A Distinct and Compact Texture Descriptor

Yuhui Quan, Yong Xu*, Yuping Sun

School of Computer Science & Engineering,
South China University of Technology, Guangzhou 510006, China

Abstract

In this paper, a statistical approach to static texture description is developed, which combines a local pattern coding strategy with a robust global descriptor to achieve highly discriminative power, invariance to photometric transformation and strong robustness against geometric changes. Built upon the *local binary patterns* that are encoded at multiple scales, a statistical descriptor, called *pattern fractal spectrum*, characterizes the self-similar behavior of the local pattern distributions by calculating fractal dimension on each type of pattern. Compared with other fractal-based approaches, the proposed descriptor is compact, highly distinctive and computationally efficient. We applied the descriptor to texture classification. Our method has demonstrated excellent performance in comparison with state-of-the-art approaches on four challenging benchmark datasets.

Key words:

Texture description, Local binary pattern, Fractal dimension, Multi-fractal analysis, Texture classification

1. Introduction

Visual texture, a powerful cue for characterizing scene structures that exhibit a high degree of similarity in their image patterns, provides a rich semantic description for many computer vision tasks such as image understanding, scene classification and visual retrieval. Although there is an abundant literature on texture description, developing an effective and efficient texture descriptor remains a challenging problem. A desired texture descriptor should be highly discriminative and robust to environmental changes, such as illumination changes, occlusions, non-rigid surface

*Corresponding author.

Email addresses: yuhui.quan@mail.scut.edu.cn (Yuhui Quan), yxu@scut.edu.cn (Yong Xu), sun.yp@mail.scut.edu.cn (Yuping Sun)

distortions and camera viewpoint changes. Meanwhile, the descriptor should be compact and computationally feasible for real-world applications.

In the past, a number of approaches [1, 2, 3, 4, 5, 6, 7, 8, 9] for texture description have been proposed to achieve robustness against environmental changes. The basic idea of these approaches is to compute a global description based on some local texture features. One popular approach to local texture description is the so-called *local binary pattern* (LBP) [10]. The LBP and its variants [11, 12, 13, 14, 15, 16] have been demonstrated effective in describing local image patterns and classifying materials. However, the performance of the existing LBP-based methods in recognizing random texture in real scene is very limited. The reason is that most of these methods use simple histogram-based statistics to integrate local features. Such histogram-based statistics only count the number of elements and lose the details of how the elements are spatially distributed.

One promising alternative to the histogram-based statistics is the fractal-based approach [17, 18, 19, 20, 21, 22] that characterizes the spatial distribution of elements using fractal geometry. Many fractal-based approaches have been successfully used in texture classification. However, in the existing approaches, often the features are highly redundant. For instance, the gradient and Laplacian multi-fractal measurements used in [19, 23] would provide similar information about edges, and the descriptor proposed in [18] is insufficiently compact (1218-dimensional). Some approaches involve large-scale clustering (e.g., [17]) or some complex alignment stages (e.g., a tight frame system is involved in [22]) and as a result, the computational cost becomes expensive.

In this paper, we are motivated to combine the power of the local pattern coding strategy and the global fractal analysis. Built upon the local binary patterns of image, a practical fractal-based texture descriptor is developed. An input image is first represented by local binary patterns, and then fractal dimension is computed on each type of pattern and all the fractal dimensions are finally concatenated as a feature vector. Owing to the effective LBP-based local representation, the proposed descriptor becomes compact. Meanwhile, benefitting from the robust fractal analysis, the proposed approach enjoys both high discriminative power and robustness against many environmental changes. We applied our descriptor to texture classification on four challenging benchmark datasets. The proposed approach has exhibited excellent performance in both effectiveness and efficiency in comparison with state-of-the-art approaches.

The remainder of this paper is organized as follows. Sec. 2 reviews related works on texture descriptor for texture classification. Sec. 3 introduces some background knowledge about local binary patterns and fractal analysis. Sec. 4 is devoted to the proposed approach. The experimental evaluations are reported in Sec. 5 and the conclusion is drawn in Sec. 6.

2. Related work

In this section, we first have a brief review on recent studies of texture classification that are most relevant to our work. Secondly, some LBP-based approaches to texture classification are mentioned. Finally, we focus on the fractal-based approaches.

Texture classification. Many modern texture classification approaches ([2, 5, 6, 24, 25, 26]) model texture patterns using the statistics of spatial texton distribution. The basic idea is to extract local features through random sampling or robust feature detection and construct a texton codebook by performing vector quantization on a set of randomly selected local features. Then some statistical measurements (mostly histogram-based) are computed with the help of the texton codebook. The seminal work can be traced back to the approach introduced by Julesz [27] and such an approach has been studied by many other researchers (e.g., [28, 29]). The main advantage of such methods is that the local feature descriptors have strong robustness to geometric and illumination changes, as well as partial occlusions. One representative work is the texture representation approach proposed by Lazebnik et al. [30]. The approach first characterizes the texture by clusters of elliptic regions, and the elliptic regions are then normalized to circles for the invariance to affine transforms. Two types of descriptors, spin image and RIFT (Rotation Invariant Feature Transform), are defined on each region. The resulting texture descriptor is the histogram of clusters of these local descriptors, and different texture descriptors are compared using the earth mover's distance [31]. This approach was comprehensively evaluated in [26] and has demonstrated promising performance on the applications of texture classification and retrieval on several benchmark datasets.

LBP-based approach. The concept of LBP is first introduced by Ojala et al. [10] and has been widely used as local feature for texture description. In [12], multi-scale LBPs were used for texture classification. The texture description is based on the joint distribution of the LBPs. Liao and Chung [13, 32] adapted the LBP coding strategy to the local dominant structures of texture by sorting the patterns according to their occurrence frequencies. The distribution of the LBPs is characterized by the co-occurrence statistics. Guo et al.[14] proposed to weight the LBP histogram by the local variance of the LBP codes. The weighted histogram is aligned to the principle orientation of image and matched by exhaustive search. In [15], the magnitudes as well as the signs of local image differences are utilized in the LBP coding strategy. The joint and hybrid distributions of the LBPs characterized by histogram are used for texture recognition. To enhance the robustness to rotation, Fourier transform is applied to LBP histogram in [33] and [16]. All of these methods have achieved satisfactory results on certain texture datasets, whereas their application to more challenging texture recognition tasks is very limited.

Fractal-based approach. One main attractive property of the fractal-based approach is its ability of capturing the self-similarities [1] of spatial distribution of textons, which is an essential characteristic of texture patterns. The early works [34, 35, 36] describe texture mainly using fractal dimension or the parameters of certain fractal models. The discriminative power of these methods is very limited. Recently, Xu et al. [19, 23] have made a progress in texture classification, by applying *multi-fractal spectrum* (MFS) analysis to texture description. The MFS descriptor has shown strong robustness to a wide range of geometrical changes, however, its robustness against photometric changes is not guaranteed.

Ji et al. [20, 21] combined MFS with multi-resolution analysis in wavelet domain to further exploit the local structures in texture. In their approach, multiple instances of the input image rotated by different angles are generated, and the tensor wavelet coefficients of these instances are averaged to achieve robustness to image rotation. In addition, a scale normalization scheme based on interest point detection is involved. Despite the impressive results achieved, the high computational cost of this approach remains a problem.

One closely related work to ours is the approach proposed in [18], which generalizes MFS as a global fractal dimension based descriptor combined with some local description. In their proposal, the local description is based on the response to several well-designed templates. The template-based feature is able to reduce the sensitivity of the original MFS to the photometric changes, but it is not scale-invariant. Thus, an extra scale alignment via multi-scale representation under a redundant system is involved. In their latter work [22], a scale selection scheme for each pixel is combined in order to further enhance the robustness. Although the classification performance of these approaches is impressive, the generated features are very redundant.

