/C}(k), \dots, H^{s_L}_{CLBP_S/C}(k))$$

where $0 \leq k \leq HS-1$;

Step 5. Accumulate the value of one new histogram to $H^T_{CLBP_S/C}$:

$$H^T_{CLBP_S/C}(k) = H^T_{CLBP_S/C}(k) + \frac{H^{f_i}_{CLBP_S/C}(k)}{N}$$

Step 6. Go back to Step 2, until all training samples are processed.

Step 7. Sort $H^T_{CLBP_S/C}$ in descending order;

Step 8. Return K patterns $DP^T_{CLBP_S/C}[1, 2, \dots, K]$ corresponding to the first K maximal frequency for the sorted histogram.

where $ROR(x, i)$ performs a circular bit-wise right shift on the P -bit number x i times. The number of $CLBP_S_{P,R}^{ri}$ ($CLBP_M_{P,R}^{ri}$) is much smaller than $CLBP_S_{P,R}$ ($CLBP_M_{P,R}$). For example, the number of $CLBP_S_{8,R}^{ri}$, $CLBP_S_{16,R}^{ri}$ and $CLBP_S_{24,R}^{ri}$ are 36, 4116 and 699252 respectively.

As CLBP_S, CLBP_M and CLBP_C produce binary strings and contain complementary information, a 2D joint histogram “CLBP_S/C” or “CLBP_M/C” could be built to get more information and avoid the huge dimension issue [27].

III. SCALE INVARIANT FEATURE EXTRACTION AND MATCHING

A. Feature Extraction Scheme

As found by Liao *et al.* [25], statistical dominant local patterns provide much of discriminant information for texture classification. When an image is zoomed in or out, a kind of dominant pattern still exists but it will occupy a larger or smaller image region. In other words, percentage of this pattern in the image does not change, but its characteristic scale does. Thus, if we can find its characteristic scale and extract its percentage at that scale, scale invariance could be achieved. Based on this assumption, in this study, we designed a novel and simple scale invariant feature extraction scheme for LBP by finding dominant patterns from scale spaces.

A training stage is applied to find dominant patterns by analyzing scale space of a training set. First, given a training sample, a scale space is derived by a 2D Gaussian filter. Then, for every image in the scale space, a local pattern histogram is built. For each pattern, the maximal frequency among different scales is kept to build a new histogram. This histogram is used as the scale invariant feature for the given training sample. Finally, some dominant patterns with high average frequency throughout the whole training set are selected. Our method extracts frequency information from learned and fixed patterns, so the pattern type containing important information is maintained. For example, for two images I_1 and I_2 , our method will generate features like this: $[\text{Frequency}^1(P_A), \text{Frequency}^1(P_B)]$ and $[\text{Frequency}^2(P_A), \text{Frequency}^2(P_B)]$, where P_A and P_B are two kinds of dominant patterns. However, DLBP [25] cares frequency information only and does not consider the pattern type. For the two example images mentioned above, output of DLBP may be $[\text{Frequency}^1(P_A), \text{Frequency}^1(P_B)]$ and $[\text{Frequency}^2(P_B), \text{Frequency}^2(P_A)]$. The pseudo codes on determining the dominant patterns across different scales for $CLBP_S_{P,R}^{ri}/C$ are presented in Algorithm 1. R and P , two parameters for extracting local binary patterns are fixed in Algorithm 1, so they are omitted here for readability. For the same reason, ri is omitted. Dominant patterns across different scales for $CLBP_M_{P,R}^{ri}/C$ are determined similarly.

Algorithm 2 Extracting a Scale Selective LBP Histogram Feature by $CLBP_{S^r_i/C}$

Input: I is a training or a test image. L and g_σ are parameters defined in Algorithm 1. $DP^T_{CLBP_S/C}[1,2,\dots,K]$ are pre-learned dominant patterns by Algorithm 1.

Output: A scale invariant histogram feature by $CLBP_S/C$ for image I .

Procedure:

Step 1. Derive a scale space for image I based on g_σ :

$$s_l = \begin{cases} I & , l=1 \\ s_{l-1} * g_\sigma, & 1 < l \leq L \end{cases} \text{, } ** \text{ is the convolution operator}$$

Step 2. For s_l , build one pattern histograms, $DPH_{CLBP_S/C}^{s_l}$ based on $CLBP_S/C$. Here, for each pixel, compute the pattern label based on $CLBP_S/C$ first, if its pattern label does not belong to $DP^T_{CLBP_S/C}[1,2,\dots,K]$, this pixel will not contribute to $DPH_{CLBP_S/C}^{s_l}$;

Step 3. A scale selective feature for image I is built:

$$DPH_{CLBP_S/C}^I(k) = \max(DPH_{CLBP_S/C}^{s_1}(k), DPH_{CLBP_S/C}^{s_2}(k), \dots, DPH_{CLBP_S/C}^{s_L}(k))$$

where $1 \leq k \leq K$.

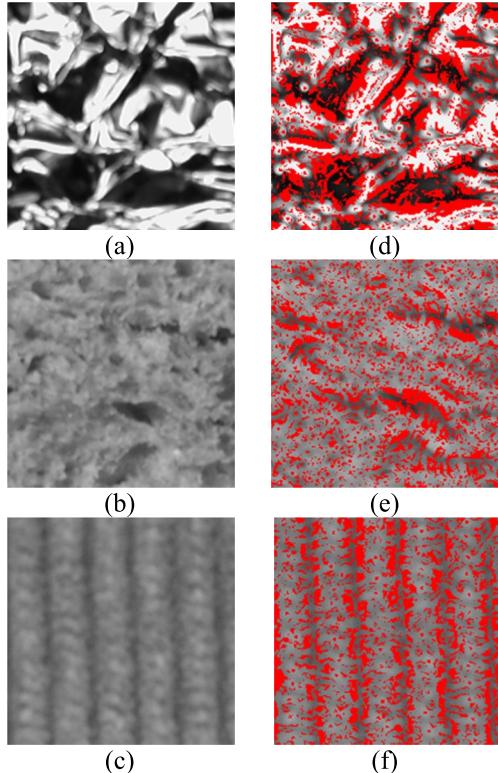


Fig. 2. An illustration example for the first 5 dominant patterns by Algorithm 1. a-c) are three texture images from KTH-TIPS[30]; d-f) show the distribution of first 5 dominant patterns, for each pixel, if there exist one of the first 5 dominant patterns selected by Algorithm 1, it is marked with red color.

Fig. 2 shows distribution of the first 5 dominant patterns by Algorithm 1. As illustrated in Fig. 2, most of dominant patterns distribute in local region with big variation. After finding dominant patterns by analyzing the whole training set, the scale invariant feature is extracted for both training and test sets. First, given a sample, a scale space is derived by the same 2D Gaussian filter. Then, for every image in the scale space, a local pattern histogram

containing only pre-learned dominant patterns is built. Finally, for each dominant pattern, the maximal frequency across different scales is selected to build a new feature histogram. This histogram is used as the scale invariant feature for the given sample. The pseudo codes on extracting scale selective features by $CLBP_{S^r_i/C}$ are shown in Algorithm 2. Here, R , P and ri are omitted for readability. Scale selective features by $CLBP_{M^r_i/C}$ are extracted similarly. Then, two features are concatenated: $DPH_{CLBP_S^r_i/C_M^r_i/C} = [DPH_{CLBP_S^r_i/C}, DPH_{CLBP_M^r_i/C}]$.

Fig. 3 shows an illustration example for SSLBP feature extraction. Here, we would like to emphasize two major differences with DLBP [25]: 1. The dominant patterns are selected from two dimensions, the scale space and the feature space, while DLBP are determined by the feature space only; 2. The pattern information is kept. Thus, only the frequencies with specified patterns are extracted, while DLBP extract maximal frequencies among all patterns and ignore pattern information.

