Spatiotemporal Lacunarity Spectrum for Dynamic Texture Classification

Yuhui Quan[a], Yuping Sun[a,b,*], Yong Xu[a]

[a]School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
[b]School of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China

Abstract

Dynamic texture (DT) in videos is the combination of texture patterns with motion patterns, and DT recognition is a key step in many vision-related applications. Owing to the additional challenges arising from the characterization on temporal organizations of texture elements, the recognition on DTs is more difficult than that on static textures. In this paper, a DT descriptor for classification is constructed, which examines the stationary irregularities of spatial and temporal distributions of local binary patterns in DT slices and encodes the irregularities by lacunarity-based features. The proposed descriptor has strong robustness to monotonic illumination changes and modest viewpoint changes, as well as strong discriminability in classification. In comparison with histogram-based methods, our approach is capable of encoding spatio-temporal details on the distribution of DT patterns. It also encodes additional details on the layout of DT patterns that recent fractal-based methods ignore. The proposed descriptor was applied to DT classification, and the experimental results show its power on several benchmark datasets.

Keywords: Dynamic Textures, Lacunarity Analysis, Local Binary Patterns, Video Classification

[*]Corresponding author.
Email addresses: csyhquan@scut.edu.cn (Yuhui Quan), ausyp@scut.edu.cn (Yuping Sun), yxu@mail.scut.edu.cn (Yong Xu)

1. Introduction

Advanced video processing technologies become more and more dependent on the capability of computers to extract stationary patterns from data. For instance, video description like MPEG-7 relies on descriptors that summarize patterns of contents [1], and video coding like H26X utilizes motion vectors for improving compression [2]. In videos, texture patterns and motion patterns are most often seen, and the combination of them leads to an interesting type of motion patterns known as Dynamic Texture (DT) [3–6]. Such patterns are very familiar in natural scenes and motions of textured objects, such as smoke, flames, clouds, springs, sea waves, swarm of birds, leaves in wind, humans in crowds, turning pages of book, etc. See Fig. 1 for some examples.

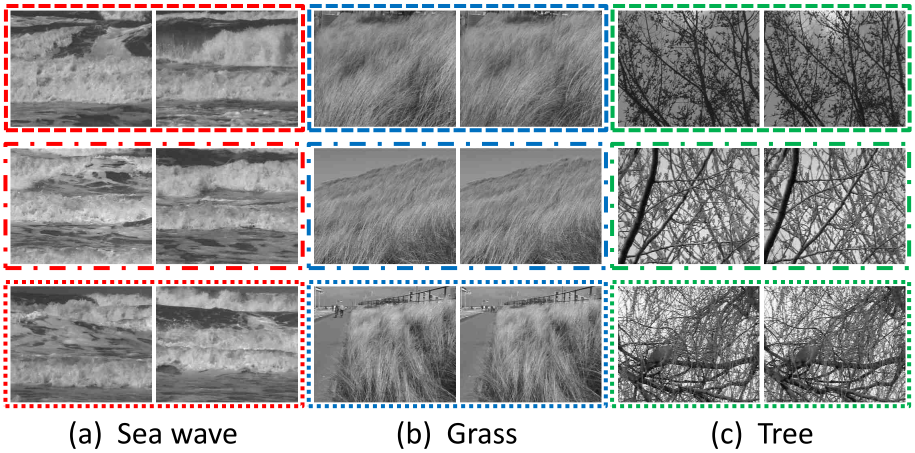


Figure 1: Examples of DT sequences. Each column shows three DT sequences from the same category, and for each DT sequence only a pair of key frames are shown.

The representation and recognition of DTs have seen many application, e.g. multimedia content representation and classification[7], content-based video indexing and retrieval[8], video quality assessment[9], object segmentation in video compression like MPEG-4[10], video coding[11], animation synthesis[12, 13], lip tracking and reading[14], facial expression analysis in HCI [15], and action detection in interaction with video games[16]. The research topics related to DTs range from description to segmentation and classification. The focus of this paper is on the design of DT descriptors for classification.

The categories of DTs are rich. DTs generated from the same material can belong to different types, e.g., water with a certain dynamic can become rivers, springs, waterfalls, or sea waves. The category of a DT is determined by both its spatial appearance and motion pattern, which are complementary to each other. Particularly, by exploiting the motion feature with the appearance feature, stability and accuracy can be improved over the single appearance feature of static images. Thus, a useful DT descriptor for classification should be able to characterize both the spatial appearance and temporal dynamics.

The design of an effective DT descriptor is challenging. DT sequences from the same class may have large variations in both the spatial arrangement and temporal organization of textual elements. Meanwhile, the descriptor should be robust to a wide range of environmental changes like changes of viewpoint and illumination. Furthermore, the computational cost of descriptor is more considerable in DTs, as videos are much bigger than images.

1.1. Motivation

The primary observation leading to our method is that local DT patterns are likely to distribute with stationary irregularity over space and time, and such irregularity varies a lot along different axes (i.e. the horizontal X axis, vertical Y axis, and temporal T axis) See the 2D slices in Fig. 2 for an illustration. The irregularity of the sea-wave sequence along X axis is significantly different to those along the Y axis and the T axis. Thus, we are inspired to develop a DT descriptor that characterizes the global irregularities of the spatial and temporal distributions of local space-time patterns in DTs.

Regarding local DT patterns, we extracted the local space-time patterns of DTs with multiple robust local binary patterns. Such patterns are invariant to image rotation and any monotonic gray-scale changes like most illumination changes, and they are also insensitive to noise. Regarding global irregularities, motivated by the effectiveness of lacunarity analysis in charactering irregularities of image surfaces and distributions of point sets [17], we developed a powerful tool called dynamic lacunarity analysis for the purpose.

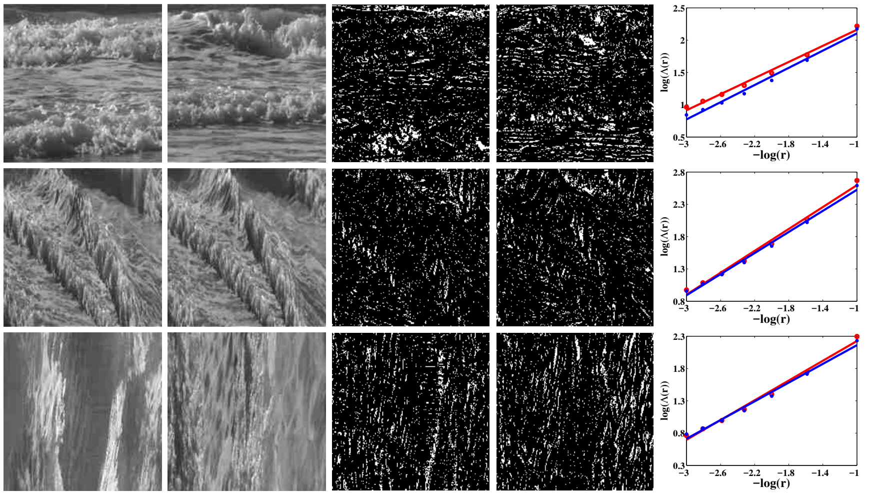


Figure 2: The stationary and distinct irregularities of the spatial and temporal distributions of local DT patterns on 2D slices along different axes. The rows from top to bottom correspond to the cases of XY, XT and YT planes respectively. For each case, we sampled two 2D slices from the DT sequence and show them in the first two columns. The next two columns show some binary images generated from the slices. The non-zero elements of the binary images denote the position of certain type of DT patterns which corresponds to some type of local DT structures. The last column shows the bi-log fitting plots for the scaling behaviors of the lacunarity of the binary images. The slope and intercept of each fitting line indeed encode the scaling behaviors of the lacunarity of local patterns and thus can be used as the description of the slices. The notations of the plots can be referred to Eqn. (7).

1.2. Contribution

The proposed descriptor has several advantages over the existing ones. Compared with the fractal-based methods [18–20], the spatiotemporal lacunarity spectrum developed in our method is more distinct than the multiple fractal dimensions used in [18–20], as it considers more details on how pixel sets are spatio-temporally distributed. Moreover, the local DT pattern encoding scheme used in our method brings two benefits. Firstly, our method enjoys the invariance to any monotonic gray-scale changes which cover various types of illumination changes. This is in contrast to [18–20] whose local descriptors (e.g. intensity and gradient in [18], wavelet coefficients in [20], and predefined templates in [20]) are sensitive to intensity. Secondly, unlike [20] where uniform partition is applied to wavelet coefficients, our method bypasses the challenge from feature bin partition.[1] Compared with the histogram-based methods [15, 22–25] which lose the spatial and temporal details on how local DT patterns are distributed, our method is able to encode the irregularities of the spatio-temporal distribution of local DT patterns. In practice, our descriptor is effective and efficient. With moderate length, ours shows noticeable improvement in experiments over many existing methods.

A preliminary conference version of this work appeared in [26]. In this paper, we replaced the original local pattern encoding scheme in [26] with a new one for improving the compactness and computational efficiency of the descriptor. The advantage of the proposed approach over its preliminary conference version [26] is that both the representation size and computational time are largely reduced while the classification accuracy decreases very slightly on average.[2] Moreover, more details and analysis have been added in the description of our method as well as the experimental evaluation. Another prior work of ours is [17] where the lacunarity concept is introduced to static texture classification. In this paper, we extend the work to the dynamic case and develop the dynamic lacunarity analysis. In addition, detailed analysis and discussion

[1] In [20], the uniform partition might produce big quantization error [21], as wavelet coefficients are usually sparse. In [18, 21], soft uniform bins are used to reduce the quantization error. But it remains an open question on how to get optimal uniform partition.