Finally, it should be mentioned that some approaches take fractal geometry as a local description instead of the global characterization. For instance, Varma et al. [17] proposed to compute local fractal-related features for each image patch. The local features are clustered as a texton dictionary for forming the texton histograms. The main drawback of this approach is the high computational complexity and the instability resulting from the clustering.

3. Preliminaries

Before presenting the detailed description of our approach, we first give an introduction of two mathematical tools upon which our approach is built.

3.1. Local Binary Patterns

There has been extensive literature on the LBP-based representation and applications. Interested readers can refer to [37] for more details. The original LBP operator proposed by Ojala et al. [10] forms labels for image pixels by thresholding the 3×3 neighborhood of each pixel with the center value and summing the resultant binary numbers weighted by powers of two. To adapt the LBP operator to the neighborhoods of different sizes [12], a circular symmetric neighborhood denoted by (P, R) is defined. Here P denotes the number of the sampling points and R denotes the radius of the neighborhood. The pixel value of a sampling point is bi-linearly interpolated if the point does not lie at the integer coordinates. See Figure 1 for an illustration of the circular symmetric neighborhood. The modified operator, denoted by $\mathrm{LBP}_{P,R}$, can be written as

$$\mathrm{LBP}_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) * 2^p, \tag{1}$$

where $s(x)$ is the sign function that is assigned the value 1 if x is positive and 0 if negative, g_c is the gray value of the center pixel and g_p $(p = 0, 1, ..., P-1)$ is the gray value of the neighbors.

The operator $\mathrm{LBP}_{P,R}$ is sensitive to image rotation. Hence, the rotation-invariant LBP operator [38], denoted by

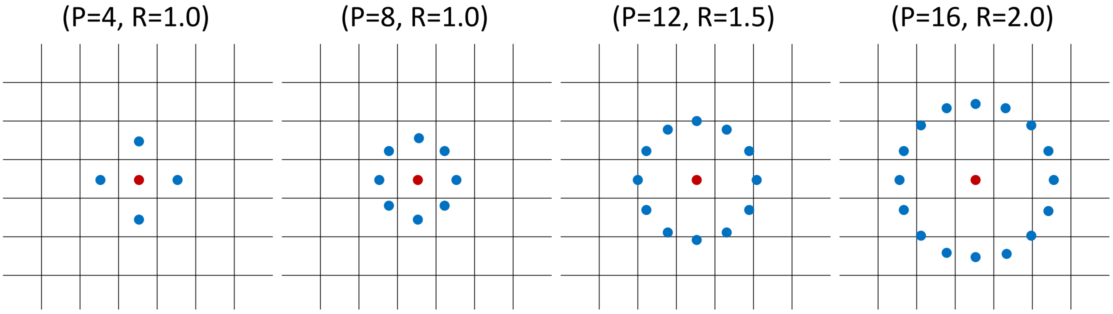


Figure 1: Circularly symmetric neighborhood used in LBP$_{P,R}$. The red point denotes the center pixel g_c and the blue points denote the sampling points g_p $(p = 0, 1, ..., P - 1)$.

LBP$^{ri}_{P,R}$, is developed by circularly rotating each LBP binary code into its minimum value:

$$\text{LBP}^{ri}_{P,R} = min\{ROR(\text{LBP}_{P,R}, p) \mid p = 0, 1, ..., P - 1\}, \tag{2}$$

where $ROR(x, p)$ performs a circular bit-wise right shift on x by p times. For instance, the bit sequences 10110000, 00101100 and 11000010 arise from different rotations of the same local pattern and they all correspond to the normalized sequence 00001011.

A further extension of LBP is to eliminate the patterns with frequent bitwise jumps in their binary codes, which can reduce the sensitivity to noise. The jump frequency is measured by a uniformity measure U defined as

$$U(\text{LBP}_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)|. \tag{3}$$

The measure U counts the number of bitwise transitions from 0 to 1 or vice versa when the bit pattern is considered circular. A local binary pattern is called uniform [12] if the uniformity measure on the pattern is at most two. The corresponding uniform rotation-invariant LBP operator LBP$^{riu2}_{P,R}$ is defined as

$$\text{LBP}^{riu2}_{P,R} = \begin{cases} \text{LBP}^{ri}_{P,R}, & if\ U(\text{LBP}_{P,R}) \leq 2; \\ P + 1, & otherwise. \end{cases} \tag{4}$$

The operator LBP$^{riu2}_{P,R}$ assigns a single label to all the non-uniform patterns, which benefits reducing the length of

LBP-based feature and implementing a simple rotation-invariant descriptor. By using a look-up table, the calculation of $LBP_{P,R}^{riu2}$ is very efficient. Note that there are many other LBP variants that have demonstrated better performance in texture description. In this paper the LBP operator $\text{LBP}_{P,R}^{riu2}$ is employed for its simplicity. Our results show that such a simple coding strategy is sufficient to perform well.

3.2. Fractal Analysis

The concept of fractal dimension which was first proposed by Mandelbrot [1] has been widely used in physics and geophysics to capture self similarities in many real world phenomena. Based on the concept of "measurement at scale δ", fractal dimension summarizes the irregularity of a given point set E in the space by measuring its power law behavior with respect to the scale δ:

$$m(\delta) \propto (\frac{1}{\delta})^q, \tag{5}$$

where $m(\delta)$ is some measurement of the given point set E at scale δ. For image, the measurement can be intensity, gradients, or filter response, *etc*. The exponential quantity q is the so-called *fractal dimension* and denoted by $\dim(E)$. One popular definition of the fractal dimension is the *box-counting* fractal dimension ([39]) defined as follows. Given a ℓ-dimensional point set Q, let the space $\mathbb{R}^\ell$ be covered by a mesh of ℓ-dimensional hypercubes with side length r (i.e., r-mesh ℓ-dimensional hypercubes) and a counting function $c(Q, r)$ is defined as the number of r-mesh ℓ-dimensional hypercubes that intersect Q. Then the *box-counting* fractal dimension $\dim(Q)$ is defined as

$$\dim(Q) = \lim_{r \to 0} \frac{\log c(Q, r)}{-\log r}. \tag{6}$$

In practice, for a point set coming from an image of size $m \times m$[1], to approximate the process of $r \to 0$, we estimate the slope of $\log c(Q, r)$ for a side-length sequence, $r_n > r_{n+1} > \cdots > r_m > 0$, $(n < m)$, using the least squares method. In our texture classification context, we define $r_i = \frac{m-i+1}{m}$, $i = n, n+1, \ldots, m$.

In order to distinguish the objects with mixture of multiple fractal structures (e.g., the statistically self-similar structures existing in texture), multi-fractal analysis extends the concept of fractal dimension to *multi-fractal spectrum*

[1] For simplicity, we assume that the image has equal length in each dimension, otherwise we can normalize it before calculating the fractal dimension.

(MFS). The basic idea of MFS is to compute fractal dimensions on the subsets of E and concatenate them as a vector as follows,

$$\text{MFS}(E) = [\dim(E(\alpha_1), \dim(E(\alpha_2), ..., \dim(E(\alpha_n)], \tag{7}$$

where $E(\alpha)$ denotes the collection of points having the same attribute value α. The point partition on E is originally based on the so-called Hölder exponent in [1] (or called density in [23]) that measures the local power-law behaviors of the points. One extension of MFS ([18, 22]) is to use a different strategy to compute the attribute value α in (7) instead of calculating Hölder exponent. By choosing a suitable strategy, the robustness of the MFS feature can be improved. It should be mentioned that there are many other types of fractal dimension defined for the point sets. We employed the box-counting fractal dimension in our implementation for its simplicity and computational efficiency.