Traditionally, people devoted to extract local [10], [12], [13], [15], [16], [46] or global [11], [17]–[19], [45], [47] scale invariant features. From implementation viewpoint, what we proposed belongs to an intermediate way. We first extracted the local scale variant feature, then, we applied a global transformation to achieve scale invariance. On the other hand, from scale space viewpoint, we proposed a new method. Instead of analyzing scale spaces locally, such as on a pixel [28] or a region [12], we analyzed scale space globally, by extracting global statistical features across different scales.

B. Feature Matching

In this paper, two kinds of classifiers are implemented. First, similar to traditional LBP methods, the NNC with the chi-square distance [27] is used to show the effectiveness of the proposed feature extraction scheme:

$$D(Y, Z) = \sum_{x=1}^X (Y_x - Z_x)^2 / (Y_x + Z_x) \quad (6)$$

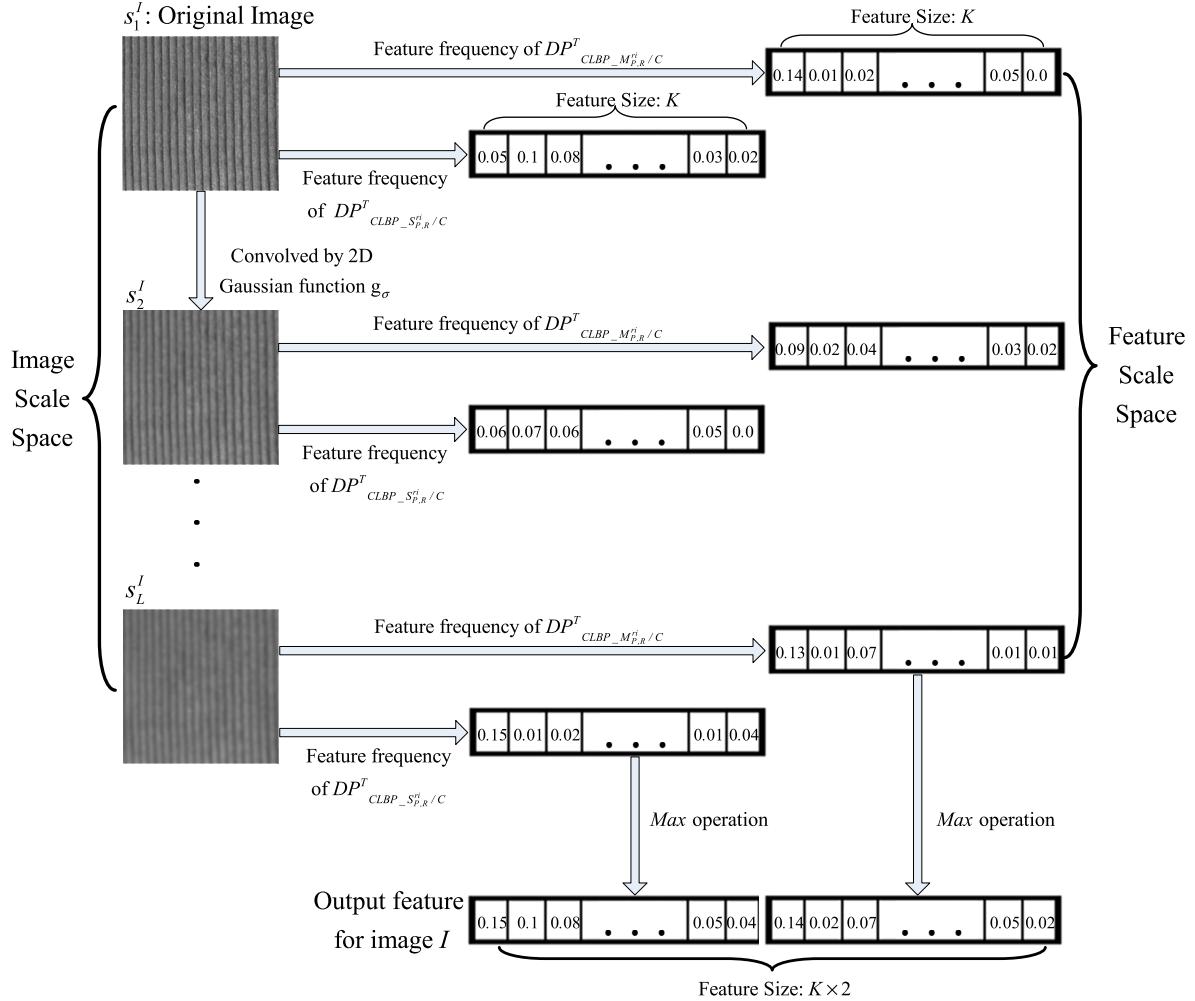


Fig. 3. An illustration example for Algorithm 2.

Where Y and Z are extracted features of a test sample and a model sample, Y is assigned to the class of model Z that minimizes the chi-square distance.

Then, an advanced classifier, NSC [37], is implemented to further improve performance. After getting frequency of the dominant patterns, to avoid over emphasizing patterns with large frequency, similar to [38], a preprocessing step is applied to the proposed feature before NSC:

$$\overline{H_k} = \text{sqrt}(H_k), \quad k = 1, 2, \dots, K \quad (7)$$

where K is the number of bins, sqrt is the mathematics operation to get square root and H_k is the original frequency of the dominant patterns at k^{th} bin.

Suppose there are C classes of textures in the database, and there are n training samples in each class. $H_c = [h_{c,1}, h_{c,2}, \dots, h_{c,n}]$ denotes a set of histograms for one class. For a test texture image y , we first build its histogram feature, denoted by h^y . We project h^y into the subspace spanned by H_c as follows:

$$\rho_c = (H_c^T H_c)^{-1} H_c^T h^y \quad (8)$$

And the projection residuals can be computed as:

$$err_c = \|h^y - H_c \rho_c\|_2 \quad (9)$$

Scale number	Relative scale	Distance to camera (cm)
1	$2^{-1.00} = 0.500$	14.00
2	$2^{-0.75} = 0.595$	16.65
3	$2^{-0.50} = 0.707$	19.80
4	$2^{-0.25} = 0.841$	23.55
5	$2^{0.00} = 1.000$	28.00
6	$2^{+0.25} = 1.189$	33.30
7	$2^{+0.50} = 1.414$	39.60
8	$2^{+0.75} = 1.682$	47.09
9	$2^{+1.00} = 2.000$	56.00

Fig. 4. Scale parameters for KTH-TIPS database.

Then, we classify y to the class with the minimal residual.

C. Scale Analysis of LBP

In this section, we use KTH-TIPS to illustrate why the proposed scheme is effective under scale variation. KTH-TIPS [30] is an image database containing 10 kinds of materials and each material was captured by 9 different scales, 3 different poses and 3 different illuminations. As shown in Fig. 4, each image is collected in a controlled distance.

Fig. 5 shows an example to illustrate the effectiveness of scale invariance by scale space analysis.