[2] The descriptor length is often an important factor in real applications. With the new local pattern scheme, the descriptor length is reduced by more than two thirds.

are given on lacunarity and its resultant features.

2. Related work

In the past, there has been an abundant literature on DT description and classification. Based on whether parameterized models are involved, most existing methods can be categorized into either parametric or non-parametric methods.

The parametric methods [27–32] model the dynamic patterns of DTs by some underlying physical dynamic systems and then perform classification based on the estimated parameters of the corresponding models. In the past, a wide range of parametric models have been used, such as the space-time autoregressive model [27] which expresses each DT voxel as the linear combination of its surrounding ones, the multi-scale dynamic autoregressive model [33, 34] which considers the space-time autoregressive model across scales, and the linear dynamical system which is equipped with parameters lying on Stiefel manifold [28] or is combined with bag-of-words system [35]. Though providing good understanding of DTs, the parametric methods need to explicitly model the generative systems of DTs, which makes it inflexible to describe the DTs generated by nonlinear physical systems with complex motion irregularities [18].

In contrast, non-parametric methods do not assume any form of the underlying physical systems of DTs, but directly extract statistical features from DT sequences. A classic type of such approaches is the field-based one where DT classification is done on the motion field [5, 36–39]. Using the estimated instantaneous motion patterns of DTs, Chetverikov et al. [5] proposed to convert the analysis of spatio-temporal sequences to that of sequences of static information. Peteri et al. [38] proposed to extract six translation invariant features based on normal flow and texture regularity to describe the dynamics and appearance of DT sequences. A metric of video sequences is defined in [39] using the velocity and acceleration fields estimated at various spatio-temporal scales. The main drawback of these methods is their heavy dependence on the estimation of motion field within video frames, which is sensitive to noise due to the ill-posedness of the optical flow estimation problem or is likely to fail in stochastic dynamics lacking of brightness constancy and local smoothness.

Regarding robustness of description, some non-parametric approaches (e.g. [15, 22, 25, 40, 41]) collect histogram-based statistics of certain local dynamic patterns. The performance of such approaches is highly dependent on the spatio-temporal appearance of DTs captured by local descriptors. The choices of local descriptors vary from spatio-temporal wavelet coefficients [20, 42, 43] to space-time oriented patterns [23–25, 44]. Recently, local binary patterns (LBP) [45] has emerged as a simple yet powerful local descriptor due to its robustness to monotonic gray-scale changes and moderate noises. Zhao et al. [15, 46] proposed two types of histogram-based features based on the volume local binary patterns and local binary patterns from three orthogonal planes. Instead of designing the hand-crafted features, Quan et al. [47] proposed to learn local DT features using an efficient sparse coding model. To further capture the nonlinearities of DT during feature learning, Favorskaya et al. [48] proposed a convolutional network for the DT recognition. The results of such learning-based approaches are impressive but the invariance of the learned features cannot be guaranteed.

There are several recent approaches combining the ideas of non-parametric methods and parametric methods. To distinguish and utilize the contributions of spatial patterns and motion patterns in classifying different types of DTs, Ghanem et al. [49] aggregated two discriminative spatial descriptors with one generative temporal descriptor by adaptive weighting via maximum margin distance learning. In the similar spirit, Yang et al. [50] proposed to aggregate various kinds of discriminative spatial features and generative temporal features via ensemble classifiers for DT classification. The fractal-based methods [18–20] can be viewed as discriminative methods with generative motivations, whose basic idea is to treat a DT sequence as being generated by some physical dynamic process with spatio-temporal self-similarities in its dynamics and then apply multi-fractal spectra to characterize such self-similarities without explicitly defining the dynamic process. Among these three methods, the seminal work is [18], and the differences of them lie in their local features and the dimension of space used for computing fractal spectra (i.e. 2D plane or 3D volume).

The fractal-based approaches [18–20] are the very closely-related work to ours. Compared with these methods, this paper aims at developing a new kind of global integration method via the lacunarity analysis instead of fractal spectrum analysis, which plays an important role in DT classification system and yields superior performance over the description based on fractal spectrum. 135

3. Preliminaries

3.1. Local binary patterns

Local binary patterns (LBP) is one type of local features that has been widely-used in computer vision. The original LBP operator [51] 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. This operator can be adapted to the neighborhoods of different sizes. Let (P, R) denote a circular symmetric neighborhood denoted with P sampling points and radius R.[3] Then the modified operator is defined as follows [45]:

$$\mathrm{LBP}_{P,R}(c) = \sum_{p=0}^{P-1} s_0(g_p - g_c) * 2^p, \tag{1}$$

where g_c is the gray value of the center pixel c, g_p $(p = 0, 1, ..., P-1)$ is the gray value of the neighbor of c indexed by p, and $s_a(y)$ is the thresholding function defined as follows:

$$s_a(y) = \begin{cases} 1, & \text{if } y \geq a; \\ 0, & \text{if } y < a. \end{cases} \tag{2}$$

See Fig. 3 for some examples of LBPs which correspond to different image structures.

The operator $\mathrm{LBP}_{P,R}$ is not invariant to image rotation, as the indices of neighboring pixels are fixed. To overcome such a weakness, the rotation-invariant LBP operator [52], denoted by $\mathrm{LBP}^{ri}_{P,R}$, aligns the neighboring pixels by circularly rotating each LBP code into its minimum value, which is done via

$$\mathrm{LBP}^{ri}_{P,R}(c) = f_P(\mathrm{LBP}_{P,R}(c)) \tag{3}$$

[3] The pixel value of a sampling point is bi-linearly interpolated if the point does not lie at the integer coordinates.

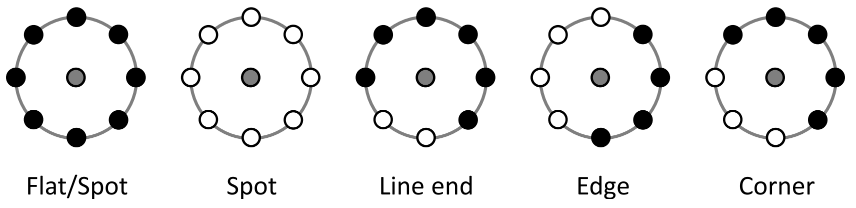


Figure 3: Image structures corresponding to different LBP patterns

with

$$f_P(x) = min\{g(x, p) \mid p = 0, 1, ..., P - 1\}, \tag{4}$$

where $g(x, p)$ performs a bit-wise circular right shift on x by p times. For example, the bit sequences 10110011, 11001110 and 10011101 are three different rotated instances of the same pattern and they become the same sequence (i.e. 00111011) after rotational shifting.

For classification, unstable patterns are useless and even harmful. In practice, the unstable patterns often contain frequent bitwise jumps in their binary codes. To eliminate such patterns, the uniform LBP scheme considers a measure U on the jump frequency of binary code, which counts the number of bitwise transitions from 0 to 1 or vice versa when the bit pattern is considered circular. Then the new LBP operator $\mathrm{LBP}^{riu2}_{P,R}$ [45] is defined by

$$\mathrm{LBP}^{riu2}_{P,R}(c) = \begin{cases} \mathrm{LBP}^{ri}_{P,R}(c), & \text{if } U(\mathrm{LBP}_{P,R}(c)) \leq 2; \\ P + 1, & \text{otherwise.} \end{cases} \tag{5}$$

In other words, the operator $\mathrm{LBP}^{riu2}_{P,R}$ packs all the non-uniform patterns whose measure values are more than two. Thus, it is called uniform LBP operator with rotational invariance. The discard of non-uniform patterns also benefits the feature length reduction as well as the robustness to noises.

The robustness of LBP to noises can also be improved by setting the threshold a of $s_a(y)$ in defining LBP to be a small positive value τ instead of zero. This scheme can reduce the sensitivity of the uniform or near-uniform regions to perturbation. In this scheme, the sign of $(g_c - g_p)$ should be treated separately. Then the positive and negative LBPs [53] are defined as follows:[4]

$$\begin{cases} \mathrm{LBP}^{+\tau}_{P,R}(c) = \sum_{p=0}^{P-1} s_{\tau}(g_p - g_c) * 2^p; \\ \mathrm{LBP}^{-\tau}_{P,R}(c) = \sum_{p=0}^{P-1} s_{-\tau}(g_p - g_c) * 2^p, \end{cases} \tag{6}$$

It is noted that all the aforementioned LBPs are invariant to monotonic gray-scale changes like many illumination changes, as such changes do not change the sign of $(g_c - g_p)$. Also note that the calculation of these LBPs can be accelerated via look-up tables.

3.2. Lacunarity analysis

While histogram has been a widely-used tool for constructing global features, it does not encode the spatial or temporal distribution of local features. As an alternative, recent fractal-based methods (e.g. [18–21]) use multiple dimensions to encode irregularities of distribution of local features, which have demonstrated their effectiveness in characterizing both static and dynamic textures. The basic idea of fractal-based methods is that surfaces of textures exhibit strong self-similarities which can be well described in fractal geometry.