4. Our proposed texture descriptor

Our proposed texture descriptor is built upon the multi-fractal analysis on the local image patterns. The local patterns are first located and encoded by the LBP operator $\text{LBP}^{riu2}_{P,R}$ defined in (4), and then the multi-fractal analysis using (7) is used to characterize the spatial distribution of the local image patterns. In other words, the proposed descriptor can be viewed as the collection of multiple fractal dimensions extracted from the local binary patterns at different scales. Thus we call it *pattern fractal spectrum* (*PFS*). The proposed approach is illustrated in Figure 2 In the rest of this section, we will detail each step in the algorithm.

4.1. Multi-scale Local Binary Patterns Extraction

Our descriptor relies on the characterization of local structures in texture. The intensity variation of local structures caused by illumination changes is very common in practice. In order to gain robustness to such global environmental changes, the LBP coding strategy is employed in our descriptor. Its robustness against local illumination changes has been demonstrated in many applications (see [12, 37] for more details). Viewpoint changes, image noises and clutters may also cause significant variations to the local structures. Thus, we use the uniform rotation-invariant LBP described in Sec. 3.1 to achieve rotational invariance and tolerance to noise. To further utilize the local structures at different scales, we compute LBP maps from different scales.

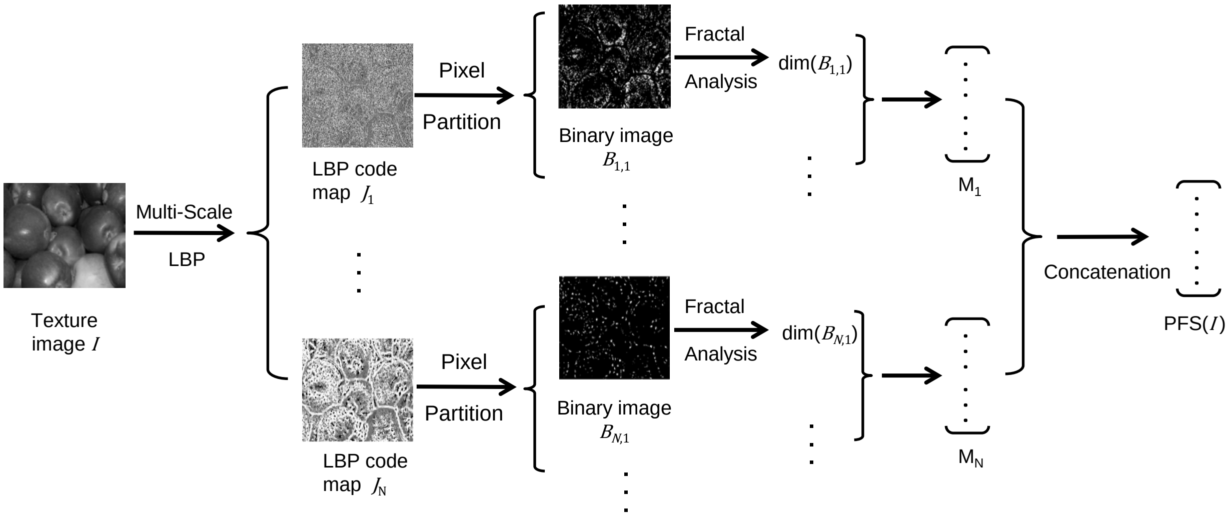
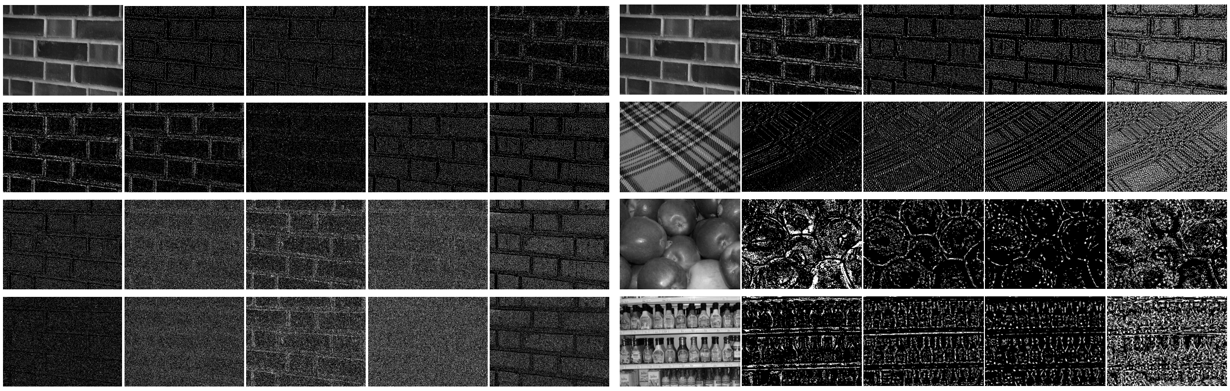


Figure 2: Flowchart of the proposed descriptor.

Given an image I, we obtain a sequence of LBP code maps $J_1, J_2, \ldots, J_N$ using the LBP operator $\text{LBP}_{P,R}^{riu2}$ on I with a series of parameters $\{(P_i, R_i)\}_{i=1}^{N}$. In details, J_i is a label image generated by computing $\text{LBP}_{P_i,R_i}^{riu2}$ on I. The parameter P_i defines coding neighborhood shape, and R_i determines the size of coding neighborhood and also the scale of encoded patterns. The code maps generated with different values of P can capture various local structures of interest in texture. By using multiple values of R, the multi-scale analysis is applied to I. Note that for digital image, larger R would result in more freedom in choosing P. Our experiment showed that, only a few values of R and P are able to characterize the rich structures of texture and achieve excellent performance. Due to the nature of the operator $\text{LBP}_{P,R}^{riu2}$, the set $\{J_i\}$ defined above is invariant to image rotation and to any monotonic transformation in grayscale.

Figure 3 (a) gives an illustration of the multi-scale LBP maps generated using the parameters $(8, 3)$, $(4, 2)$ and $(4, 1)$. It can be seen that each LBP code encodes different image structures and the LBP code maps across scales are able to capture different structures of the texture image, e.g., the $(8, 3)$ and $(4, 1)$ cases. Sometimes several LBP code maps at different scales might be similar, e.g., the $(4, 2)$ and $(4, 1)$ cases, but the slightly different information provided by such code maps might still be discriminative for recognizing different textures. Thus, by choosing appropriate multi-scale parameters, the resulted features can achieve the balance of distinction and compactness.

In fact, determining the scale and radius parameters (P, R) in multi-scale LBP coding scheme is not difficult, and we recommend the setting as follows. The scales (Rs) are defined as a series of integers which start from 1 and is increased by a factor of 1.5, which is very common for multi-scale representation. To obtain a compact representation, only four scales are used. The sampling number P in each scale is set to be 4, 8 ,or 16, according to magnitude of the scale. Because more freedom is allowed in larger scale, multiple Ps can be considered when the scale R is large. The corresponding configurations of the multi-scale LBP parameters (P, R) are $(4, 1)$, $(16, 2)$, $(16, 3)$, $(8, 5)$ and $(16, 5)$.



(a) Multi-scale LBP code maps

(b) Pixel classification

Figure 3: Binary images generated from the $\mathrm{LBP}^{riu2}_{P,R}$ coding. (a) The multi-scale LBP binary images are generated from a texture image (the upper left). Images in the first two rows (except the upper-left one) correspond to the LBP scale parameters $(8, 3)$. Images in the third row correspond to the LBP scale parameters $(4, 2)$, and images in the fourth row correspond to the LBP scale parameters $(4, 1)$. Here we omit the binary images of the nonuniform patterns. (b) The LBP binary images (the last four columns) are generated from four texture images (the first column) of different types. The binary images correspond to the LBP scale parameter $(16, 5)$. The code values of the columns of the binary images are 0,1,6,7 respectively.