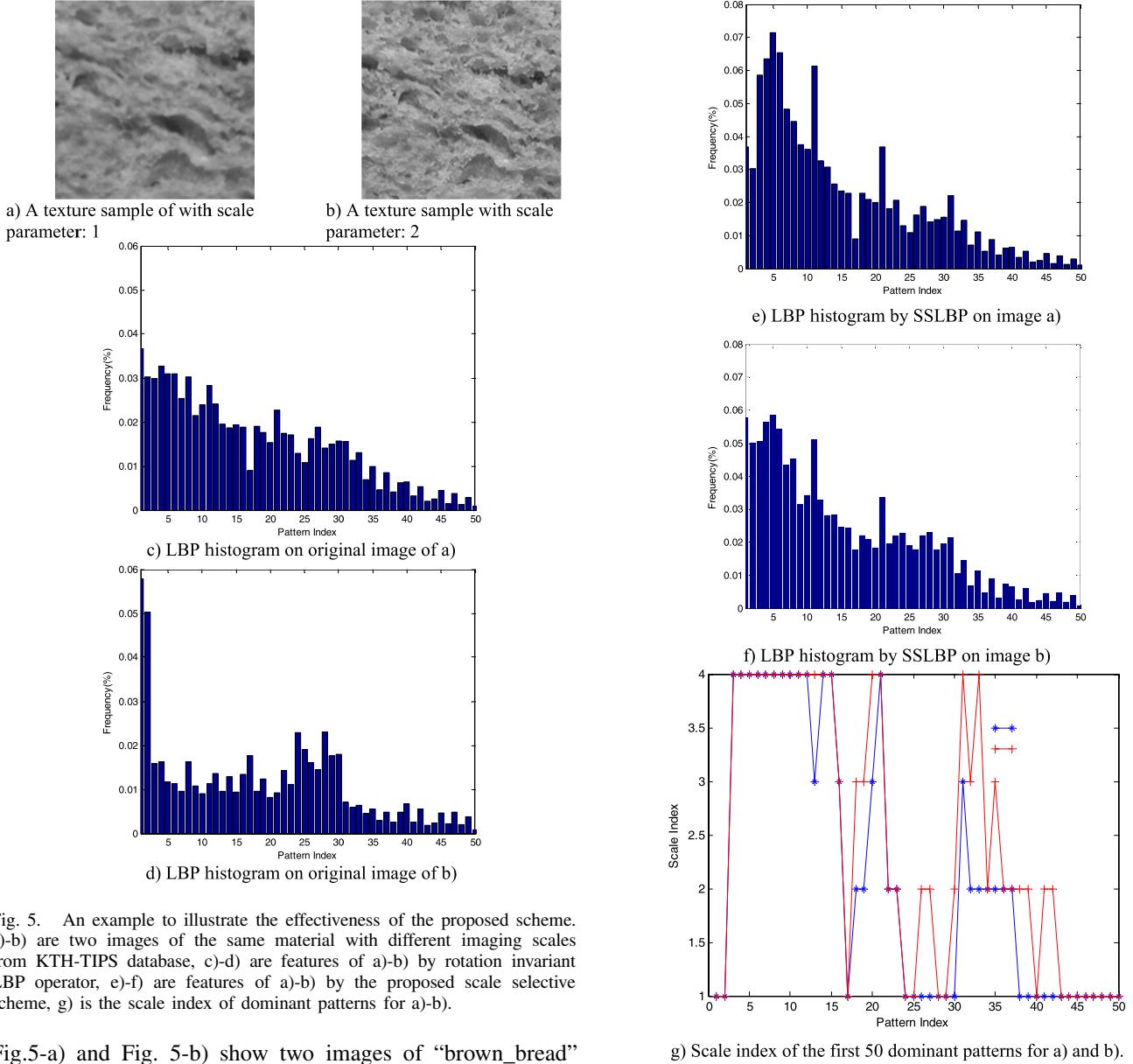


Fig. 5. An example to illustrate the effectiveness of the proposed scheme. a)-b) are two images of the same material with different imaging scales from KTH-TIPS database, c)-d) are features of a)-b) by rotation invariant LBP operator, e)-f) are features of a)-b) by the proposed scale selective scheme, g) is the scale index of dominant patterns for a)-b).

Fig. 5-a) and Fig. 5-b) show two images of “brown_bread” in KTH-TIPS with different scale parameters. Fig. 5-c) and Fig. 5-d) are traditional LBP histograms without scale invariance, while Fig. 5-e) and Fig. 5-f) are LBP features extracted by the proposed scheme. For better visualization, only 50 patterns are selected. The distance between Fig. 5-c) and Fig. 5-d) is 0.3552, while the distance between Fig. 5-e) and Fig. 5-f) is much smaller, 0.1778 only. L and σ are set to 4 and $2^{0.25}$ here. In Step 3 of Algorithm 2, for a given pattern k , scale index is defined as the number corresponding to the maximal frequency among L values:

$$\text{ScaleIndex}(k) = \arg \max_l (DPH_{CLBP_S/C}^{SI}(k)), \quad l = 1, 2, \dots, L \quad (10)$$

Fig. 5-g) plots scale index for image Fig. 5-a) and Fig. 5-b). As shown in Fig. 5-g), the difference of scale index of many patterns is just 1, which is also coincided with the scale difference between these two images.

To further illustrate the relationship between the feature scale of LBPs and the image scale, we did one

Fig. 5. (Continued.) An example to illustrate the effectiveness of the proposed scheme. e)-f) are features of a)-b) by the proposed scale selective scheme, g) is the scale index of dominant patterns for a)-b).

more experiment. We divided KTH-TIPS into 9 image sets (IS_z , $z = 1, 2, \dots, 9$) according to scale parameters, so each set contains 90 images ($10^3 \times 3^3$). Then average feature scale for each set is computed as:

$$AFS_z = \frac{\sum_{I \in IS_z} FS_I}{90}, \quad z = 1, 2, \dots, 9 \quad (11)$$

$$FS_I = \frac{\sum_{k=1}^K \text{ScaleIndex}_I(k)}{K} \quad (12)$$

Where $\text{ScaleIndex}_I(k)$ is computed by Eq. (10) for a given image I .

AFS for different sets is plotted in Fig. 6. As shown in it, the average feature scale increases as the image scale increases, thus scale invariance could be achieved by selecting maximal

TABLE I
A BRIEF INTRODUCTION OF FIVE PUBLIC TEXTURE DATABASES

Texture Database Name	Imaging property	Image Size	Number of classes	Number of samples per class
CUReT [29]	It is designed to contain large intraclass variation and is widely used to assess classification performance. The images are captured under different illumination and viewing directions. 92 images from which a sufficiently large region could be cropped (200×200) across all texture classes are selected [32]. All the cropped regions are converted to gray-scale.	Fixed, 200×200	61	92
KTH-TIPS [30]	It extends CUReT by imaging new samples of ten of the CUReT textures at a subset of the viewing and lighting angles used in CUReT but also over a range of scales. We followed Zhang <i>et al.</i> [13] in treating it as a stand-alone dataset.	Varied, 196×201 (average)	10	81
UIUC [12]	It has been designed to require local invariance. Textures are acquired under significant scale and viewpoint changes, arbitrary rotations, and uncontrolled illumination conditions, even including textures with non-rigid deformation.	Fixed, 640×480	25	40
UMD [17]	It has been designed in a similar way as UIUC, while the image resolution is 4 times of UIUC.	Fixed, 1280×960	25	40
ALOT [48]	ALOT is recorded for scientific purposes. It is systematically collected with varied viewing angles, illumination angles, and illumination colors for each material which is similar in spirit to the CUReT collection. Although the number of view-illumination directions per material is only half of the BRDF resolution of CUReT, ALOT extends the number of materials almost by a factor of 5, and it improves upon image resolution and color quality. All images are converted to gray-scale.	Varied, 1536×891 (average)	250	100

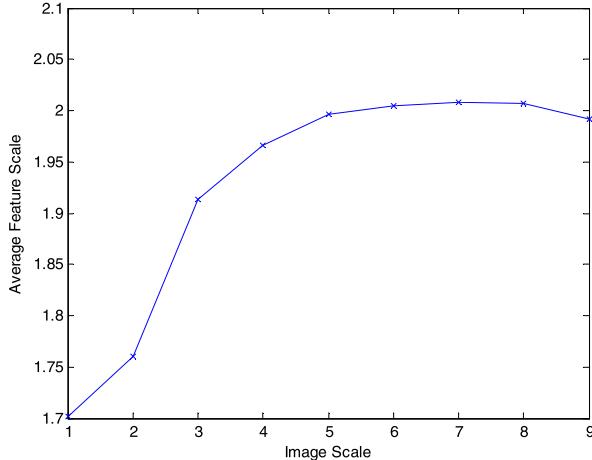


Fig. 6. Average feature scale for images with the same scale parameter.

frequency among scale spaces, which is the essence of the proposed method.