Besides fractal dimensions, there are other fractal tools that can be used for describing irregularities of distribution. One of them is the so-called lacunarity which early was used in [54] for natural scene description with limited performance and recently has been exploited for classifying texture images in [17]. Compared with fractal dimension, lacunarity considers more details on the irregularity of distribution. Given a point set $\mathcal{B}$, let $n(\mathcal{B}, r, m)$ be the number of r-mesh squares[5] that intersect m points in $\mathcal{B}$.[6] Then the lacunarity of $\mathcal{B}$ at scale r, denoted as $\Lambda_r(\mathcal{B})$, is defined as follows [55]:

$$\Lambda_r(\mathcal{B}) = \frac{E_m[n^2(\mathcal{B}, r, m)]}{(E_m[n(\mathcal{B}, r, m)])^2}, \tag{7}$$

[4] In many scenarios, this scheme is called local ternary patterns.

[5] The r-mesh squares refer to the non-overlapping adjoint squares with side length equal to r.

[6] In the case of fractal dimension, m is only considered as a binary variable, i.e. $m = 0$ and $m > 0$. Thus, only the squares with $m > 0$ will contribute to the calculation of fractal dimension.

where $E_y[\boldsymbol{x}]$ is the expectation value of $\boldsymbol{x}$ over variable $\boldsymbol{y}$. By using $D_{\boldsymbol{y}}(\boldsymbol{x}) = E_{\boldsymbol{y}}(\boldsymbol{x}^2) - [E_{\boldsymbol{y}}(\boldsymbol{x})]^2$ where $D_{\boldsymbol{y}}(\boldsymbol{x})$ is the variance of $\boldsymbol{x}$ over variable $\boldsymbol{y}$, we can rewrite (7) as

$$\Lambda_r(\mathcal{B}) = \frac{D_m[n(\mathcal{B}, r, m)]}{(E_m[n(\mathcal{B}, r, m)])^2} + 1. \tag{8}$$

In other words, $\Lambda_r(\mathcal{B})$ is a dimensionless representation of the variance to mean ratio which measures statistical dispersion. Smaller $E_m[n(\mathcal{B}, r, m)]$ implies the squares in the mesh contain fewer points on average and hence bigger lacunarity, while bigger $D_m[n(\mathcal{B}, r, m)]$ implies larger diversity of point distribution and hence lacunarity, and vice versa. Under certain scale, the point sets with different irregularities of distribution would have distinct lacunarity. See Fig. 4 for some examples.

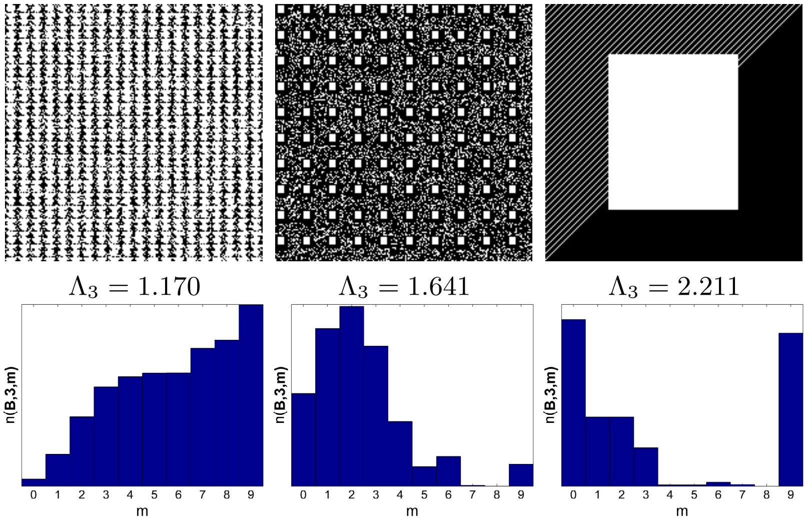


Figure 4: Lacunarity values of three different binary images. The first row shows three binary images with different irregularities of distribution. The second row shows the corresponding lacunarity values of these images under scale $r = 3$. The third row shows the corresponding histograms (i.e. $n(\mathcal{B}, r, m)$ w.r.t. m) in the calculation of lacunarity. It can be seen that under certain scale the binary images with a wider range of gap sizes incline to be more lacunar.

In [17], the behavior of $\Lambda_r(\mathcal{B})$ over scale r is assumed to exhibit power law for texture images, that is

$$\Lambda_r(\mathcal{B}) \propto (\frac{1}{r})^{P(\mathcal{B})}, \tag{9}$$

where $P(\mathcal{B})$ is a scale-independent exponent encoding the irregularity of $\mathcal{B}$. By taking logarithm on both sides of Eqn. (9), we obtain

$$\ln \Lambda_r(\mathcal{B}) = P(\mathcal{B}) \ln r + L(\mathcal{B}), \tag{10}$$

where $L(\mathcal{B})$ is a scale-independent scalar which encodes the structure of $\mathcal{B}$. In implementation, both $P(\mathcal{B})$ and $L(\mathcal{B})$ can be estimated with bi-logarithmic least square fitting over a finite sequence of box sizes that are often set to be consecutive integers. Intuitively, the value of $\exp(L(\mathcal{B}))$ can be interpreted as a scale-independent measure on the size of lacunarity for objects with identical $P(\mathcal{B})$. In other words, for the objects which cannot be well distinguished by $P(\mathcal{B})$, we can use $L(\mathcal{B})$ as a discriminative complement to $P(\mathcal{B})$. Examples are objects with similar irregularities, such as homogeneous regions, edges and corners. See Fig. 5 for an illustration.

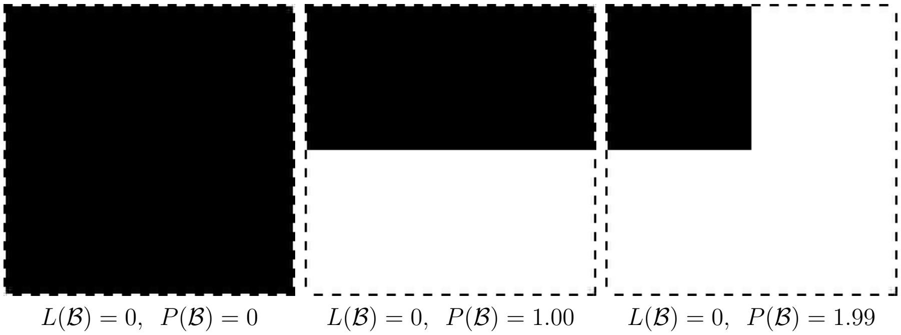

$L(\mathcal{B}) = 0,\ \ P(\mathcal{B}) = 0$
$L(\mathcal{B}) = 0,\ \ P(\mathcal{B}) = 1.00$
$L(\mathcal{B}) = 0,\ \ P(\mathcal{B}) = 1.99$

Figure 5: Three texture primitives with identical $P(\mathcal{B})$ but distinct $L(\mathcal{B})$.

4. Our Method

It can be observed from Fig. 2 that the distributions of local DT patterns in the spatial and temporal domains exhibit significant differences on the 2D slices sampled from different planes (i.e., one spatial plane denoted by XY and two space-time planes denoted by YT and XT respectively) while being similar on the slices along the same axis. Thus, we characterize the distribution of local DT patterns on each spatial and temporal plane. Then the resultant feature vectors are integrated along the X, Y and T axis respectively.

More concretely, our method examines the spatio-temporal distribution of local DT patterns from three orthogonal planes respectively. To fully exploit the stable DT patterns on the 2D slices sampled from different planes, we extract local binary patterns with three efficient encoding schemes, which resist to illumination changes and provide a natural way for partitioning feature bins. Then we calculate the lacunarity spectra regarding the distribution of each pattern along each axis. This process is called *dynamic lacunarity analysis*. As the distributions of local DT patterns on slices are similar along the same axis, for robustness, the resultant slice-wise feature vectors are averaged along each axis respectively and then concatenated as the final descriptor. We call our descriptor as *spatial-temporal lacunarity spectrum (STLS)*. Our method is outlined in Fig. 6, which consists of four steps: DT sequence slicing, space-time pattern encoding, lacunarity analysis, and feature integration. Each step will be detailed in the following subsections.

The notations used throughout the paper are as follows: bold letters are used for matrices and vectors, regular letters for scalars (such as vector components and dimensions), and calligraphic English alphabets for operators and sets.

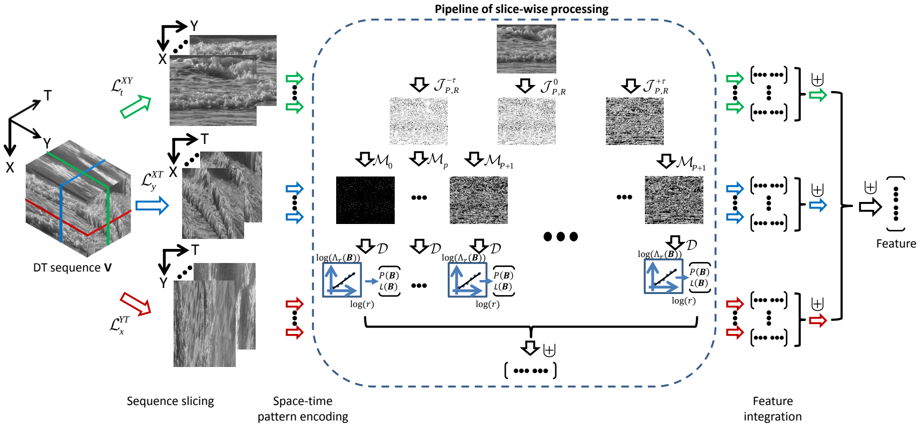


Figure 6: Outline of dynamic lacunarity analysis. Given a DT sequence, the 2D slices are sampled along the X, Y, and T axes. Then for each 2D slice, three types of local pattern encoding schemes are applied, resulting in multiple binary images generated by pixel-wise classification. Finally, lacunarity analysis is applied to each binary image via bi-logarithmic least square fitting on the calculated lacunarities over a finite sequence of box sizes. The resulting lacunarity features on the 2D slices are collected as the final DT feature.