4.2. *Multi-fractal analysis on image patterns*

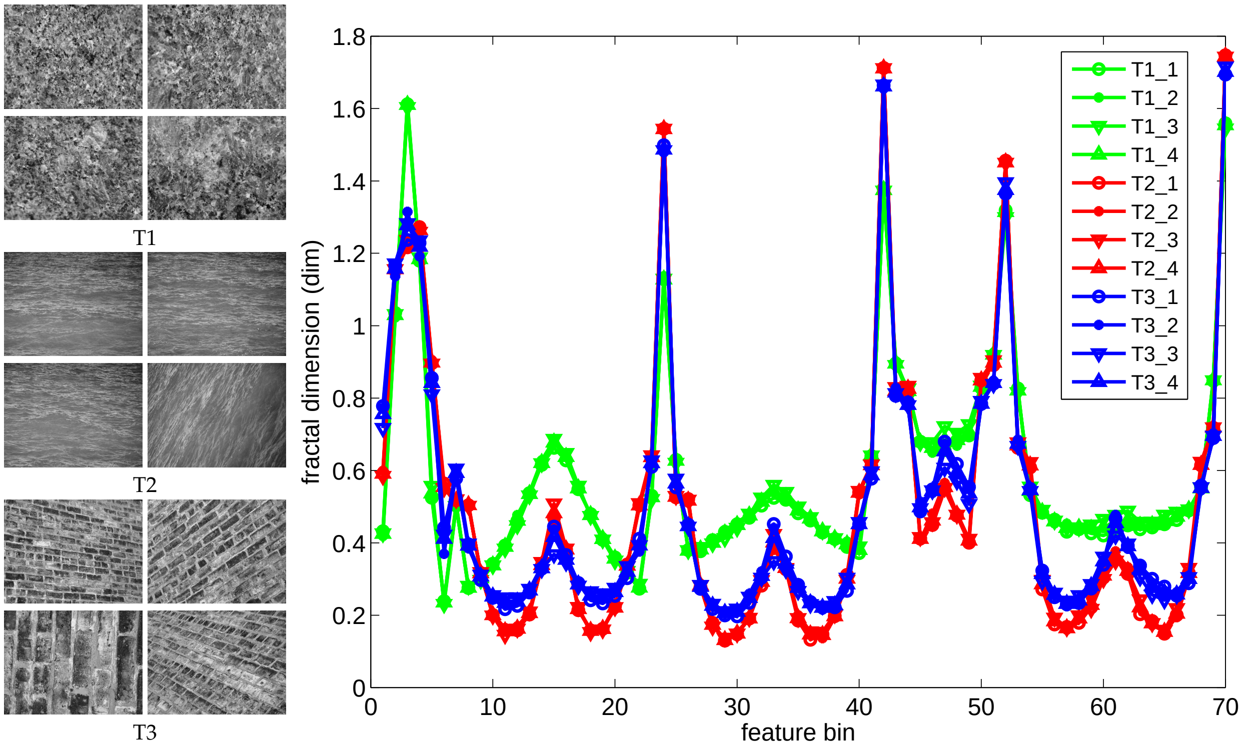
As described in the previous section, the calculated multi-scale LBP code maps $\{J_i\}_{i=1}^N$ can well encode the local structures of textures and provide many effective local descriptions. The next step is to integrate the local descriptions into a distinct global description in a statistical manner. In order to characterize the spatial distribution of the LBPs at each scale, multi-fractal analysis is applied by computing multiple fractal dimensions on each LBP code map. The main reason we apply multi-fractal analysis is that it has the capability of capturing the self-similarities of the texture patterns and the robustness against global geometric changes (see [19] for more details).

To apply multi-fractal analysis using (7), a point set partition scheme is required. For each LBP code map, we define the attribute value α in (7) as the LBP code of the corresponding pixel. Thus, for each possible LBP code (attribute value α), we can obtain a binary image on one LBP code map by setting the pixel to 1 if it is of that code, and 0 otherwise. In other words, multiple binary images $\{B_{i,j}, j = 1, ..., P_i + 2\}$ are generated from each code map

J_i as follows:

$$B_{i,j}(x,y) = \begin{cases} 1, & if\ J_i(x,y) = j; \\ \\ 0, & otherwise. \end{cases} \tag{8}$$

Each binary image represents the locations of the image patterns of the same type. See Figure 3 (b) for an illustration of the binary images obtained from the LBP code maps. From the figure we can find that, the binary images corresponding to the same LBP code value exhibit different spatial distributions and hence can well distinguish the local structures of different texture.



(a) Texture images

(b) The corresponding PFS feature

Figure 4: The 70-dimensional PFS descriptors computed on three types of texture. (a) Each type of texture contains four sample images. (b) The corresponding 70-dimensional PFS vectors.

To characterize the distribution of image patterns of the same type, the box-counting fractal dimension defined in (6) is computed on each binary images $B_{i,j}$. By concatenating all the fractal dimensions corresponding to the same

LBP code map, we obtain the MFS vector on each LBP code map J_i using (7):

$$\mathrm{MFS}(J_i) = [\dim(B_{i,1}), \dim(B_{i,2}), ..., \dim(B_{i,P_i+2})]. \tag{9}$$

Finally we define our PFS descriptor as

$$\mathrm{PFS} = \biguplus_i \mathrm{MFS}(J_i), i = 1, ..., N, \tag{10}$$

where $\biguplus$ denotes the operation of vector concatenation. The PFS descriptor can well capture the irregularity of the spatial distribution of the image patterns.

Note that the length of the MFS vector of J_i varies with the parameter (P_i, R_i). By the definition of the LBP operator $\mathrm{LBP}_{P,R}^{riu2}$, the length of the vector $\mathrm{MFS}(J_i)$ is $(P_i + 2)$. Thus, the total length of the proposed feature $\mathrm{PFS}(I)$ is $\sum_{i=1}^{N}(P_i + 2)$. According to parameters setting mentioned in Section 4.1, our method finally generates a 70-dimensional texture descriptor. See Figure 4 for an illustration of the 70-dimensional PFS descriptor. It can be observed that the proposed PFS descriptor has strong discriminative power as well as robustness against viewpoint changes.

5. Experimental evaluation

We evaluated the proposed texture descriptor by applying it to texture classification. The evaluation was conducted on four public texture datasets, including the UMD dataset [40], the UIUC dataset [41], the KTH-TIPS dataset [42], and the ALOT dataset [43]. The reason to choose these datasets is that all of them are challenging and have been used in the evaluation of many state-of-the-art texture classification methods. The details of these datasets are summarized in Table 1. The sample images of these four datasets can be referred to Figures 6, 7, 9 respectively.

We computed a 70-dimensional PFS feature for each image, using the multi-scale LBP parameters recommended in Section 4.1. The side-length sequence $\{r_i\}$ used in the calculation of the box-counting fractal dimension is $\{2, 3, 4, 5, 6\}$. The features of the images are directly sent to a classifier for training and testing. The classifier used in our classification experiment is the support vector machine (SVM) with the RBF kernel, which have been used in many

Table 1: Details of four texture datasets

Dataset	Num. of images	Num. of classes	Resolution	Scale change	Illumination condition
UMD	1000	25	1280×900	significant	uncontrolled
UIUC	1000	25	640×480	significant	uncontrolled
KTH-TIPS	810	10	200×200	medium	controlled
ALOT	25000	250	1536×1024	significant	controlled

texture classification approaches [7, 19, 21, 22, 26]. Following the work of [21, 26], we used the SVM implemented by Pontil et al. [44] with the 1-vs-all classification strategy adopted. The cost factor of the SVM was set as the number of images in the dataset. The kernel parameter was determined by the standard cross-validation within the range of $[0.25, 0.50]$. The classification accuracy is employed as the evaluation criterion, which is defined as the percentage of the samples that can be correctly classified in the dataset.

