Classifying dynamic textures via spatiotemporal fractal analysis ☆

Yong Xu [a], Yuhui Quan [a,*], Zhuming Zhang [b], Haibin Ling [c], Hui Ji [d]

[a] School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China
[b] Department of Computer Science & Engineering, The Chinese University of Hong Kong, Hong Kong, China
[c] Department of Computer & Information Sciences, Temple University, Philadelphia, PA 19122, USA
[d] Department of Mathematics, National University of Singapore, Singapore 117542, Singapore

ABSTRACT

The large-scale images and videos are one kind of the main source of big data. Dynamic texture (DT) is essential for understanding the video sequences with spatio-temporal similarities. This paper presents a powerful tool called dynamic fractal analysis to DT description and classification, which integrates rich description of DT with strong robustness to environmental changes. The proposed dynamic fractal spectrum (DFS) for DT sequences is composed of two components. The first one is a volumetric dynamic fractal spectrum component (V-DFS) that captures the stochastic self-similarities of DT sequences by treating them as 3D volumes; the second one is a multi-slice dynamic fractal spectrum component (S-DFS) that encodes fractal structures of repetitive DT patterns on 2D slices along different views of the 3D volume. To fully exploit various types of dynamic patterns in DT, five measurements of DT pixels are collected for the analysis on DT sequences from different perspectives. We evaluated our method on four publicly available benchmark datasets. All the experimental results have demonstrated the excellent performance of our method in comparison with state-of-the-art approaches.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

The explosive growth in the amount of data makes big data processing and analytics one of the hottest research topics. Roughly speaking, big data analytics aims at examining a large amount of data from various sources to uncover hidden patterns, unknown correlations as well as other useful information. One of the most-visible sources of big data is video, which is being generated pervasively by billions of sensors embedded in various types of devices like surveillance cameras and mobile phones. For analyzing the underlying patterns captured by videos, a fundamental issue is the feature extraction and description of dynamic patterns which are often in the form of dynamic texture.

Dynamic textures (DTs) are often regarded as video sequences of moving scenes that possess certain stationary properties in both space domain and time domain [1,2]. Such video sequences are ubiquitous in real world, like video clips of boiling water, rivers, sea waves, fountains, clouds, smoke, fire, swarm of birds, traffic flow, pedestrians in crowds, whirligig, facial expressions, etc. There are many applications concerning DT, such as video compression, video quality assessment, surveillance, detection of the onset of emergencies, foreground/background separation, and human–computer interaction; see e.g. [3–6]. In recent years, the related topics of DT in computer vision community have ranged from DT modeling and synthesis to recognition and classification. In this paper, we focus on the development of effective DT description and classification techniques, which can be integrated to many recognition systems that involve the characterization of dynamics, e.g. vision sensor based fire detection, DT segmentation based dynamic scene retrieval, real-time facial expression analysis, biometrics, etc.

Compared to static textures, dynamic textures vary not only on the spatial distribution of texture elements, but also on the organization and dynamics over time. One main challenge in the study of DT classification is how to reliably capture the motion behaviors of texture elements, i.e. the properties of dynamics of texture elements over time. Many existing approaches model the dynamics either by treating videos as samples of stochastic dynamical systems or by directly measuring the motion field of

videos, which are suitable for dynamic textures with regular motions. However, the effectiveness of existing approaches is not satisfactory for dynamic textures with complex motions driven by non-linear stochastic dynamic systems with certain chaos, e.g. turbulent water and bursting fire. This inspired us to develop an effective DT descriptor for classifying DT sequences with complex dynamic behaviors.

1.1. Related work

There are many DT classification approaches, which could be roughly categorized as either generative or discriminative methods. The generative methods attempt to quantitatively model the underlying physical dynamic system that generates DT sequences, and classify DT sequences based on the system parameters of the corresponding physical model. The main difference of these methods lies in the models they build up, e.g. the spatio-temporal autoregressive model [7] and its multi-scale version [8], the linear dynamical systems [9,10], the kernel-based model [11], and the phase-based model [12]. The main drawback of the generative methods is the inflexibility to describe the DT sequences generated by nonlinear physical systems with complex motion irregularities.