4.1. Sequence slicing

A sequence $\boldsymbol{V} \in \mathcal{R}^{M \times N \times K}$ of DT is considered as a 3D cube with the X, Y and T axes. Considering the co-occurrence statistics along the T axis, $\boldsymbol{V}$ can be viewed as a stack of XY slices. Similarly, $\boldsymbol{V}$ can be viewed as a stack of XT slices or YT slices, depending on the selected axis. As illustrated in Fig. 2, slices sampled along different axes exhibit stationary but distinct behaviors. This inspires us to extract the appearance or motion-appearance features of $\boldsymbol{V}$ from all 2D slices sampled along the three axes. With this purpose, we first apply three slicing operators $\mathcal{L}^T$, $\mathcal{L}^Y$ and $\mathcal{L}^X$ to decompose $\boldsymbol{V}$ into slices along the T, Y and X axes respectively, which are defined as follows:

$$\begin{cases} \mathcal{L}_t^T \circ \boldsymbol{V} = \boldsymbol{V}(:,:,t),\ t = 1, \ldots, K; \\ \mathcal{L}_y^Y \circ \boldsymbol{V} = \boldsymbol{V}(:,y,:),\ y = 1, \ldots, N; \\ \mathcal{L}_x^X \circ \boldsymbol{V} = \boldsymbol{V}(x,:,:),\ t = 1, \ldots, M. \end{cases} \tag{11}$$

As a result, we can obtain three types of slices, i.e., $\{\mathcal{L}_t^T \circ \boldsymbol{V}\}_{t=1}^K$, $\{\mathcal{L}_y^Y \circ \boldsymbol{V}\}_{y=1}^N$, and $\{\mathcal{L}_x^X \circ \boldsymbol{V}\}_{x=1}^M$.

In the latter stages, slices along different axes are separately processed. By this slicing scheme we can capture rich discriminative features in both spatial and spatiotemporal planes. Such a scheme has also been used in other existing methods (e.g. [19, 41]). However, independently processing slices along different axes indeed much weakens the rotation invariance of the descriptor. For example, rotating the camera by 90 degrees about the optical axis will swap the slices along horizontal and vertical axes. When the rotation is not along the optical axis, plenty of existing methods (e.g. [20, 34, 49]) also have limited robustness to the rotation, as these methods involve slicing operations in either local or global manner and the slices are not well aligned under the rotation. In other words, all the aforementioned methods implicitly assume there only exists modest camera rotation in data or registration of data has been done. Fortunately, such an assumption is often true in practical applications, e.g., when taking videos from natural scenes like rivers and seas we seldom generate a rotated version, and when analyzing facial expression we usually fix the camera. The assumption is also supported by the experimental results from existing literature, where many of the aforementioned methods still perform well in real datasets.

4.2. Slice-wise pattern encoding

The second step of our method is to locate different kinds of local DT patterns on each DT slice. To strike the balance of discriminability and robustness, we combine the ideas from the uniform rotation-invariant local binary patterns $\mathrm{LBP}_{P,R}^{riu2}$ with the positive and negative local binary patterns $\mathrm{LBP}_{P,R}^{+(-)\tau}$ [53] given in Eqn. (5) and (6) respectively. Define $\mathcal{J}_{P,R}^{+\tau}$ and $\mathcal{J}_{P,R}^{-\tau}$ to be

$$\mathcal{J}_{P,R}^{+\tau}(c) = \begin{cases} f_P(\mathrm{LBP}_{P,R}^{+\tau}(c)), & \text{if } U(\mathrm{LBP}_{P,R}^{+\tau}(c)) \leq 2; \\ P+1, & \text{otherwise,} \end{cases} \tag{12}$$

and

$$\mathcal{J}_{P,R}^{-\tau}(c) = \begin{cases} f_P(\mathrm{LBP}_{P,R}^{-\tau}(c)), & \text{if } U(\mathrm{LBP}_{P,R}^{-\tau}(c)) \leq 2; \\ P+1, & \text{otherwise.} \end{cases} \tag{13}$$

These two operators calculate the uniform rotation-invariant codes on the positive and negative LBP maps. Note that the sign-less LBP defined in Eqn. (5) characterizes local structures of DTs in a different way from the positive and negative LBPs. See an illustration of such difference in Fig. 7. In other words, sign-less LBP provides additional information over the positive and negative ones. For further discrimination of features, we also consider

$$\mathcal{J}_{P,R}^{0}(c) = \mathrm{LBP}_{P,R}^{riu2}(c) \tag{14}$$

in our local pattern extraction process.

For a DT slice $\boldsymbol{I}$, we generate the code maps $\boldsymbol{C}^{+}$, $\boldsymbol{C}^{-}$ and $\boldsymbol{C}^{0}$ by applying $\mathcal{J}_{P,R}^{+\tau}$, $\mathcal{J}_{P,R}^{-\tau}$ and $\mathcal{J}_{P,R}^{0}$ to $\boldsymbol{I}$ respectively:

$$\begin{cases} \boldsymbol{C}^{+} = \mathcal{J}_{P,R}^{+\tau} \circ \boldsymbol{I}; \\ \boldsymbol{C}^{-} = \mathcal{J}_{P,R}^{-\tau} \circ \boldsymbol{I}. \\ \boldsymbol{C}^{0} = \mathcal{J}_{P,R}^{0} \circ \boldsymbol{I}. \end{cases} \tag{15}$$

Benefiting from the properties of $\mathrm{LBP}_{P,R}^{riu2}$ and $\mathrm{LBP}_{P,R}^{+(-)\tau}$, the pattern code maps $\boldsymbol{C}^{+}$, $\boldsymbol{C}^{-}$

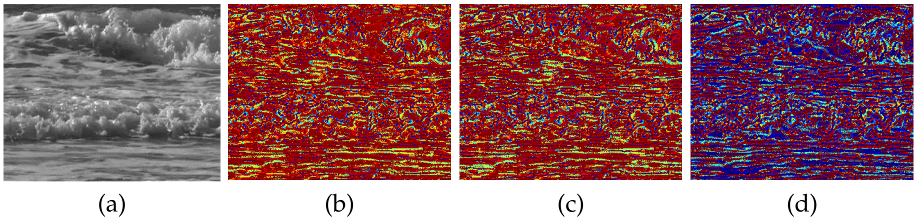


Figure 7: Coding results by different schemes. (a) Original images; (b) Code map by J^{-}; (c) Code map by J^{0}; (d) Code map by J^{+}. Different colors denote different code values.

and C^0 enjoy both the robustness to illumination changes and resistance to random noises.

Given a pattern code map $\boldsymbol{C}$ ($\boldsymbol{C}^{+}$, $\boldsymbol{C}^{-}$ or C^{0}) generated from a 2D slice $\boldsymbol{I}$, we partition all its voxels into groups based on the code values and then generate multiple binary images, each of which corresponds to the spatial-temporal distribution of one type of local DT pattern on $\boldsymbol{I}$. In details, we define $\mathcal{M}_p$ as the operator to extract a binary image from a given pattern code map $\boldsymbol{C}$ with respect to the code value p:

$$(M_p \circ \boldsymbol{C})(x) = \begin{cases} 1, & \text{if } \boldsymbol{C}(x) = p; \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

Each binary image encodes the locations of local DT patterns of certain type in the slice $\boldsymbol{I}$.

4.3. Lacunarity analysis on binarized feature slices

With the extracted binary images from each 2D slice, the next step is to characterize the distribution of each type of local patterns. We adopt the lacunarity analysis to this task, as the stochastic self-similarities which exist in a wide range of DTs (e.g. the amplitude of temporal frequency spectra of many video sequences like camera movements, weather and biological movements by one or more humans, indeed fits power-law models [18, 21, 56–58]) can be well characterized with lacunarity-based features.

Given a binary image $\boldsymbol{B}$ from slice $\boldsymbol{I}$, we view $\boldsymbol{B}$ as a set of pixels existing at the positions of the non-zeros, and then define an operator $\mathcal{D}$ which calculates the lacunarity-based features on $\boldsymbol{B}$ as follows:

$$\mathcal{D} \circ \boldsymbol{B} = [P(\boldsymbol{B}), L(\boldsymbol{B})]. \tag{17}$$

Then the lacunarity-based features calculated from all the binary images that correspond to $\boldsymbol{I}$ are gathered as the description for the slice. For convenience, we define the operator $\mathcal{S}^{\tau}_{P,R}$ for this process as follows:

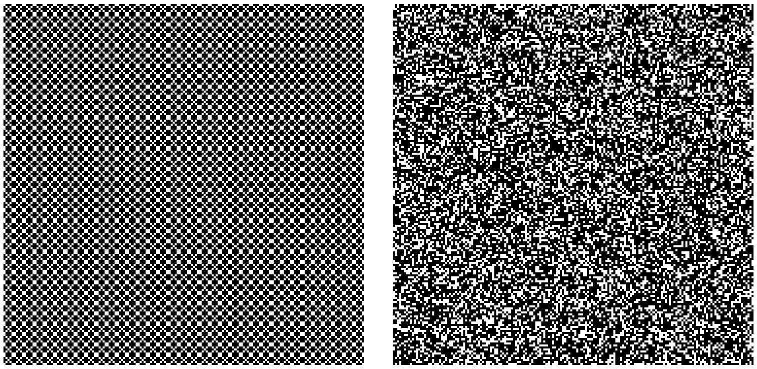
$$\begin{aligned} \mathcal{S}^{\tau}_{P,R} \circ \boldsymbol{I} = \Bigg[& \biguplus_{p=0}^{P+1} \mathcal{D} \circ \mathcal{M}_p \circ \mathcal{J}^{+\tau}_{P,R} \circ \boldsymbol{I}, \\ & \biguplus_{p=0}^{P+1} \mathcal{D} \circ \mathcal{M}_p \circ \mathcal{J}^{-\tau}_{P,R} \circ \boldsymbol{I}, \\ & \biguplus_{p=0}^{P+1} \mathcal{D} \circ \mathcal{M}_p \circ \mathcal{J}^{0}_{P,R} \circ \boldsymbol{I} \Bigg], \end{aligned} \tag{18}$$

where $\biguplus$ denotes vector concatenation.