Due to the implementation complexity of many existing methods and required parameter tune-up for optimal performance, we compare our methods against those which either have the code available online, or have classification results reported on the tested datasets. To have a fair comparison, we only refer the results generated by the same classifier (SVM with RBF kernel) as we used. Note that the applications of the existing LBP-based methods are mostly on regular textures. As a result, we mainly focus on the comparison with the fractal-based approaches with state-of-the-art results.

5.1. Classification on the UMD dataset and the UIUC dataset

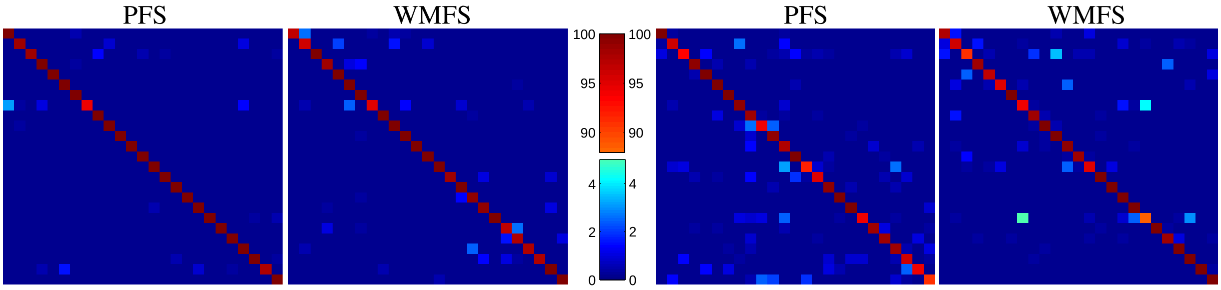
The UMD dataset and the UIUC dataset have been widely used in many previous studies [17, 19, 22, 23, 26]. These two datasets both have 1000 images from 25 different classes. We followed the experimental configuration of [21], which is also used in [19, 22]. The training set was selected as a random subset composed of 20 samples of the class, and all remaining images were used as the test set. The reported classification accuracy is the average over 200 random subsets. Five existing methods are selected for comparison:

- The *(H+L)(S+R)* method [30], characterizes texture by the histogram of self-adapting affine-invariant regions.
- The *VG-fractal* method [17], uses the local density function of various image measurements to produce a 13-dimensional local descriptor. The texture descriptor is the histogram of clusters of these local descriptors.

- The *MFS* method [23], applies multi-fractal analysis on image intensity, image gradient and image Laplacian respectively. The texture descriptor is obtained by combining the resulted MFS vectors.
- The *OTF* method [22], combines multi-fractal analysis with multi-scale local orientation histogram. A scale alignment strategy using wavelet tight frame decomposition is involved.
- The *WMFS* method, proposed in [20] and improved in [21], applies multi-fractal analysis on two complementary wavelet pyramids of image. The wavelet-based MFS is computed on the 16 instances of the original image rotated by different angles and averaged. A scale normalization based on interest point detection is involved.

Table 2: Classification accuracies (%) on the UMD dataset and the UIUC dataset.

Dataset	VG-fractal [17]	MFS [23]	(H+L)(S+R) [30]	OTF [22]	WMFS [21]	PFS
UMD	96.36	93.93	96.95	98.84	98.68	<u>99.38</u>
UIUC	92.31	92.74	97.02	98.14	<u>98.60</u>	97.92

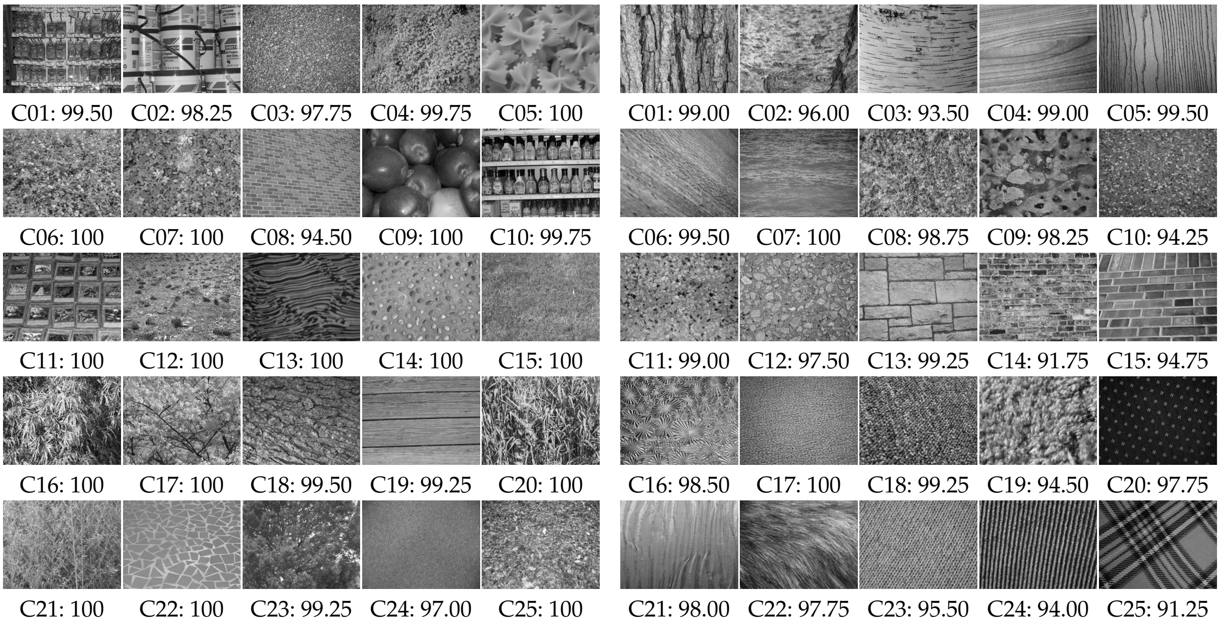


(a) UMD dataset

(b) UIUC dataset

Figure 5: Confusion matrices by the PFS descriptor on the UIUC and UMD datasets.

The experimental results of the tested methods on the two datasets are summarized in Table 2. On the UMD dataset, our method performed the best. On the UIUC dataset, our approach outperformed several state-of-the-art methods including the (H+L)(S+R), VG-fractal and MFS method, but did not perform as well as the OTF and WMF method. Nevertheless, the result of our method is still competitive because it is only slightly worse than the OTF method. The classification accuracies achieved by the PFS descriptor for individual class on both datasets are shown in Figure 6 (a) and (b) respectively. It can be seen that more than one half of the classes on the UMD dataset can be 100% correctly classified. On the UIUC dataset, the worst cases happened in the classes *C25* and *C14*. To summarize



(a) UMD

(b) UIUC

Figure 6: Classification accuracy (%) by the PFS descriptor on the UMD and UIUC datasets.

the misclassification rates between each two classes, the confusion matrices of our method and the WMFS method are shown in Figure 5.

5.2. *Classification on the KTH-TIPS dataset*

The KTH-TIPS dataset is a well-collected dataset devoted to the research on texture recognition. For each class of the dataset, totally 81 images were captured from the same material using a combination of nine scales, three poses and three illumination conditions. Some sample images from the dataset are shown in Figure 7. Two experiments were conducted on the KTH-TIPS dataset. The first one is the general texture classification similar to that on the UMD dataset in Section 5.1. The second one is the scale-specified and illumination-specified classification, which can evaluate the robustness against the scale and illumination changes.