IV. EXPERIMENT AND DISCUSSION

A. Texture Databases

To evaluate the effectiveness of the proposed method, we carried out a series of experiments on five large and commonly used texture databases: UIUC [12], CUReT [29], [32], KTH-TIPS [30], UMD [17] and ALOT [48] texture databases. A brief introduction of these databases is shown in Table I and Fig. 7 shows several samples of each database. Detailed information of each database could be found in Ref. [12], [17], [29], [30], [48].

B. Experimental Results

There are several parameters in the proposed method. P is set to 24 as this value could provide good discriminant

features and get much better results than 8 or 16, while large neighbourhood value will increase the computation burden significantly. Similar to [34], σ is set to $2^{0.25}$. And L is set to 4 empirically. How to select these two parameters adaptively will be our future work. The multiresolution analysis could be used to improve the classification accuracy, by employing multiple operators of various ($24, R$). In this study, we used two radii: 3 and 9. To get a feature length comparable with other methods [10], [31], [32], K is set 600 here, thus the feature length of one image is 2400. Besides the simple and intuitive NNC, we applied one advanced classifier NSC [37] on the proposed feature to improve performance. We chose NNC and NSC because both methods are easy to be implemented and parameter free, unlike the widely used support vector machine (SVM) classifier, so we can focus more on feature extraction. More advanced feature extraction and classifier [51]–[56] will be studied in our future work.

In the following, except our method, we implemented original $LB P_{8,1}^{riu2}$, multiscale $LB P_{P,R}^{riu2}(8, 1+16, 3+24, 5)$ [20], multiscale $CLBP_S_{P,R}^{riu2}/M_{P,R}^{riu2}/C(8, 1+16, 3+24, 5)$ [27], $DLBP_{R=3,P=24}$ [25] and scale- and rotation-invariant $LB P_{8,R(x,y)}^{sri-su2}$ [28] with NNC. The image sample was normalized to have an average intensity of 0 and a standard deviation of 1 [31], [32]. The first quarter samples of each class were used to learn the dominant patterns for the proposed method. To get statistically significant experimental results [31], [32], half of the samples per class are randomly selected for training and the remaining half for testing. The partition is implemented 1000 times independently. The average accuracy over 1000 random splits is listed in Table II. Table II also gives a comprehensive summary of the result for SSLBP with 20 recent state-of-the-art results. Classifiers for different methods are shown in bracket.

Several findings could be noted from Table II. First, with the simple NNC, the proposed feature could get good results.

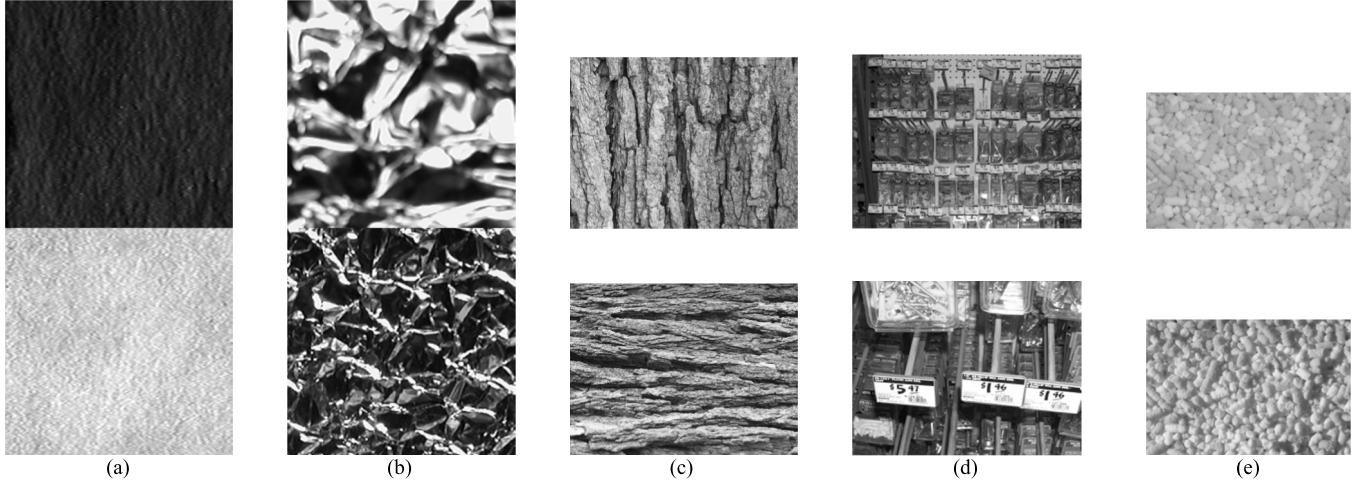


Fig. 7. Several image samples of each database. a) Two sample images of the first class from CUReT database; b) Two sample images of the first class from KTH-TIPS database; c) Two sample images of the first class from UIUC database; d) Two sample images of the first class from UMD database; e) Two sample images of the first class from ALOT database.

TABLE II

EXPERIMENTAL RESULTS ON FIVE PUBLIC TEXTURE DATABASES (%). FOR METHOD 1-20, RESULTS ARE ORIGINALLY REPORTED, EXCEPT FOR THOSE MARKED (*) WHICH ARE TAKEN FROM THE COMPARATIVE STUDY IN ZHANG et al. [13]. THE BRACKETED NUMBERS ARE THE NUMBER OF THE TRAINING SAMPLES PER CLASS USED FOR THE CORRESPONDING DATABASE

No.	Method	CUReT (46)	KTH-TIPS (40)	UIUC (20)	UMD (20)	ALOT (50)
1	SRP [15] (SVM)	99.37	99.29	98.56	99.30	-
2	RP [16] (NNC)	98.52	97.71	96.27	99.13	-
3	VZ-MR8 [32] (NNC)	97.43	-	-	-	-
4	VZ Patch [31] (NNC)	98.03	92.4(*)	97.83	-	-
5	Caputo et al. [36] (SVM)	98.46	94.8(*)	92.0(*)	-	-
6	Lazebnik et al. [12] (NNC)	72.5(*)	91.3(*)	96.03	-	-
7	Zhang et al. [13] (SVM)	95.3	96.1	98.7	-	-
8	Varma and Ray [33] (SVM)	-	-	98.9	-	-
9	Varma and Garg [10] (NNC)	97.5	-	95.4	-	-
10	BIF [34] (Shift Matching NNC)	98.6	98.5	98.8	-	-
11	MFS [17] (NNC)	-	-	92.74	93.93	85.64
12	OTF [18] (SVM)	-	-	97.44	98.42	95.6
13	WMFS [11] (SVM)	-	-	97.62	98.68	96.94
14	PRICO-LBP [35] (SVM)	98.4	98.4	-	-	-
15	LEP [14] (Shift Matching NNC)	-	97.56	-	-	-
16	scLBP[41] (SVM)	99.29	-	98.45	99.25	-
17	COV-LBPD[42] (NNC)	-	98.0	-	-	-
18	PLS [43] (SVM)	-	98.4	96.57	98.99	93.35
19	RLBP[44] (NNC)	-	-	96.7	-	-
20	PFS [45] (SVM)	-	97.35	97.92	99.38	97.5
21	$LBP_{8,1}^{riu2}$ [20] (NNC)	80.63	82.67	55.26	88.23	63.33
22	$DLBP_{R=3,P=24}$ [25] (NNC)	84.93	86.99	60.73	89.87	78.38
23	$LBP_{8,R(x,y)}^{sri_su2}$ [28] (NNC)	85.00	89.73	70.05	91.71	71.29
24	$LBP_{P,R}^{riu2}$ (8,1+16,3+24,5) [20] (NNC)	95.84	95.17	76.88	94.36	87.99
25	$CLBP - S_{P,R}^{riu2} / M_{P,R}^{riu2} / C(8,1+16,3+24,5)$ [27] (NNC)	97.40	97.19	93.26	98.00	93.28
26	SSLBP (NNC)	98.55	97.80	97.02	98.84	96.69
27	SSLBP (NSC)	99.51	99.39	99.36	99.46	99.71