Compared with the fractal dimension used in previous fractal-based methods, the lacunarity in the proposed method considers more details of distribution. Recall the variable $n(\boldsymbol{B}, r, m)$ in the calculation of lacunarity, which considers the mass of points falling into the intersection. In the case of fractal dimension, m degrades to a binary variable which corresponds to whether the intersection between the mesh and point set is empty. Thus, our lacunarity-based features can be more distinct than the fractal-dimension-based ones. See an example in Fig. 8, which shows that our lacunarity-based features can reflect the changes of gaps in binary images.

4.4. Feature integration

Considering the similarity of pattern distribution in the DT slices along the same axis, we calculate the mean vector of the descriptions of all slices along the X, Y and T axis respectively. This averaging operation both reduces the complexity and enhances the robustness of the resultant features. Finally, our STLS descriptor for DT description, denoted by $\mathcal{F}$, is defined as the concatenation of all the three mean vectors,



Fractal Dim of $\boldsymbol{B} = 1.46$
$P(\boldsymbol{B}) = 0.20, L(\boldsymbol{B}) = 0.53$

Fractal Dim of $\boldsymbol{B} = 1.46$
$P(\boldsymbol{B}) = 0.23, L(\boldsymbol{B}) = 0.67$

Figure 8: Two binary images with different irregularities of point distribution. Both two images have the same fractal dimension calculated by the box-counting method [20, 21, 59], while their lacunarity-based features computed by Eqn. (10) with scale range $[2, 8]$ are different.

i.e.,

$$\begin{aligned}\mathcal{F} \circ \boldsymbol{V} = \Bigg[& \frac{1}{K} \sum_{t=1}^{K} \mathcal{S}_{P,R}^{\tau} \circ \mathcal{L}_t^T \circ \boldsymbol{V}, \\ & \frac{1}{N} \sum_{y=1}^{N} \mathcal{S}_{P,R}^{\tau} \circ \mathcal{L}_y^Y \circ \boldsymbol{V}, \\ & \frac{1}{M} \sum_{x=1}^{M} \mathcal{S}_{P,R}^{\tau} \circ \mathcal{L}_x^X \circ \boldsymbol{V} \Bigg].\end{aligned} \tag{19}$$

Recall that (P, R) and τ are the parameters for both the positive and negative LBPs and the sign-less LBP. These parameters can be set different for different types of LBP and for different axes. For simplicity, we set them the same for all axes and all LBP coding schemes. In practice, we use multiple (P, R)s to improve the discriminability of features.[7]

To illustrate the power of our STLS descriptor, we calculated it on the three types of DT sequences shown in Fig. 1 with $(P, R, \gamma) = (8, 5, 5)$. The results are shown in Fig. 9, which demonstrates both the inter-class discrimination and intra-class similarity of our method.

[7] This strategy does not change the outline of our method, as it only adds more binary images to the second step of our method.

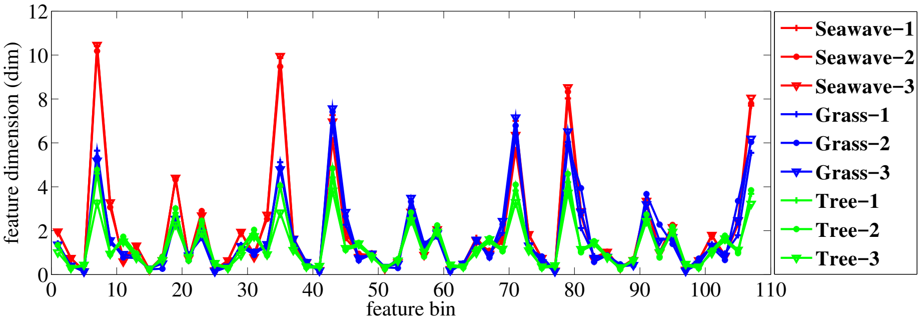


Figure 9: Illustration of the power of our method via comparing the STLS descriptors on three types of DT sequences shown in Fig. 1.

5. Experimental Evaluation

In this section, we present a detailed experimental evaluation on our method, which is conducted on two public benchmark datasets with several breakdowns. Instead of using single-scale LBPs, we used multiple (P, R)s in the test for performance improvement. 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, i.e., 1, 2, 3, 5, are used. The sampling number P in each scale is set to be a multiple of 4, i.e., 4, 16. When the scale R is large, multiple Ps are considered for the purpose of allowing more freedom in large scale. The parameter τ is set to 5, as suggested in [53]. According to the aforementioned protocol, the parameters $\{(P_i, R_i, \tau_i)\}_i$ are set to be $(4, 1, 5), (16, 2, 5), (16, 3, 5), (16, 5, 5)$ in multi-scale local pattern encoding. The box sizes used for calculating lacunarity spectra are set to be integers from 2 to 8. With this setting, our method finally generates a 1080-dimensional DT descriptor for each DT sequence. In implementation, we used look-up tables for accelerating the local pattern coding process and used integral images for accelerating the calculation of mass histogram in lacunarity.

5.1. Datasets and configurations

There are mainly two public DT datasets for evaluating DT classification methods: the UCLA dataset [4] and the DynTex dataset [38, 49]. Note that constructing

Table 1: Configurations of different breakdowns of the test datasets

Dataset	Breakdown	#Samples / #Classes	#Classes	#Training set / #Classes
UCLA	UCLA-50	4	50	3
	UCLA-9	4∼108	9	2
	UCLA-8	4∼20	8	2
	UCLA-7	8∼240	7	4
	UCLA-SIR	8	50	4
DynTex	Basic	10	35	9
	PlusPlus	100	36	50
	Alpha	20	3	5
	Beta	7∼20	10	5
	Gamma	7∼38	10	5

DT sequences is much more difficult than taking photos of static texture images, thus only a limited number of DT datasets are available in current research. To remedy this problem, many studies rearrange the datasets to generate different breakdowns (i.e. sub datasets) for evaluation. The configurations of all these breakdowns used in our experiment are summarized in Table 1 and will be detailed in the following subsections. To remove the benefits of color to classification, all color slices of DT sequences were converted to gray-scale images throughout the experiment.

5.1.1. The UCLA-DT Dataset

The UCLA-DT dataset has been widely used in many previous studies (e.g. [4, 23, 28, 35, 49]). It originally contains 50 DT categories, each with four video sequences captured from different viewpoints. Each video sequence includes 75 frames with 160×110 pixels. Figure 10 shows some samples from the dataset. For the purpose of reuse as well as adding challenges and reducing ambiguity in evaluation, the dataset is reorganized into five different breakdowns for evaluating DT classification algorithms:

- *50-Category [49]:* The original 50 categories are directly used for classification, with 75% samples (i.e. 3 sequences) per category as training set.

- *9-Category [49]:* The original 50 DT categories are clustered to 9 categories by

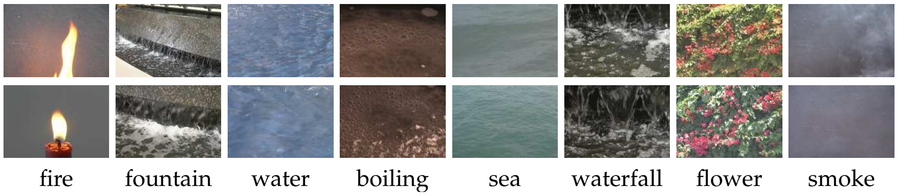


Figure 10: Sample snapshots taken from the UCLA-DT dataset.

combining the sequences from different viewpoints. Then 50% samples (i.e. two training sequences) per category are used for training.

- *8-Category [35]:* The 9 categories of above are further reduced to 8 categories by removing the category which contains too many sequences. One half of samples per category are used for training.
- *7-Category [23]:* The 400 sequences are obtained by cutting 200 video sequences into non-overlapping parts. These sequences were represented into 7 categories. For training 4 samples per category are used.
- *Shift-invariant recognition (SIR) [23]:* Each of the original 200 video sequences is cut into non-overlapping parts. Specifically, each sequence is spatially partitioned into left and right halves and 400 sequences are obtained in the end. The test was implemented by comparing the sequences only between different halves to test the shift-invariance of the descriptors. Note that the intra-class variations in this setting is much larger than other settings.