5.2.1. *General texture classification*

On the KTH-TIPS dataset, we followed the experiment setup in [26]. The training set was selected as a random subset composed of 40 samples from each class, and all remaining images were used as the test set. The average accuracy is reported over 100 runs. To keep consistency with the result of [21], the variance of the classification

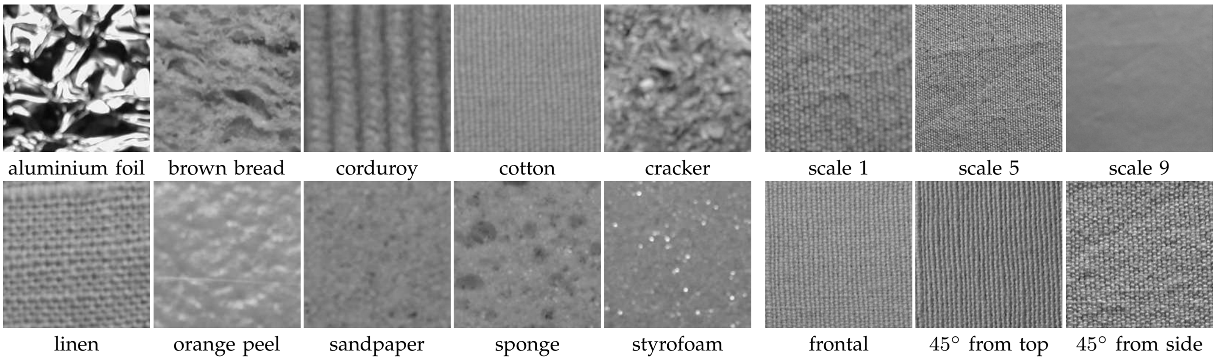


Figure 7: Sample images from the KTH-TIPS texture dataset. The first five columns show the samples from different classes. The remaining columns show the samples from the class *cotton* with different scale changes (the upper row) and illumination changes (the bottom row).

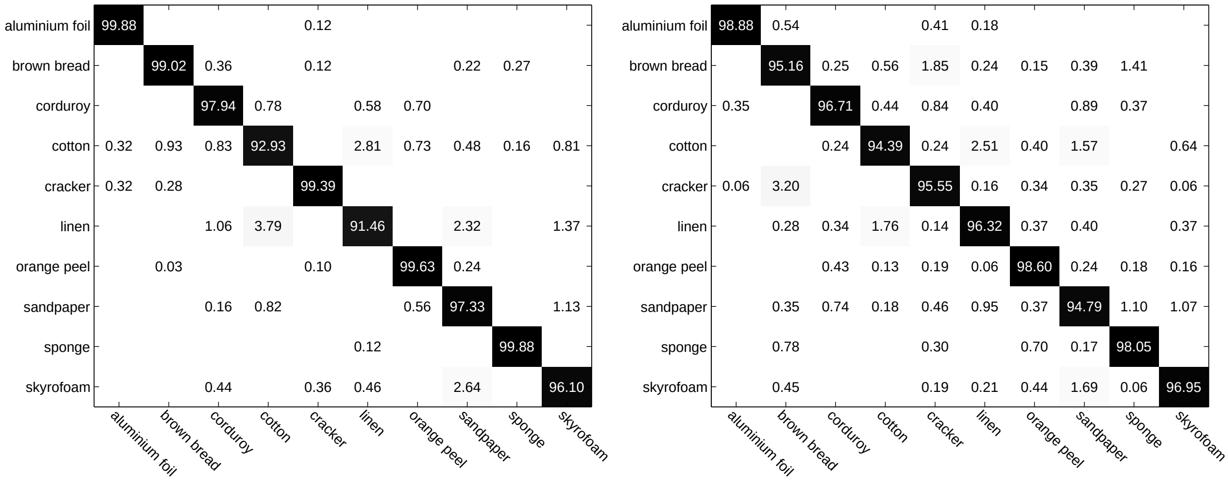
accuracies over runs is also reported. The methods for comparison include the aforementioned (H+L)(S+R) and WMFS methods. Besides, two approaches are involved:

- *Hayman et al.'s* method [7], describes texture image using filter bank responses.
- *Zhang et al.'s* method [26], is an extension of the (H+L)(S+R) method.

Table 3 summarizes the classification accuracies of the tested methods on the KTH-TIPS dataset. It can be seen that our method performed the best among all the tested methods. To summarize the misclassification rates between each two classes, the confusion matrices of our method and the WMFS method are shown in Figure 8. Our method performed the worst in dealing with classes *linen* and *cotton*, and these two classes were misclassified as each other with relatively high rates. The main reason is that both the classes contain the samples with similar patterns, which are difficult to be distinguished by the $\mathrm{LBP}_{P,R}^{riu2}$ coding. Note that the class *cotton* is also the worst one for the WMFS method due to the large variance existing in the class over scales (see the three images on the upper right of Figure 7). From Figure 8 it can be also seen that the confusion matrix of the WMFS method is denser than that of our method. One reason is that the WMFS descriptor emphasizes more on the invariance to changes and hence loses some degree of the distinction between classes. By contrast, our method can better strike the tradeoff between the invariance and the distinction.

Table 3: Classification accuracies (%) on the KTH-TIPS dataset.

Method	Hayman et al. [7]	(H+L)(S+R) [30]	Zhang et al. [26]	WMFS [21]	PFS
Accuracy (%)	94.80 ± 1.20	91.30 ± 1.40	96.10 ± 1.20	96.54 ± 1.07	97.35 ± 1.25



(a) PFS

(b) WMFS [21]

Figure 8: Confusion matrices on the KTH-TIPS dataset.

5.2.2. *Scale-specified and illumination-specified texture classification*

A well-designed experimental protocol was proposed in [24], with the purpose of investigating the behavior of the descriptor according to the scale and illumination change. The basic idea is to train the classifier using the samples from the several scales/illumintion conditions and then classify the samples from other scales/illumintion conditions. There are two parts of the experimental protocol:

- The scale-specified classification, in which images from one, three and five scales are used for training respectively. The details of the scale selection can be referred to Table 4. We compare our method with the *MR8* method evaluated in [24] as well as the aforementioned WMFS method.

- The illumination-specified classification, in which one illumination condition was fixed in training and two other conditions were used in testing. The three illumination conditions differ in the direction of the light. See Table 5 for the details. Since the result of the MR8 method on the illumination-specified classification is not available, we only compare our method with the aforementioned WMFS method.

Table 4: Classification accuracies (%) on the KTH-TIPS dataset. Experiments were conducted using for training (a) the central scale and 3 views per material (1 scale); (b) 3 equally spaced scales and 3 views per scale per material (3 scales); (c) 5 equally spaced scales and 3 views per scale per material (5 scales). The test set consisted of all the views of the remaining scales.

Material	(a) One scale			(b) Three scales			(c) Five scales		
	MR8 [24]	WMFS [21]	PFS	MR8 [24]	WMFS [21]	PFS	MR8 [24]	WMFS [21]	PFS
aluminium foil	88.89	93.06	93.06	96.29	100	100	100	100	100
brown bread	54.17	68.06	88.89	74.07	75.93	100	91.67	94.44	100
corduroy	61.11	88.89	75.00	74.07	83.33	100	86.11	100	100
cotton	34.72	27.78	6.94	55.56	51.67	42.41	77.78	87.50	86.11
cracker	79.17	56.94	75.00	92.59	90.93	97.59	94.44	88.89	100
linen	41.67	38.89	22.22	70.37	92.41	70.09	91.67	100	91.67
orange peel	41.67	77.78	45.83	57.40	82.78	98.15	80.55	100	97.22
sandpaper	75.00	69.44	68.06	74.07	77.78	91.57	83.33	86.39	97.22
sponge	77.78	88.89	63.89	92.59	96.30	100	100	94.44	100
styrofoam	79.17	75.00	72.22	96.29	90.74	92.59	100	97.22	97.22
average	63.33	68.47	61.11	78.33	84.19	89.24	90.56	94.89	96.94

Table 5: Classification accuracies (%) on the KTH-TIPS dataset. Experiments were conducted using for training when the illumination direction is (a) frontal ; (b) 45°from top; (c) 45°from side. The test set consisted of all the views of the remaining illumination conditions.