When an advanced classifier is chosen, such as NSC, performance could be improved and our approach scores very well on five commonly used datasets, producing the best result on these databases. For a given test sample, NSC uses all of the training samples of the same class while NNC uses one training sample only. As correlation between different samples

could be used, NSC can be more robust to outlier and noise. That is why NSC can get much better results than NNC for the same feature extraction method. It needs to be emphasized that the same parameters ($\sigma = 2^{0.25}$, $L = 4$, $R = [3, 9]$, $P = 24$, $K = 600$) are used on these five databases, although specific parameter tuning could improve the performance further.

Second, by NNC, DLBP can get better results than original LBP, but its performance is much poorer than the proposed method. This is mainly because DLBP do not consider scale variation, so it cannot get good results in these difficult datasets. However, its performance can be improved if an advanced classifier, such as SVM, is used [25].

Third, previously proposed scale-invariant feature $LBP_{8,R(x,y)}^{sri-su2}$ could get better results than $LBP_{8,1}^{riu2}$. However, it was worse than $LBP_{P,R}^{riu2}(8, 1+16, 3+24, 5)$ and $CLBP_{S,P,R}^{riu2}/M_{P,R}^{riu2}/C(8, 1+16, 3+24, 5)$, and much worse than the proposed method. This is mainly because it is hard to extract consistent and accurate scale for all pixels. Thus, $LBP_{8,R(x,y)}^{sri-su2}$ could get sound performance on some environment controlled databases [28], but failed to get good performance on more complicated databases.

Fourth, although the images are captured by varied conditions, sometime, when the training sample per class are enough and comprehensive, even rotation invariant and gray-scale invariant operator could provide good performance, such as $CLBP_{S,P,R}^{riu2}/M_{P,R}^{riu2}/C(8, 1+16, 3+24, 5)$ could get sound performance on CUReT, KTH-TIPS and UMD datasets.

Fifth, although BIF [34] could get better result than the proposed method with NNC, this is mainly because BIF uses pyramid histograms with shifting matching schemes. For example, BIF could get only 96% for UIUC dataset without the shift matching scheme [34].

Last but not least, previously proposed LBP methods could not get good results on UIUC and ALOT databases. The best accuracy is only 93.26% and 93.28% for UIUC and ALOT respectively, while our method could get 99.36% and 99.71% accuracy for these challenging databases, it validates the effectiveness of the proposed scale selective feature by finding dominant patterns across scale spaces.

C. Image Resolution Test

As LBP operator is based on a local region, unlike some methods [17], [45] requiring high resolution, the proposed method could get good results on different resolutions of the same images. To show the robustness to different image resolutions, three experiments are implemented on UMD database whose resolution is high and fixed.

For each image of UMD, it is resized to 1/2 and 1/4 of its original size. Then, the proposed feature extraction is applied on three different resolutions. And NT images were randomly chosen from each class as the training set while the remaining 40- NT images per class were used as the test set. The partition is implemented 1000 times independently. The average accuracy and standard deviation over 1000 random splits is listed in Table III.

As shown in Table III, the proposed feature extraction method could still get very good results when the image resolution is reduced. It shows that the number of training samples plays an important role when the image size is small, as the classification rate decreases significantly when the number of training samples is small. For example, when the image size is only 320×240 , its classification rate is still 99.38% for $NT=20$; while NT drops to 5, its classification decreases

TABLE III
CLASSIFICATION RATE (% , MEAN+STANDARD DEVIATION) OF UMD
BY DIFFERENT NUMBER OF TRAINING SAMPLES AND IMAGE
RESOLUTION WITH SSLBP AND NSC

<i>Image size NT</i>	20	15	10	5
1280×960	99.46±0.46	99.31±0.51	98.81±0.70	96.41±1.28
640×480	99.71±0.26	99.48±0.37	98.77±0.68	95.68±1.45
320×240	99.38±0.45	98.80±0.69	97.41±1.06	92.74±1.84

TABLE IV
A COMPARISON OF COMPUTATIONAL COMPLEXITY BETWEEN RP,
VZ_PATCH AND SSLBP FOR ONE TEST IMAGE

Method	Scale Space Building	Pattern/ Patch Processing	Histogramming	Classification (NNC)
RP[1]	-	$S_p \cdot S_R \cdot I_p$	$S_R \cdot C \cdot S_T \cdot I_p$	$C \cdot S_T \cdot (C \cdot T_n)$
VZ_Patch [31]	-	-	$S_p \cdot C \cdot S_T \cdot I_p$	$C \cdot S_T \cdot (C \cdot T_n)$
SSLBP	$(L-1) \cdot S_g \cdot I_p$	$2 \cdot L \cdot P \cdot I_p$	$2 \cdot L \cdot I_p$	$4 \cdot K \cdot (C \cdot T_n)$

Here $L=4$ is the size of scale space, $P=24$ is the number of neighbours for LBP, I_p denotes the number of pixels per image sample, S_g denotes the size of Gaussian smooth kernel, $K=600$ is the number of selected dominant patterns. S_p represents the size of a local patch, usually 7×7 [31], S_R is the dimension of random projection, usually 15[1], C denotes the number of classes, S_T represents the number of clustered textons per class, here $C \cdot S_T \approx 4 \cdot K$. T_n is the number of training samples per class.

to 92.74% which is around 4% lower than its original image size and more than 6% lower than $NT=20$. This is mainly because when the image size is reduced, some detailed information will lose, so the possibility that two images from different classes look similar will increase. It indicates that for some applications, if the image resolution is limited, enlarging the training set is a good way to compensate the influence of low resolution. It is interesting to point out that the classification rate increases sometime when the image is resized to 1/2 of its original size. For example, in the proposed method, it could get 99.71% when $NT=20$. This is probably because some noise is removed when an image is resized.

D. Time Cost

The proposed method inherits the simple and efficient property of LBP. Here, computation complexity between RP [1] and VZ_Patch [31] are shown in Table IV. As texton dictionary [1], [31] and dominant patterns could be learned off-line, they will not influence the speed for classifying a given image. Here we only list key steps for three methods. As listed in Table IV, VZ_Patch takes the longest time for feature extraction, and the complexity of the proposed method is much smaller than RP and VZ_Patch. As shown in Table II, SRP [15] is the method that gets the closest result with our proposed method. In fact, it is an improvement of the method RP, and needs an extra operation of sorting grey values of a local region, so its complexity is even higher than RP.