5.1.2. The DynTex Dataset

The DynTex dataset [60] is a diverse collection of high-quality dynamic texture videos. It contains more than 650 DT sequences, ranging from struggling flames to whelming waves, from sparse curling smoke to dense swaying branches. These video sequences were taken under different environmental conditions involving scaling and rotation, by using static cameras as well as moving ones. See Fig. 11 for the samples in DynTex. The DynTex dataset has been used for DT classification experiments in many previous studies by different rearrangements.

- *Basic [15]:* The basic version of the DynTex dataset contains 35 DT categories, with 10 samples per category. The samples of each category are generated from the same original DT sequence in the big DynTex pool by manual panning.

- *PlusPlus [49]:* The DynTex++ dataset consists of 36 DT categories, each of which contains 100 sequences of a fixed size $50 \times 50 \times 50$. The dataset is well-designed to provide a reasonable benchmark for DT recognition. One half of samples per category are used for training.

- *Alpha/Beta/Gamma [21]:* These three datasets are composed of 60/162/275 DT sequences divided into 3/10/10 categories. The number of samples each category is not uniform. For all these datasets, the training set was constructed by randomly picking up 5 samples per category. Note that noticeable variations of illumination, viewpoint changes and appearance can be observed in these settings.

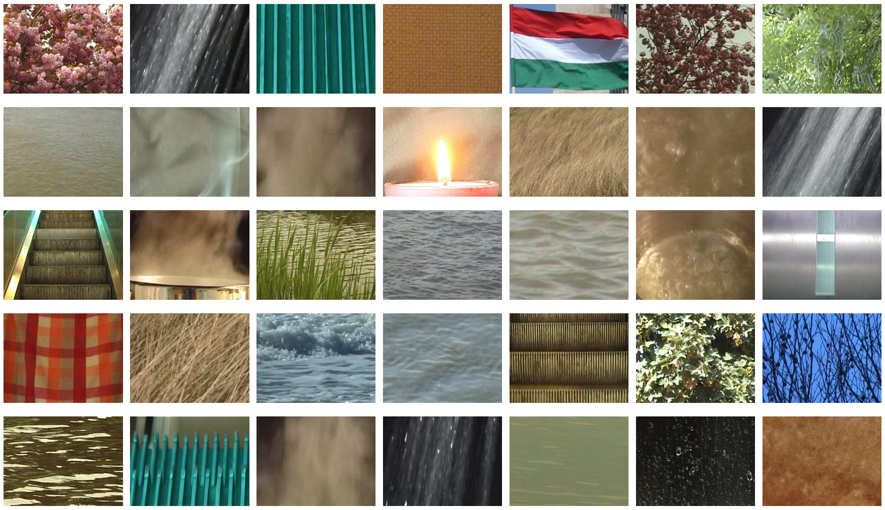


Figure 11: Sample snapshots taken from the DynTex dataset.

5.2. Classification Results

We compare our method with several recent DT classification approaches, including LBP-TOP (Local Binary Patterns from Three Orthogonal Planes) [15], DL-PEGASOS (Distance Learning method based on the PEGASOS algorithm) [49], SODM (Space-time Orientation Distribution Matching) [23], DFS+ (Dynamic Fractal Spectrum Plus) [21], OTF (Oriented Template based Feature) [19], WMFS (Wavelet-based Multi-Fractal Spectrum) [20], OTDL (Orthogonal Tensor Dictionary Learning) [47] and ASTF (Aggregated Spatial and Temporal Features) [50]. The preliminary conference version of our work [26], denoted by PRE, is also included for comparison. Besides, for verifying the improvement of lacunarity analysis over fractal analysis, a baseline method, denoted by BASE, was implemented by replacing the lacunarity analysis with fractal analysis in the proposed framework. The reported results of the compared methods are available in the literature or obtained by running the codes which are available online with parameters finely tuned up. The classifier used in the classification stage is the RBF-kernel SVM. The classification accuracy is reported as the average over a number of trials.

Table 2: Classification accuracies (%) of all compared methods on the UCLA-DT dataset.

Method	50-Class	9-Class	8-Class	7-Class	SIR
DL-PEGASOS [49]	99.0	95.6	-	-	-
SODM [23]	81.0	-	-	92.3	60.0
DFS+ [21]	**100**	97.5	99.2	98.6	74.2
OTF [19]	87.1	97.2	**99.5**	98.4	67.5
WMFS [20]	99.8	97.1	97.0	98.5	61.3
OTDL [47]	99.8	**98.2**	**99.5**	**99.5**	75.2
ASTF [50]	**100**	-	-	-	-
PRE [26]	99.7	96.8	99.2	98.1	74.9
BASE	99.1	96.2	99.0	97.8	72.3
Ours	99.5	97.4	**99.5**	98.4	**75.5**

5.2.1. Results on the UCLA-DT dataset

The classification accuracies on the UCLA-DT dataset are shown in Tab. 2, from which it can be seen that our method is very competitive. It is noted that with the great development of DT classification techniques, the UCLA dataset is losing its challenge. Many methods have classification accuracies over 95% on the UCLA dataset except in the SIR setting. In particular, there is little difference on accuracies among DFS+, WMFS, ST-PLS and our method in the 7-Class, 8-Class, 9-Class and 50-Class breakdowns. Even so, such results have still demonstrated the comparable performance of our descriptor with the state-of-the-art methods and shown our method performed consistently well on easy data.

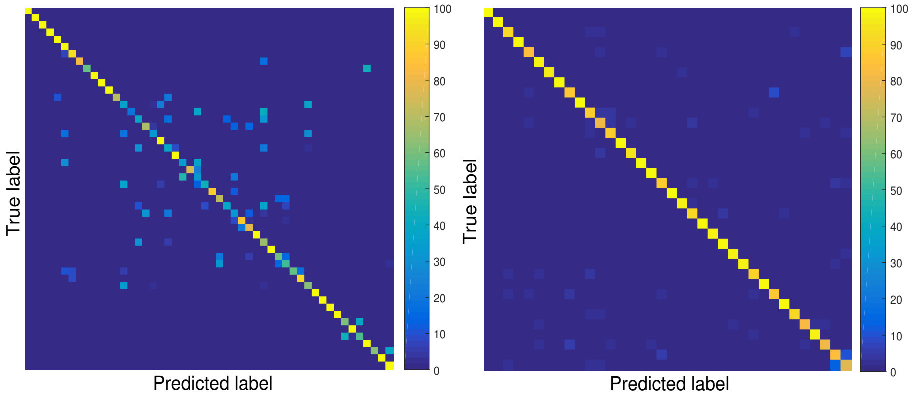


Figure 12: Confusion matrices by our approach on the UCLA-SIR dataset (left) and the DynTex++ dataset (right). The horizontal and vertical axes denote the class indices.

Among all the breakdowns, the SIR test is the most challenging as it evaluates the robustness to viewpoint changes on the descriptor. In this case, our approach outperformed all the compared methods. The improvement of STLS over its preliminary version comes from the use of the improved version of LBPs defined by Eqn. (12)-(14) instead of the original ones used in [26]. Also note that the superior performance our method to BASE, especially in the SIR test, has demonstrated that improvement of lacunarity analysis over fractal analysis. The confusion matrix by our method in the SIR setting is shown in Fig. 12.

By comparing the results of BASE and ours on all the breakdowns, we can verify that using lacunarity spectrum analysis indeed bring improvement over fractal spectrum analysis.

5.2.2. Results on the DynTex dataset

The experimental results on the DynTex dataset are summarized in Tab. 3. It can be seen that our approach is very competitive. The Basic breakdown is the most easy one where all the compared methods have accuracies over 96%. On the PlusPlus breakdown, our approach achieved the third best result with 0.2% accuracy less than OTDL and 0.3% less than PRE. Notice that OTDL only focuses on learning local DT features, which can be combined with our lacunarity-based global description. The weakness of OTDL is that it does not work well when the training samples are insufficient. The reason that our method performed worse than PRE is that the resolution of samples in the PlusPlus breakdown is small and our uniform parameters are not suitable for this case. With finely tunned-up parameters, our method can perform on a par with PRE. The confusion matrix of our method on the PlusPlus breakdown is shown in Fig. 12.

On the other three breakdowns, our method outperformed other methods except ASTF and PRE. The performance gap between our method and our preliminary work is very small, while ASTF achieved a really good classification accuracy on the Gamma breakdown. But note that the focus of ASTF is not on developing new types of DT features but on feature selection and aggregation using ensemble methods. The accuracy as well as improvement of ASTF heavily relies on the development of useful DT features, and ASTF can also be combined with our proposed descriptor.

In general, the performance of our method on the DynTex dataset degrades a bit compared to our preliminary work. However, as can be seen in the next subsection, our descriptor is much more compact and faster than the preliminary one. Here, the advantage of the proposed approach which we emphasize over its preliminary version is that, both the representation size and computational time are largely reduced while the classification accuracy exhibits very slight decrease on average. Again, by comparing the results of BASE and ours on all the breakdowns, we can verify that using lacunarity spectrum analysis indeed bring improvement over fractal spectrum analysis.

In DT classification, it is interesting to check whether utilizing temporal cues indeed bring improvement to the accuracy. With this purpose, we conducted an additional test on DynTex++, which was done by discarding the features computed on the XT and

Table 3: Classification accuracies (%) of all compared methods on the DynTex dataset.