Material	(a) Frontal		(b) 45°from top		(c) 45°from side	
	WMFS [21]	PFS	WMFS [21]	PFS	WMFS [21]	PFS
aluminium foil	100	100	98.15	100	85.09	94.54
brown bread	33.33	100	82.87	96.30	16.02	62.96
corduroy	94.44	59.17	35.46	50.00	51.76	68.80
cotton	48.15	73.61	57.41	38.89	69.26	62.69
cracker	61.30	100	72.69	92.59	63.06	86.48
linen	98.15	81.48	81.94	100	85.28	75.93
orange peel	90.74	100	100	100	85.19	90.74
sandpaper	46.20	91.20	70.37	100	27.78	90.09
sponge	48.15	100	76.39	100	90.74	100
styrofoam	51.85	88.89	66.67	96.30	87.04	98.15
average	67.23	89.44	74.19	87.41	66.12	83.04

The experimental result of the scale-specified classification is listed in Table 4. It can be seen that our method performed the best in the cases of using three scales and five scales, but it did not perform better than the MR8 and WMFS methods when using one scale. Particularly, the performance of our method was very bad in the class *cotton* and poor in the class *linen*. We found that these two classes were likely to be misclassified as each other. This is similar to what happened in Section 5.2.2. Table 5 summarizes the experimental result of the illumination-specified classification. We can see that our method significantly outperformed the WMFS method. This does not surprise us because our method is very robust against the illumination changes due to the invariance of the LBP coding to any monotonic transformation in grayscale.

5.3. Classification on the ALOT dataset

The ALOT dataset is a large-scale dataset with high-resolution colorful texture images (see Figure 9). To speed up the computation, the images were converted to grayscale version and downsampled by half before feature extraction. We followed the experimental configuration of [21]. The training size for each class was set to be 5,10,15,...,50 and the corresponding classification accuracies are calculated respectively. The reported classification accuracies are the averages over 2000 runs. As there are only a few experimental results available on the ALOT dataset, we focus on the comparisons with three fractal-based methods with published results. These methods are the aforementioned MFS, OTF and WMFS methods.

Figure 9: Sample images from the ALOT dataset [43].

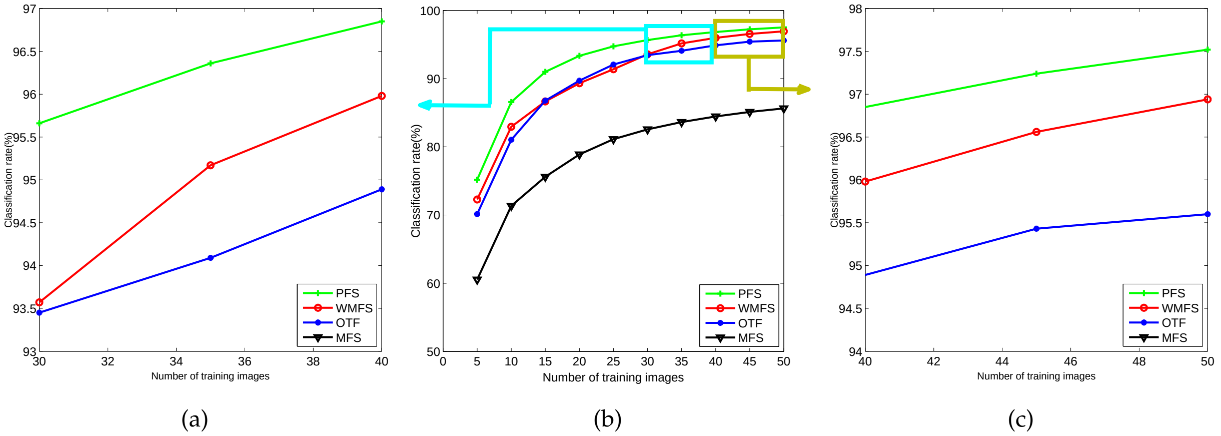


Figure 10: Classification accuracy (%) vs. number of training samples for the ALOT dataset.

The experimental results on the ALOT dataset are plotted as curves in Figure 10. It can be seen that the accuracy curves of the OTF method and the WMFS method are not stable with two intersections. One reason for such instability is that the OTF and WMFS methods are insufficiently robust to all kinds of significant environmental changes contained in images of the ALOT dataset. Thus, the performance of these two methods will be affected when the training set is not big enough to cover the samples with unfavorable environmental changes. Unlike the WMFS method, our method still performed stably well on the large-size dataset and achieved the best accuracies through all the training sizes. This demonstrates that our method has the discriminative ability as well as the robustness against a wide range of environmental changes.

5.4. *Efficiency*

Our descriptor has demonstrated its discriminative power by achieving the best performance among the compared methods on the UMD, KTH-TIPS and ALOT datasets and the competitive result on the UIUC dataset. We emphasize that our descriptor is not only highly distinct, but also compact and computationally efficient. To demonstrate this, we compare the feature length and the computational time of our approach with other two top fractal-based approaches, *i.e.,* the OTF method and the WMFS method. To have a fair comparison, all the tested methods is implemented in MATLAB. The software environment is the MATLAB 2012b platform run on the Windows 7 operating system, and the hardware platform is a desktop computer with Intel Core i7 quad-core 3.5GHz CPU and 8GB memory. We run the codes on the aforementioned four benchmark datasets and report the average running time over images on each dataset respectively.

Table 6: Comparison on the feature length (dim) and the running time (s).

Method	Feature length	Running time (s) per image on			
		KTH-TIPS	UIUC	UMD	ALOT
		(200 × 200)	(640 × 480)	(1280 × 900)	(1536 × 1024)
WMFS [22]	1218	1.53	7.46	29.60	36.01
OTF [21]	338	1.51	6.32	25.40	30.59
PFS	70	1.51	2.34	5.70	6.82

The result is listed in Table 6. It can be seen that our approach achieved both the shortest feature length and the lowest computational cost among all the compared methods. It can be also seen that as the image size increases, the running time of the OTF and WMFS methods grow rapidly. In contrast, with reasonably increasing computational cost, our method demonstrates its scalable capability of processing large-scale visual data (e.g., high-resolution images and high-definition videos). It should be mentioned that the OTF method requires more than 4GB memory when the image size is over 900×900, while our method only requires memory less than 2GB. Furthermore, benefitting from the compact and short feature, our classification process is also very efficient. The running time for each trial is about 0.12 seconds with 25 classes. Finally it is worth noting that our proposed approach does not require feature detection and clustering. Thus a large image size does not impose much computational burden on the computation of the PFS descriptor.

6. Summary and Conclusions

In this paper, a statistical texture descriptor is proposed, which applies the global fractal analysis to the local binary image patterns. The proposed descriptor has strong robustness to both local and global illumination changes and is robust to many geometric changes. Locally, robustness against image rotation, illumination changes and geometric variations is achieved by using the LBP coding strategy. The LBP also provides a compact representation for our descriptor. Globally, the multi-fractal analysis is employed to obtain further robustness against global environmental changes. The multi-fractal analysis is also very efficient in characterizing the strong multi-scale power law relationship in natural texture patterns, which makes our descriptor highly discriminative. Compared with other fractal-based approaches that have achieved impressive results, our approach is computationally efficient as it avoids any complex alignment stages, interest point detection, texton generation and cross-bin comparisons. Four challenging public texture datasets are used for evaluating the proposed descriptor. Our approach outperform other fractal-based approaches on three datasets and is competitive on the other. In addition, the feature length of our descriptor is the shortest among all the methods for comparison.

The LBP coding strategy used in our approach is very classical and simple. We believe that by using other LBP variants with improved robustness, our descriptor can achieve better performance. In future research, we would like to investigate how to utilize the joint spatial distribution of the LBPs in our framework and how to apply our descriptor to scene understanding.

References

[1] B. B. Mandelbrot, “The Fractal Geometry of Nature”, Macmillan, 1983.