Usually, scholars pay more attention on how to achieve better classification rate, thus it is often neglected whether the proposed method could be applied for real time application. To illustrate that the proposed method is fast enough for

TABLE V
AVERAGE TIME COST FOR ONE IMAGE ON FOUR DATABASES

	CURET	KTH-TIPS	UIUC	UMD	ALOT
Feature extraction of MFS [17] (Unit: Second)	0.09	0.08	0.62	2.60	2.67
Feature extraction of VZ_MR8 [32] (Unit: Second)	1.03	0.93	9.98	37.35	40.11
Feature extraction of VZ_Patch [31] ($S_p = 7 \times 7$) (Unit: Second)	12.44	11.28	96.97	309.51	346.64
Feature extraction of the proposed scheme (Unit: Second)	0.24	0.23	1.80	7.63	8.46
Matching (NNC) (Unit: Millisecond)	177.47	25.09	30.78	31.1	297.42
Matching (NSC) (Unit: Millisecond)	2.25	0.64	0.90	0.92	5.05

real time applications, Table V listed the average time cost for one image on five databases by the proposed method with several methods. The test is based on a HP desktop, with Intel i7-3770 3.4 GHz CPU, 4GB RAM, and Matlab R2010b. As shown in Table V, the complexity of the proposed method is linearly related with image size and the number of training samples. It could extract scale invariant features for a 200×200 image within 0.24 seconds, which is much faster than VZ_Patch [31] (here, a KDTree [39] search algorithm is used to speed up feature extraction), while the latter takes about 12.44 seconds. As discussed above, SR and SRP is a little faster than VZ_Patch, so they may not be applicable for some real-time vital applications neither. Although the proposed method is not the fastest method, it takes longer time than MFS [17] but it could process a middle-size image in real time, such as images of CURET, KTH-TIPS and UIUC. Although, it takes a little longer for images of UMD and ALOT, as discussed in Section IV.C, a large image could be down-sampled before feature extraction. Since the code is not fully optimized and not parallelized, how to optimize the code for large texture image and implement the proposed method on GPU to improve speed will be our future work.

E. Dominant Patterns Analysis

As shown in Section III.A, the proposed method needs to select some dominant patterns from the training samples; otherwise, the dimension of scale invariant LBP feature will be too huge. For example, feature size of $CLBP_S_{P,R}^i/C$ and $CLBP_M_{P,R}^i/C$ is 699252×2 when $P = 24$. It loses a very precious advantage of traditional CLBP, training free. However, the number of possible candidates is much smaller than those statistical textons [1], [15], [31], [32] (taking VZ_Patch with 7×7 patch [31] as an example, there are maximally 256^{49} possible textons). Thus, the selected patterns are similar among different training sets. Table VI shows the percentage of identical dominant patterns between different training sets. As shown in it, although the training images are different, the selected patterns are very similar.

Furthermore, the proposed method focuses on the frequency of those dominant patterns only, so it is robust to training sets. To validate it, some cross database experiments are carried out. Those dominant patterns selected by CURET database

TABLE VI
PERCENTAGE OF IDENTICAL DOMINANT PATTERNS
BETWEEN DIFFERENT TRAINING SETS

	CURET	KTH-TIPS	UMD	UIUC
CURET	100%	80.33%	73%	74.54%
KTH-TIPS	-	100%	76.12%	80.16%
UMD	-	-	100%	87.87%
UIUC	-	-	-	100%

TABLE VII
CLASSIFICATION RATE (% , MEAN+STANDARD DEVIATION)
OF CURET BY DIFFERENT DOMINANT PATTERNS WITH
THE PROPOSED FEATURE AND NSC

NT	46	23	12	6
Patterns Learned by CURET Database	99.51\pm0.37	98.28\pm1.17	95.18\pm2.63	88.19\pm4.57
Patterns Learned by KTH-TIPS Database	99.49 \pm 0.30	98.20 \pm 1.10	94.98 \pm 2.65	87.67 \pm 4.48
Patterns Learned by UIUC Database	99.34 \pm 0.36	97.97 \pm 1.24	94.52 \pm 2.65	87.64 \pm 4.39
Patterns Learned by UMD Database	99.40 \pm 0.31	98.04 \pm 1.10	94.72 \pm 2.64	87.34 \pm 4.47
Patterns Learned by SUN Database	98.09 \pm 0.56	95.11 \pm 1.26	88.68 \pm 2.43	76.38 \pm 3.50

are used for KTH-TIPS, UMD and UIUC database, in other words Algorithm 1 in Section III.A is omitted for KTH-TIPS, UMD and UIUC database and a pre-defined dominant pattern set is used to extract scale invariant LBP features. Similarly, dominant patterns selected by KTH-TIPS, UMD and UIUC databases are used to extract features for other databases. More challengingly, we use Algorithm 1 to select some dominant patterns from a scene database, SUN database [40]. 2400 dominant patterns are selected by 2 scene images of each class, totally 397×2 images from SUN database. Then, these selected dominant patterns are used to extract features for these CURET, KTH-TIPS, UMD and UIUC texture databases.

TABLE VIII

CLASSIFICATION RATE (% , MEAN+STANDARD DEVIATION)
OF KTH-TIPS BY DIFFERENT DOMINANT PATTERNS
WITH THE PROPOSED FEATURE AND NSC

<i>NT</i>	40	20	10	5
Patterns Learned by KTH-TIPS Database	99.39 \pm 0.85	97.21 \pm 1.93	91.58 \pm 3.40	81.54 \pm 5.54
Patterns Learned by CUReT Database	99.39 \pm 0.87	97.12 \pm 2.04	91.49 \pm 3.41	82.04 \pm 5.01
Patterns Learned by UIUC Database	98.54 \pm 0.87	95.88 \pm 1.93	90.28 \pm 3.27	80.52 \pm 5.09
Patterns Learned by UMD Database	98.26 \pm 0.93	95.48 \pm 2.01	89.72 \pm 3.16	80.55 \pm 5.04
Patterns Learned by SUN Database	96.72 \pm 1.27	92.40 \pm 2.40	84.86 \pm 3.45	75.23 \pm 4.51

TABLE IX

CLASSIFICATION RATE (% , MEAN+STANDARD DEVIATION)
OF UIUC BY DIFFERENT DOMINANT PATTERNS WITH
THE PROPOSED FEATURE AND NSC

<i>NT</i>	20	15	10	5
Patterns Learned by UIUC Database	99.36 \pm 0.37	98.87 \pm 0.47	97.53 \pm 0.76	92.41 \pm 1.49
Patterns Learned by CUReT Database	99.35 \pm 0.37	98.87 \pm 0.46	97.53 \pm 0.77	92.32 \pm 1.50
Patterns Learned by KTH-TIPS Database	99.40 \pm 0.36	98.91 \pm 0.43	97.50 \pm 0.75	92.46 \pm 1.58
Patterns Learned by UMD Database	99.40 \pm 0.35	98.86 \pm 0.48	97.50 \pm 0.76	92.40 \pm 1.52
Patterns Learned by SUN Database	99.03 \pm 0.50	98.47 \pm 0.57	97.06 \pm 0.80	92.11 \pm 1.55

Similar to Section IV.C, *NT* images were randomly chosen from each class as the training set while the remaining 92-*NT*, 81-*NT*, 40-*NT*, 40-*NT*, images per class were used as the test set for CUReT, KTH-TIPS, UMD and UIUC respectively. The partition is implemented 1000 times independently. The average accuracy and standard deviation over 1000 random splits are computed and listed in TableVII-X.