Method	Basic	PlusPlus	Alpha	Beta	Gamma
LBP-TOP [15]	97.1	89.8	83.3	73.4	72.0
DL-PEGASOS [49]	-	63.7	-	-	-
DFS+ [21]	97.2	91.7	85.2	76.9	74.8
OTF [19]	96.7	89.2	-	-	-
WMFS [20]	96.5	88.8	-	-	-
OTDL [47]	**99.0**	94.7	87.8	76.7	74.8
ASTF [50]	-	-	-	-	**99.5**
PRE [26]	97.9	**94.8**	**89.6**	**80.9**	79.9
BASE	97.5	93.0	86.7	80.2	76.8
Ours	98.2	94.5	89.4	80.8	79.8

YT planes and only keeping the feature computed on the XY plane. The classification accuracy drops from 94.5% to 86.7%. In other words, the use of temporal features in our method can significantly improve the classification accuracy. Such a result also demonstrates the need of considering motion cues in recognizing textures in dynamic scenes.

5.3. Efficiency

Our method has demonstrated its discriminative power on the UCLA and DynTex datasets. Next, we tested the computational efficiency of our method in terms of feature length and running time, DFS+, OTF and WMFS. For fair comparison, all the compared methods are implemented in MATLAB and tested on the same desktop computer with Intel Xeon E3-1230 V2 3.30GHz CPU and 32GB memory. For the time test, we report the average running time per sample on the DynTex++ dataset.

The experimental results are listed in Table 4. It can be seen that the length of our descriptor are much shorter than our preliminary version. The ratio of lengths is around $1/3$. Such shorter length can benefit the reduction of the time cost in classifier training. Meanwhile, benefiting from the shorter feature length which saves time in computing additional lacunarity features, our method is much faster than our preliminary version [26].[8] The running time for a DT sequence of PRE is about two times

[8] The local pattern coding schemes of both our method and PRE are with efficient lookup-table-based

as ours. Thus, we can conclude that our method have the same level of discriminative power as our preliminary version, while have much less computational complexity and feature length. Compared with other methods, our method do not show advantages in the feature length and running time. But recall that the performance of our method is better than these methods.

Table 4: feature length and running time (s) of several tested methods on the DynTex++ dataset.

Method	Feature Length	Running Time (s)
LBP-TOP [15]	768	1.2
DFS+ [21]	500	6.6
OTF [19]	290	22
WMFS [20]	702	9.8
PRE [26]	3456	81.1
Ours	1080	39.8

5.4. Influence of parameters

To analyze the influence of parameter setting in the proposed method, we tested the performance of our STLS descriptor by adjusting one of the parameters (P, R) and τ while keeping the other unchanged. The test which varies (P, R) and fixes τ was done on the DynTex PlusPlus, Alpha, Beta and Gamma datasets. The results are shown in Fig. 13, from which we can observe that the performance of STLS is not sensitive to single (P, R) within a reasonably small range, and when combing multiple (P, R)s, the performance has noticeable increase. The test which adjusts τ while fixing (P, R) was done on the Alpha, Beta and Gamma breakdowns of DynTex, and the results are shown in Fig. 14. It can be observed that the performance of STLS exhibits a small disturbance when τ is within $[1, 8]$.

implementation. Thus, the main difference in running time comes from how many lacunarity-based features are to compute, which is directly determined on the number of feature bins, i.e. feature length.

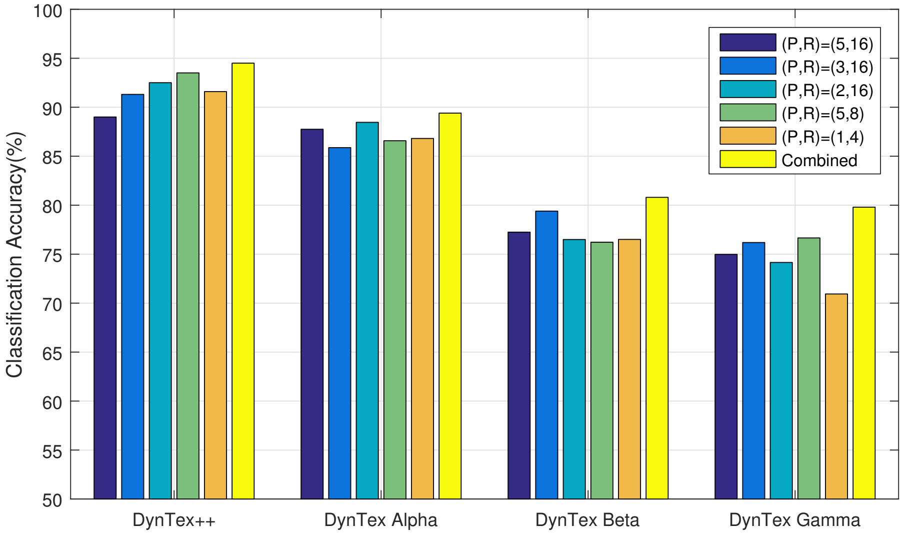


Figure 13: Influence of selection of (P, R) to the performance of the proposed method on the four breakdowns of DynTex. Here τ is fixed to be 5.

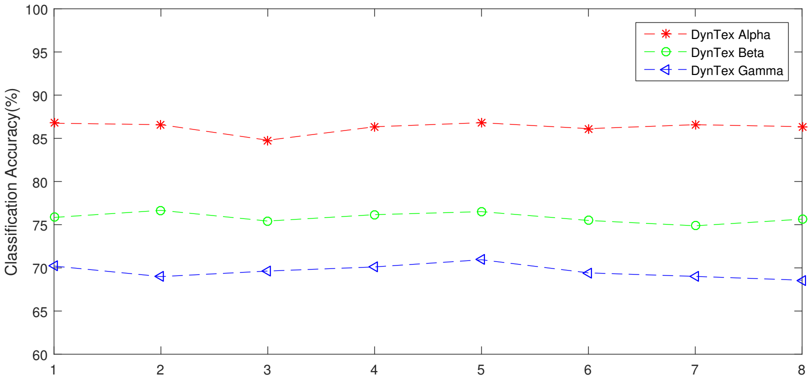


Figure 14: Influence of selection of τ to the performance of the proposed method on the three breakdowns of DynTex. Here (P, R) is fixed to be $(1, 4)$.

6. Conclusion

We developed a powerful DT descriptor by using lacunarity analysis on local binary patterns along two spatial axes and one temporal axis. The proposed method decomposes a DT sequence into multiple spatial and temporal binary pattern maps and characterizes each feature map with lacunarity analysis. The resulting DT descriptor enjoys both strong robustness and high discriminability. Experiments on several benchmark datasets have demonstrated the power of our method. The major advantage of our approach over its preliminary conference version is that both the representation size and computational time are largely reduced while the classification accuracy exhibits very slight decrease on average.

The major limitation of the proposed method is the lack of rotation invariance as it separately processes slices along different axes. In future, we would like to investigate the remedy to the rotation invariance of slice-wise processing. As can be seen in the experiment, existing datasets for DT classification are very limited. Thus, we would also build up more challenging DT benchmark datasets for DT classification and propose new methodologies for evaluating DT classification algorithms. Moreover, we would like to investigate possible combination of histogram, fractal dimensions and lacunarity spectrum and its application to dynamic scene recognition.

Acknowledgment

We thank Drs. G. Doretto, B. Ghanem, and G.Zhao for their help with the datasets.

References

[1] M. Abdel-Mottaleb, S. Krishnamachari, Multimedia descriptions based on mpeg-7: extraction and applications, IEEE Trans. Multimedia 6 (3) (2004) 459–468.

[2] J. Jung, M. Antonini, M. Barlaud, Optimal decoder for block-transform based video coders, IEEE Trans. Multimedia 5 (2) (2003) 145–160.

[3] A. Rahman, M. Murshed, Temporal texture characterization: a review, in: Computational Intell. Multimedia Process.: Recent Advances, 2008, pp. 291–316.

[4] G. Doretto, A. Chiuso, Y. Wu, S. Soatto, Dynamic textures, Int. J. Comput. Vision 51 (2) (2003) 91–109.

[5] D. Chetverikov, R. Péteri, A brief survey of dynamic texture description and recognition, in: Comput. Recognition Syst., 2005, pp. 17–26.

[6] D. Tiwari, V. Tyagi, Dynamic texture recognition: a review, in: Inform. Syst. Design and Intelligent Applicat., Springer, 2016, pp. 365–373.

[7] A. Ekin, A. M. Tekalp, R. Mehrotra, Integrated semantic-syntactic video modeling for search and browsing, IEEE Trans. Multimedia 6 (6) (2004) 839–851.

[8] J. Shao, Z. Huang, H. T. Shen, X. Zhou, E.-P. Lim, Y. Li, Batch nearest neighbor search for video retrieval, IEEE Trans. Multimedia 10 (3) (2008) 409–420.

[9] G. Zhai, J. Cai, W. Lin, X. Yang, W. Zhang, M. Etoh, Cross-dimensional perceptual quality assessment for low bit-rate videos, IEEE Trans. Multimedia 10 (7) (2008) 1316–1324.

[10] K. Hariharakrishnan, D. Schonfeld, Fast object tracking using adaptive block matching, IEEE Trans. Multimedia 7 (5) (2005) 853–859.

[11] C. Zhu, X. Sun, F. Wu, H. Li, Video coding with spatio-temporal texture synthesis and edge-based inpainting, in: Proc. IEEE Int. Conf. Multimedia and Expo, 2008, pp. 813–816.