[2] K. Dana and S. Nayar, “Histogram Model for 3D Textures”, in: Proceedings of the CVPR, pp. 618-624, 1998.

[3] T. Leung and J. Malik, “Representing and Recognizing the Visual Appearance of Materials Using Three-dimensional Textons”, International Journal of Computer Vision, 43(1), pp. 29-44, 2001.

[4] B. S. Manjunath, J. R. Ohm, V. V. Vasudevan, and A. Yamada, “Color and Texture Descriptors”, IEEE Transactions on Circuits and Systems for Video Technology, 11(6), pp. 703-715, 2001.

[5] M. Varma and A. Zisserman, “Classifying Images of Materials: Achieving Viewpoint and Illumination Independence”, in: Proceedings of the ECCV, pp. 255-271, 2002.

[6] S. Wu and M. S. Chantler, “Combining Gradient and Albedo for Rotation Invariant Classification of 2D Surface Texture”, in: Proceedings of the ICCV, pp. 848-855, 2003.

[7] E. Hayman, B. Caputo, M. Fritz, and S. O. Eklundh, “On the Significance of Real-world Conditions for Material Classification”, in: Proceedings of the ECCV, pp. 253-266, 2004.

[8] F. Mindru, T. Tuytelaars, L. Van Gool, and T. Moons, “Moment Invariants for Recognition under Changing Viewpoint and Illumination”, Computer Vision and Image Understanding, 94(1), pp. 3-27, 2004.

[9] S. Lazebnik, “Local, Semi-local and Global Models for Texture, Object and Scene Recognition”, University of Illinois at Urbana-Champaign, Champaign, IL, 2006.

[10] T. Ojala, M. Pietikäinen, and D. Harwood, “A Comparative Study of Texture Measures with Classification Based on Feature Distributions”, Pattern Recognition, 29(1), pp. 51-59, 1996.

[11] M. Pietikäinen, T. Ojala, and Z. Xu, “Rotation-invariant Texture Classification Using Feature Distributions”, Pattern Recognition, 33(1), pp. 43-52, 2000.

[12] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7), pp. 971-987, 2002.

[13] S. Liao, M. W. Law and A. C. Chung, “Dominant Local Binary Patterns for Texture Classification”, IEEE Transactions on Image Processing, 18(5), pp. 1107-1118, 2009.

[14] Z. Guo, L. Zhang and D. Zhang, “Rotation Invariant Texture Classification Using LBP Variance (LBPV) with Global Matching”, Pattern Recognition, 43(3), pp. 706-719, 2010.

[15] Z. Guo, L. Zhang, and D. Zhang, “A Completed Modeling of Local Binary Pattern Operator for Texture Classification”, IEEE Transactions on Image Processing, 19(6), pp. 1657-1663, 2010.

[16] G. Zhao, T. Ahonen, J. Matas and M. Pietikäinen, “Rotation Invariant Image and Video Description with Local Binary Pattern Features”, IEEE Transactions on Image Processing, 21(4), pp. 1465-1477, 2012.

[17] M. Varma and R. Garg, “Locally Invariant Fractal Features for Statistical Texture Classification”, in: Proceedings of the ICCV, pp. 1-8, 2007.

[18] Y. Xu, S. B. Huang, H. Ji, and C. Fermuller, “Combining Powerful Local and Global Statistics for Texture Description”, in: Proceedings of the CVPR, pp. 573-580, 2009.

[19] Y. Xu, H. Ji, and C. Fermuller, “Viewpoint Invariant Texture Description Using Fractal Analysis”, International Journal of Computer Vision, 83(1), pp. 85-100, 2009.

[20] Y. Xu, X. Yang, H. Ling, and H. Ji, “A New Texture Descriptor Using Multifractal Analysis in Multi-orientation Wavelet Pyramid”, in: Proceedings of the CVPR, pp. 161-168, 2010.

[21] H. Ji, X. Yang, H. Ling, and Y. Xu, “Wavelet Domain Multi-fractal Analysis for Static and Dynamic Texture Classification”, IEEE Transactions on Image Processing, 22(1-2), pp. 286-299, 2013.

[22] Y. Xu, S. B. Huang, H. Ji, and C. Fermuller, “Scale-space Texture Description on SIFT-like Textons”, Computer Vision and Image Understanding, 116(9), pp. 999-1013, 2012.

[23] Y. Xu, H. Ji, and C. Fermuller, “A Projective Invariant for Textures”, in: Proceedings of the CVPR, pp. 1932-1939, 2006.

[24] B. Caputo, E. Hayman, M. Fritz, and J. O. Eklundh, “Classifying Materials in the Real World”, Image and Vision Computing, 28(1), pp. 150-163, 2010.

[25] S. Lazebnik, C. Schmid, and J. Ponce, “A Discriminative Framework for Texture and Object Recognition Using Local Image Features”, Toward Category-Level Object Recognition, pp. 423-442, Springer, 2006.

[26] 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, 73(2), pp. 213-238, 2007.

[27] B. Julesz and R. Bergen, “Textons: the Elements of Texture Perception, and Their Interactions”, Nature, 290, pp. 91-97, 1981.

[28] S. Malik, S. Belongie, T. Leung, and S. Shi, “Contour and Texture Analysis for Image Segmentation”, International Journal of Computer Vision, 43(1), pp. 7-27, 2001.

[29] S. C. Zhu, C. Guo, Y. Wang, and Z. Xu, “What are Textons?”, International Journal of Computer Vision, 62(1), pp. 121-143, 2005.

[30] S. Lazebnik, C. Schmid, and S. Ponce, “A Sparse Texture Representation Using Affine-invariant Regions”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(8), pp. 1265-1278, 2005.

[31] Y. Rubner, C. Tomasi, and L. S. Guibas, “The Earth Mover’s Distance as a Metric for Image Retrieval”, International Journal of Computer Vision, 40(2), pp. 99-121, 2000.

[32] S. Liao and A. C. S. Chung, “Texture Classification by Using Advanced Local Binary Patterns and Spatial Distribution of Dominant Patterns”, in: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing, 1, pp. I1221-I1224, 2007.

[33] T. Ahonen, J. Matas, C. He and M. Pietikäinen, “Rotation Invariant Image Description with Local Binary Pattern Histogram Fourier Features”, Image Analysis, pp. 61-70, 2009.

[34] J. L. Vehel, P. Mignot, and J. P. Berroir, “Multifractals, Texture, and Image Analysis”, in: Proceedings of the CVPR, pp. 661-664, 1992.

[35] L. M. Kaplan, “Extended Fractal Analysis for Texture Classification and Segmentation”, IEEE Transactions on Image Processing, 8(11), pp. 1572-1585, 1999.

[36] S. Deguy, C. Debain, and A. Benassi, “Classification of Texture Images Using Multi-scale Statistical Estimators of Fractal Parameters”, in:

Proceedings of the BMVC, pp. 1-10, 2000.

[37] T. Mäenpää, "The Local Binary Pattern Approach to Texture Analysis: Extenxions and Applications", Oulun yliopisto, 2003.

[38] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns", in: Proceedings of the ECCV, pp. 404-420, 2000.

[39] K. S. Falconer, "Techniques in Fractal Geometry", Wiley Chichester, 1997.

[40] UMD texture dataset: http://www.cfar.umd.edu/~fer/High-resolution-data-base/hr_database.htm.

[41] UIUC texture dataset: http://www-cvr.ai.uiuc.edu/ponce_grp/data/index.html.

[42] KTH-TIPS texture dataset: http://www.nada.kth.se/cvap/databases/kth-tips.

[43] ALOT texture dataset: http://www.science.uva.nl/~aloi/public_alot.

[44] M. Pontil and A. Verri, "Support Vector Machines for 3D Object Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(6), pp. 637-646, 1998.