As shown in these tables, except the pattern set learnt by SUN database for CUReT and KTH-TIPS, there is no significant difference even with a pre-defined dominant pattern set. It is a good merit compared with those statistical textons based methods [1], [15], [31], [32], which are sensitive to the training set and require large amount of training samples to construct the representative textons dictionary [27]. And this merit is very useful for practical applications, because getting enough and representative training sets is not a trivial issue. For CUReT and KTH-TIPS databases, the patterns

TABLE X

CLASSIFICATION RATE (% , MEAN+STANDARD DEVIATION)
OF UMD BY DIFFERENT DOMINANT PATTERNS WITH
THE PROPOSED FEATURE AND NSC

<i>NT</i>	20	15	10	5
Patterns Learned by UMD Database	99.46 \pm 0.46	99.31 \pm 0.51	98.81 \pm 0.70	96.41 \pm 1.28
Patterns Learned by CUReT Database	99.44 \pm 0.47	99.22 \pm 0.57	98.76 \pm 0.72	96.22 \pm 1.38
Patterns Learned by KTH-TIPS Database	99.46 \pm 0.46	99.24 \pm 0.56	98.80 \pm 0.71	96.27 \pm 1.39
Patterns Learned by UIUC Database	99.45 \pm 0.50	99.29 \pm 0.55	98.82 \pm 0.69	96.33 \pm 1.34
Patterns Learned by SUN Database	99.48 \pm 0.45	99.29 \pm 0.53	98.80 \pm 0.69	96.34 \pm 1.39

learned by SUN database could not get very good recognition accuracy. This is probably because CUReT and KTH-TIPS were collected by controlled environments, so the patterns learnt from uncontrolled scenes may fail to represent the controlled textures very well.

V. CONCLUSION

In this study, we proposed a novel and simple scale invariant texture classification: scale invariance is achieved by analyzing the scale space of LBPs. We demonstrated that the proposed method could get very promising results on five public texture databases. And the proposed method is fast enough for many real time applications. The main contribution of this study lies in two aspects. First, LBP is a simple but efficient operator to address rotation and gray-scale invariance. However, it could not get good result on images with scale variation. Our study empirically shows that with a proper scheme, LBP operator can address scale invariance as well. Second, we proposed a new but efficient method to address scale invariance by extracting globally statistical features from scale space. Different from traditional local scale invariant feature extraction methods [10], [12], [13], [15], [16], [28], [46], we do not need to estimate local scale and spend time on the time-consuming normalization of the local region, here we extract scale sensitive local features and apply a global operation to achieve scale invariance. Meanwhile, achieving scale invariance by analyzing multiple histograms of local feature is a new global scale invariant feature extraction scheme. It is apparently different from traditional schemes such as extracting statistical features from one histogram [19], normalizing one image globally [11], [47] or computing fractal dimension globally [17], [18], [45]. The proposed method can be used for many potential applications, such as material classification [30], defect detection [49] and medical image retrieval [50].

Although the proposed method could get very good performance on five public texture databases, the underlying

relationship between images and parameters of the proposed method are not fully studied. We believe it will greatly help us understand texture images if the relationship could be unveiled. Furthermore, it is not explored whether the proposed scheme suits for other texture operators, such as BIF [34]. This is another direction of our future work. The proposed method will be also explored for new applications [57]–[59] in our future work.

ACKNOWLEDGMENT

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

REFERENCES

- [1] F. S. Cohen, Z. Fan, and S. Attali, “Automated inspection of textile fabrics using textural models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 8, pp. 803–808, Aug. 1991.
- [2] H. Anyis and D.-C. He, “Evaluation of textural and multipolarization radar features for crop classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 33, no. 5, pp. 1170–1181, Sep. 1995.
- [3] Q. Ji, J. Engel, and E. Craine, “Texture analysis for classification of cervix lesions,” *IEEE Trans. Med. Imag.*, vol. 19, no. 11, pp. 1144–1149, Nov. 2000.
- [4] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural features for image classification,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.
- [5] T. Randen and J. H. Husoy, “Filtering for texture classification: A comparative study,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 4, pp. 291–310, Apr. 1999.
- [6] R. L. Kashyap and A. Khotanzad, “A model-based method for rotation invariant texture classification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 4, pp. 472–481, Jul. 1986.
- [7] J. Mao and A. K. Jain, “Texture classification and segmentation using multiresolution simultaneous autoregressive models,” *Pattern Recognit.*, vol. 25, no. 2, pp. 173–188, 1992.
- [8] J.-L. Chen and A. Kundu, “Rotation and gray scale transform invariant texture identification using wavelet decomposition and hidden Markov model,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 2, pp. 208–214, Feb. 1994.
- [9] H. Deng and D. A. Clausi, “Gaussian MRF rotation-invariant features for image classification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 7, pp. 951–955, Jul. 2004.
- [10] M. Varma and R. Garg, “Locally invariant fractal features for statistical texture classification,” in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.
- [11] H. Ji, X. Yang, H. Ling, and Y. Xu, “Wavelet domain multifractal analysis for static and dynamic texture classification,” *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 286–299, Jan. 2013.
- [12] S. Lazebnik, C. Schmid, and J. Ponce, “A sparse texture representation using local affine regions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1265–1278, Aug. 2005.
- [13] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, “Local features and kernels for classification of texture and object categories: A comprehensive study,” *Int. J. Comput. Vis.*, vol. 73, no. 2, pp. 213–238, 2007.
- [14] J. Zhang, J. Liang, and H. Zhao, “Local energy pattern for texture classification using self-adaptive quantization thresholds,” *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 31–42, Jan. 2013.
- [15] L. Liu, P. Fieguth, G. Kuang, and H. Zha, “Sorted random projections for robust texture classification,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 391–398.
- [16] L. Liu and P. W. Fieguth, “Texture classification from random features,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 574–586, Mar. 2012.
- [17] Y. Xu, H. Ji, and C. Fermüller, “Viewpoint invariant texture description using fractal analysis,” *Int. J. Comput. Vis.*, vol. 83, no. 1, pp. 85–100, 2009.
- [18] Y. Xu, S. Huang, H. Ji, and C. Fermüller, “Scale-space texture description on SIFT-like textons,” *Comput. Vis. Image Understand.*, vol. 116, no. 9, pp. 999–1013, 2012.
- [19] C.-H. Yao and S.-Y. Chen, “Retrieval of translated, rotated and scaled color textures,” *Pattern Recognit.*, vol. 36, no. 4, pp. 913–929, 2003.
- [20] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
- [21] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllonen, and S. Huovinen, “Outex—New framework for empirical evaluation of texture analysis algorithms,” in *Proc. 16th Int. Conf. Pattern Recognit.*, 2002, pp. 701–706.
- [22] T. Ahonen, A. Hadid, and M. Pietikäinen, “Face description with local binary patterns: Application to face recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.
- [23] G. Zhao and M. Pietikäinen, “Dynamic texture recognition using local binary patterns with an application to facial expressions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 915–928, Jun. 2007.
- [24] X. Huang, S. Z. Li, and Y. Wang, “Shape localization based on statistical method using extended local binary pattern,” in *Proc. 3rd Int. Conf. Image Graph.*, 2004, pp. 184–187.
- [25] S. Liao, M. W. K. Law, and A. C. S. Chung, “Dominant local binary patterns for texture classification,” *IEEE Trans. Image Process.*, vol. 18, no. 5, pp. 1107–1118, May 2009.
- [26] M. Heikkilä, M. Pietikäinen, and C. Schmid, “Description of interest regions with local binary patterns,” *Pattern Recognit.*, vol. 42, no. 3, pp. 425–436, 2009.
- [27] Z. Guo, L. Zhang, and D. Zhang, “A completed modeling of local binary pattern operator for texture classification,” *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1657–1663, Jun. 2010.
- [28] Z. Li, G. Liu, Y. Yang, and J. You, “Scale- and rotation-invariant local binary pattern using scale-adaptive texton and subuniform-based circular shift,” *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2130–2140, Apr. 2012.
- [29] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink, “Reflectance and texture of real-world surfaces,” *ACM Trans. Graph.*, vol. 18, no. 1, pp. 1–34, 1999.
- [30] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh, “On the significance of real-world conditions for material classification,” in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 253–266.
- [31] M. Varma and A. Zisserman, “A statistical approach to material classification using image patch exemplars,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 11, pp. 2032–2047, Nov. 2011.
- [32] M. Varma and A. Zisserman, “A statistical approach to texture classification from single images,” *Int. J. Comput. Vis.*, vol. 62, nos. 1–2, pp. 61–81, 2005.
- [33] M. Varma and D. Ray, “Learning the discriminative power-invariance trade-off,” in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.
- [34] M. Crosier and L. D. Griffin, “Using basic image features for texture classification,” *Int. J. Comput. Vis.*, vol. 88, no. 3, pp. 447–460, 2010.
- [35] X. Qi, R. Xiao, C.-G. Li, Y. Qiao, J. Guo, and X. Tang, “Pairwise rotation invariant co-occurrence local binary pattern,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2199–2213, Nov. 2014.
- [36] B. Caputo, E. Hayman, M. Fritz, and J.-O. Eklundh, “Classifying materials in the real world,” *Image Vis. Comput.*, vol. 28, no. 1, pp. 150–163, 2010.
- [37] K.-C. Lee, J. Ho, and D. Kriegman, “Acquiring linear subspaces for face recognition under variable lighting,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 684–698, May 2005.
- [38] O. Barkan, J. Weill, L. Wolf, and H. Aronowitz, “Fast high dimensional vector multiplication face recognition,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1960–1967.
- [39] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2008.
- [40] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, “SUN database: Large-scale scene recognition from abbey to zoo,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3485–3492.
- [41] J. Ryu, S. Hong, and H. S. Yang, “Sorted consecutive local binary pattern for texture classification,” *IEEE Trans. Image Process.*, vol. 24, no. 7, pp. 2254–2265, Jul. 2015.
- [42] X. Hong, G. Zhao, M. Pietikäinen, and X. Chen, “Combining LBP difference and feature correlation for texture description,” *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2557–2568, Jun. 2014.
- [43] Y. Quan, Y. Xu, Y. Sun, and Y. Luo, “Lacunarity analysis on image patterns for texture classification,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 160–167.