[12] H. Li, G. Liu, K. N. Ngan, Guided face cartoon synthesis, IEEE Trans. Multimedia 13 (6) (2011) 1230–1239.

[13] C. M. Funke, L. A. Gatys, A. S. Ecker, M. Bethge, Synthesising dynamic textures using convolutional neural networks, arXiv preprint arXiv:1702.07006.

[14] G. Zhao, M. Barnard, M. Pietikäinen, Lipreading with local spatiotemporal descriptors, IEEE Trans. Multimedia 11 (7) (2009) 1254–1265.

[15] G. Zhao, M. Pietikainen, Dynamic texture recognition using local binary patterns with an application to facial expressions, IEEE Trans. Pattern Anal. Mach. Intell. 29 (6) (2007) 915–928.

[16] M. A. Tahir, F. Yan, P. Koniusz, M. Awais, M. Barnard, K. Mikolajczyk, A. Bouridane, J. Kittler, A robust and scalable visual category and action recognition system using kernel discriminant analysis with spectral regression, IEEE Trans. Multimedia 15 (7) (2013) 1653–1664.

[17] Y. Quan, Y. Xu, Y. Sun, Y. Luo, Lacunarity analysis on image patterns for texture classification, in: Proc. IEEE Conf. Comput. Vision and Pattern Recognition, 2014, pp. 160–167.

[18] Y. Xu, Y. Quan, H. Ling, H. Ji, Dynamic texture classification using dynamic fractal analysis, in: Proc. IEEE Int. Conf. Comput. Vision, 2011, pp. 1219–1226.

[19] Y. Xu, S. Huang, H. Ji, C. Fermüller, Scale-space texture description on sift-like textons, Comput. Vision and Image Understanding 116 (9) (2012) 999–1013.

[20] H. Ji, X. Yang, H. Ling, Y. Xu, Wavelet domain multifractal analysis for static and dynamic texture classification, IEEE Trans. Image Process. 22 (1) (2013) 286–299.

[21] Y. Xu, Y. Quan, Z. Zhang, H. Ling, H. Ji, Classifying dynamic textures via spatiotemporal fractal analysis, Pattern Recognition 48 (10) (2015) 3239–3248.

[22] G. Zhao, T. Ahonen, J. Matas, M. Pietikainen, Rotation-invariant image and video description with local binary pattern features, IEEE Trans. Image Process. 21 (4) (2012) 1465–1477.

[23] K. G. Derpanis, R. P. Wildes, Dynamic texture recognition based on distributions of spacetime oriented structure, in: Proc. IEEE Conf. Comput. Vision and Pattern Recognition, 2010, pp. 191–198.

[24] K. G. Derpanis, R. P. Wildes, Spacetime texture representation and recognition based on a spatiotemporal orientation analysis, IEEE Trans. Pattern Anal. Mach. Intell. 34 (6) (2012) 1193–1205.

[25] K. G. Derpanis, M. Lecce, K. Daniilidis, R. P. Wildes, Dynamic scene understanding: The role of orientation features in space and time in scene classifica-

tion, in: Proc. IEEE Conf. Comput. Vision and Pattern Recognition, 2012, pp. 1306–1313.

[26] Y. Sun, Y. Xu, Y. Quan, Characterizing dynamic textures with space-time lacunarity analysis, in: Proc. IEEE Int. Conf. Multimedia and Expo, 2015, pp. 1–6.

[27] M. Szummer, R. W. Picard, Temporal texture modeling, in: Proc. IEEE Int. Conf. Image Process., Vol. 3, 1996, pp. 823–826.

[28] P. Saisan, G. Doretto, Y. Wu, S. Soatto, Dynamic texture recognition, in: Proc. IEEE Conf. Comput. Vision and Pattern Recognition, Vol. 2, 2001, pp. 58–63.

[29] A. B. Chan, N. Vasconcelos, Probabilistic kernels for the classification of auto-regressive visual processes, in: Proc. IEEE Conf. Comput. Vision and Pattern Recognition, Vol. 1, 2005, pp. 846–851.

[30] A. B. Chan, N. Vasconcelos, Classifying video with kernel dynamic textures, in: Proc. IEEE Conf. Comput. Vision and Pattern Recognition, 2007, pp. 1–6.

[31] B. Ghanem, N. Ahuja, Phase based modelling of dynamic textures, in: Proc. IEEE Int. Conf. Comput. Vision, 2007, pp. 1–8.

[32] B. Ghanem, N. Ahuja, Extracting a fluid dynamic texture and the background from video, in: Proc. IEEE Conf. Comput. Vision and Pattern Recognition, 2008, pp. 1–8.

[33] G. Doretto, E. Jones, S. Soatto, Spatially homogeneous dynamic textures, in: Proc. European Conf. Comput. Vision, 2004, pp. 591–602.

[34] F. Woolfe, A. Fitzgibbon, Shift-invariant dynamic texture recognition, in: Proc. European Conf. Comput. Vision, 2006, pp. 549–562.

[35] A. Ravichandran, R. Chaudhry, R. Vidal, View-invariant dynamic texture recognition using a bag of dynamical systems, in: Proc. IEEE Conf. Comput. Vision and Pattern Recognition, 2009, pp. 1651–1657.

[36] R. Polana, R. Nelson, Motion-Based Recognition, Springer, 1997, Ch. Temporal texture and activity recognition, pp. 87–124.

[37] Y. Wang, S.-C. Zhu, Modeling textured motion: Particle, wave and sketch, in: Proc. IEEE Int. Conf. Comput. Vision, 2003, pp. 213–220.

[38] R. Péteri, D. Chetverikov, Dynamic texture recognition using normal flow and texture regularity, in: Pattern Recognition and Image Anal., 2005, pp. 223–230.

[39] Z. Lu, W. Xie, J. Pei, J. Huang, Dynamic texture recognition by spatio-temporal multiresolution histograms, in: IEEE Workshops on Applicat. of Comput. Vision, Vol. 2, 2005, pp. 241–246.

[40] R. P. Wildes, J. R. Bergen, Qualitative spatiotemporal analysis using an oriented energy representation, in: Proc. European Conf. Comput. Vision, 2000, pp. 768–784.

[41] S. R. Arashloo, J. Kittler, Dynamic texture recognition using multiscale binarized statistical image features, IEEE Trans. Multimedia 16 (8) (2014) 2099–2109.

[42] J. R. Smith, C. Lin, M. Naphade, Video texture indexing using spatio-temporal wavelets, in: Proc. IEEE Int. Conf. Image Process., Vol. 2, 2002, pp. 437–440.

[43] X. You, L. Du, Y.-m. Cheung, Q. Chen, A blind watermarking scheme using new nontensor product wavelet filter banks, IEEE Trans. Image Process. 19 (12) (2010) 3271–3284.

[44] D. Teney, M. Brown, Segmentation of dynamic scenes with distributions of spatiotemporally oriented energies, in: Proc. IEEE British Machine Vision Conf., University of Bath, 2014.

[45] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.

[46] G. Zhao, M. Pietikäinen, Dynamic texture recognition using volume local binary patterns, in: Dynamical Vision, 2007, pp. 165–177.

[47] Y. Quan, Y. Huang, H. Ji, Dynamic texture recognition via orthogonal tensor dictionary learning, in: Proc. IEEE Int. Conf. Comput. Vision, 2015, pp. 73–81.

[48] M. Favorskaya, A. Pyataeva, Convolutional recognition of dynamic textures of preliminary categories., Int. Archives of the Photogrammetry, Remote Sensing and Spatial Inform. Sciences 42.

[49] B. Ghanem, N. Ahuja, Maximum margin distance learning for dynamic texture recognition, in: Proc. European Conf. Comput. Vision, 2010, pp. 223–236.

[50] F. Yang, G.-S. Xia, G. Liu, L. Zhang, X. Huang, Dynamic texture recognition by aggregating spatial and temporal features via ensemble svms, Neurocomputing 173 (2016) 1310–1321.

[51] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, Pattern Recognition 29 (1) (1996) 51–59.

[52] T. Ojala, M. Pietikäinen, T. Mäenpää, Gray scale and rotation invariant texture classification with local binary patterns, in: Proc. European Conf. Comput. Vision, 2000, pp. 404–420.

[53] X. Tan, B. Triggs, Enhanced local texture feature sets for face recognition under difficult lighting conditions, IEEE Trans. Image Process. 19 (6) (2010) 1635–1650.

[54] A. P. Pentland, Fractal-based description of natural scenes, IEEE Trans. Pattern Anal. Mach. Intell. (6) (1984) 661–674.

[55] C. Allain, M. Cloitre, Characterizing the lacunarity of random and deterministic fractal sets, Physical review A 44 (6) (1991) 3552–3558.

[56] V. A. Billock, G. C. de Guzman, J. Scott Kelso, Fractal time and 1/f spectra in dynamic images and human vision, Physica D: Nonlinear Phenomena 148 (1) (2001) 136–146.

[57] D. W. Dong, J. J. Atick, Statistics of natural time-varying images, Network: Computation in Neural Syst. 6 (3) (1995) 345–358.

[58] J. Van Hateren, Processing of natural time series of intensities by the visual system of the blowfly, Vision research 37 (23) (1997) 3407–3416.

[59] K. Falconer, Techniques in fractal geometry, Vol. 16, Wiley Chichester (W. Sx.), 1997.

[60] R. Péteri, S. Fazekas, M. J. Huiskes, Dyntex: A comprehensive database of dynamic textures, Pattern Recognition Lett. 31 (12) (2010) 1627–1632.