- [44] J. Chen, V. Kellokumpu, G. Zhao, and M. Pietikäinen, “RLBP: Robust local binary pattern,” in *Proc. Brit. Mach. Vis. Conf.*, 2013, pp. 122.1–122.10.
- [45] Y. Quan, Y. Xu, and Y. Sun, “A distinct and compact texture descriptor,” *Image Vis. Comput.*, vol. 32, no. 4, pp. 250–259, 2014.
- [46] J. Han and K.-K. Ma, “Rotation-invariant and scale-invariant Gabor features for texture image retrieval,” *Image Vis. Comput.*, vol. 25, no. 9, pp. 1474–1481, 2007.
- [47] C.-M. Pun and M.-C. Lee, “Log-polar wavelet energy signatures for rotation and scale invariant texture classification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 590–603, May 2003.
- [48] G. J. Burghouts and J.-M. Geusebroek, “Material-specific adaptation of color invariant features,” *Pattern Recognit. Lett.*, vol. 30, no. 3, pp. 306–313, 2009.
- [49] F. Tajeripour, E. Kabir, and A. Sheikhi, “Fabric defect detection using modified local binary patterns,” *EURASIP J. Adv. Signal Process.*, vol. 2008, Nov. 2007, Art. ID 783898.
- [50] L. Nanni, A. Lumini, and S. Brahnam, “Local binary patterns variants as texture descriptors for medical image analysis,” *Artif. Intell. Med.*, vol. 49, no. 2, pp. 117–125, 2010.
- [51] B. Gu, V. S. Sheng, K. Y. Tay, W. Romano, and S. Li, “Incremental support vector learning for ordinal regression,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1403–1416, Jul. 2015.
- [52] B. Gu, V. S. Sheng, Z. Wang, D. Ho, S. Osman, and S. Li, “Incremental learning for v-support vector regression,” *Neural Netw.*, vol. 67, pp. 140–150, Jul. 2015.
- [53] Z. Lai, W. K. Wong, Y. Xu, J. Yang, and D. Zhang, “Approximate orthogonal sparse embedding for dimensionality reduction,” *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2015.2422994.
- [54] Y. Lu, Z. Lai, Y. Xu, X. Li, D. Zhang, and C. Yuan, “Low-rank preserving projections,” *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2015.2457611.
- [55] Y. Chen, W. Xie, and X. Zou, “A binary differential evolution algorithm learning from explored solutions,” *Neurocomputing*, vol. 149, pp. 1038–1047, Feb. 2015.
- [56] P. Zhu, Q. Hu, W. Zuo, and M. Yang, “Multi-granularity distance metric learning via neighborhood granule margin maximization,” *Inf. Sci.*, vol. 282, no. 1, pp. 321–331, 2014.
- [57] F. Liu, D. Zhang, and L. Shen, “Study on novel curvature features for 3D fingerprint recognition,” *Neurocomputing*, vol. 168, no. 1, pp. 599–608, 2015.
- [58] B. Chen, H. Shu, G. Coatrieux, G. Chen, X. Sun, and J. L. Coatrieux, “Color image analysis by quaternion-type moments,” *J. Math. Imag. Vis.*, vol. 51, no. 1, pp. 124–144, 2015.
- [59] J. Liu, B. Zhang, H. Zeng, L. Shen, J. Liu, and J. Zhao, “The BeiHang keystroke dynamics systems, databases and baselines,” *Neurocomputing*, vol. 144, no. 1, pp. 271–281, 2014.



Zhenhua Guo received the M.S. degree in computer science from the Harbin Institute of Technology, in 2004, and the Ph.D. degree in computer science from The Hong Kong Polytechnic University, in 2010. Since 2010, he has been with the Graduate School at Shenzhen, Tsinghua University. His research interests include pattern recognition, texture classification, biometrics, and video surveillance.

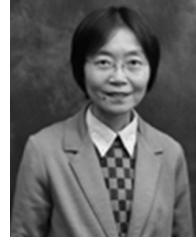


Xingzheng Wang received the B.Sc. degree in mechanical engineering and automation from the North China University of Technology, Beijing, China, in 2004, the M.Sc. degree in mechanical engineering from Tsinghua University, Beijing, in 2007, and the Ph.D. degree in computer science from The Hong Kong Polytechnic University, Hong Kong, in 2013.

He has been a Post-Doctoral Fellow with the Graduate School at Shenzhen, Tsinghua University, since 2013. His research interests include computational photography, computer vision, and pattern recognition.



Jie Zhou (M’01–SM’04) received the B.S. and M.S. degrees from the Department of Mathematics, Nankai University, Tianjin, China, in 1990 and 1992, respectively, and the Ph.D. degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, China, in 1995. Since 1997, he has served as a Post-Doctoral Fellow with the Department of Automation, Tsinghua University, Beijing, China. Since 2003, he has been a Full Professor with the Department of Automation, Tsinghua University. He is the Head of the Department of Automation with Tsinghua University. His research interests include computer vision, pattern recognition, and image processing. In recent years, he has authored over 100 papers in peer-reviewed journals and conferences. Among them, more than 40 papers have been published in top journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, and CVPR. He is an Associate Editor of the *International Journal of Robotics and Automation* and two other journals. He received the National Outstanding Youth Foundation of China Award.



Jane You received the B.Eng. degree in electronics engineering from Xi’an Jiaotong University, in 1986, and the Ph.D. degree in computer science from La Trobe University, Australia, in 1992. She was a Lecturer with the University of South Australia and a Senior Lecturer with Griffith University from 1993 to 2002. She is currently a Professor with The Hong Kong Polytechnic University. Her research interests include image processing, pattern recognition, medical imaging, biometrics computing, multimedia systems, and data mining.