In contrast to the generative methods, the discriminative methods are able to describe DT effectively without explicitly modeling the underlying dynamic system. The basic idea of the discriminative methods is to characterize the distribution of local DT patterns. To efficiently extract local DT patterns, many methods have been proposed, e.g. the spatio-temporal filtering for specific motion patterns [13,14], the spatiotemporal extensions of local binary pattern (LBP) encoding [6], the wavelet pattern extraction [5,15], the optical flow based pattern estimation [1,16,17], the space–time oriented pattern analysis [18–20], etc. In practice, the discriminative methods exhibit better performance than the generative methods in DT classification and show advantages in the robustness to environmental changes and viewpoint changes. However, the merits of existing discriminative methods are quite limited in the case of DTs with complex motions, as they are not capable of reliably capturing inherent stochastic stationary properties of such video sequences.

1.2. Motivation and contribution

Reliable characterization on DT motion behaviors is crucial to the development of an effective DT descriptor. We notice that although the motion patterns of many DT sequences could be highly irregular, they are quite consistent when viewed from different spatial and temporal scales. In other words, similar mechanisms are operating at various spatial and temporal scales in the underlying physical dynamics. Such multi-scale self-similarities are referred as to *power law* or *fractal structure* [21]. In fact, the existence of fractal structures in a large spectrum of dynamic nature images has been observed by many researchers, e.g., the amplitude of temporal frequency spectra of many video sequences, including camera movements, weather and biological movements by one or more humans, indeed fits power-law models [22,23,21,24,25].

In this paper, motivated by the existence of stochastic self-similarities in a wide range of DTs, we propose to model DTs by using non-linear stochastic dynamic systems with certain inherent multi-scale self-similarities, i.e., dynamic textures are likely to be generated by some mechanism with similar stochastic behaviors operating at various spatial and temporal scales. A novel method called dynamic fractal analysis is proposed for DT description, which measures such self-similarities of the underlying system based on fractal geometry. The proposed method can be viewed as a discriminative method with generative motivation, as we assume that DT sequences are generated by some dynamic systems with self-similarities. The resulting DFS (dynamic fractal spectrum) descriptor allows us to bypass the quantitative estimation of the underlying physical model, which is challenging in practice. Meanwhile, the DFS descriptor has the merits of both categories of methods: the discriminative power of generative methods for modeling stochastic behaviors of DT and the robustness of discriminative methods to environmental changes.

A preliminary conference version of this work appeared in [26]. The main extensions of this paper include the development of an additional spatio-temporal measure of DT pixels that brings extra discriminability, the evaluation on an additional test dataset, and more detailed analysis on the proposed method. It is noted that fractal analysis has been exploited in recent literature for DT recognition; see e.g. [14,15]. These methods mainly focus on static texture classification and are applied to DT classification by either simply averaging the original features on each DT frame (e.g. [15]), or directly extending the descriptors to 3D case (e.g. [14]). Compared with these fractal-based methods, our method captures both the global self-similar behaviors on an entire DT sequence and the statistical self-similarities of the repetitive patterns on each DT slice. Thus, our method enjoys higher discriminative power in DT classification.

2. Basics on fractal analysis

Before presenting the details of the proposed method, we first briefly introduce the theory and numerical implementation of fractal analysis. Interested readers are referred to [27–29] for more details. Fractal analysis is built on the concept of *fractal dimension* which was first proposed by Mandelbrot [28] as a description for power laws. The power laws exist in numerous natural phenomena, e.g., the amplitude of temporal frequency spectra $A(f)$ of many video sequences fits $1/f^\beta$ power-law models [22,23,21]

$$A(f) \propto f^{-\beta}, \tag{1}$$

where f denotes the frequency.

The fractal dimension is about self-similarity defined as the power law which the measurements of objects obey at various scales. One widely used fractal dimension in Geophysics and Physics is the so-called *box-counting* fractal dimension. Let the n-dimensional Euclidean space $\mathbb{R}^n$ be covered by a mesh of n-dim hypercubes with diameter $\frac{1}{m}$. Given a point set $E \subset \mathbb{R}^n$, the *box-counting* fractal dimension $\beta(E)$ of E is defined as the following [27]:

$$\beta(E) = \lim_{m \to \infty} \frac{\log \# \left(E, \frac{1}{m} \right)}{-\log \frac{1}{m}}, \tag{2}$$

where $\#(E, \frac{1}{m})$ is the number of mesh hypercubes that intersect E for $m = 1, 2, \ldots$. In numerical implementation, it can be done by using least squares fitting in the log–log coordinate system with a finite sequence of ordered integers.

For the physical phenomena with mixtures of multiple fractal structures, the so-called multi-fractal analysis extends the fractal dimension to describe and distinguish more complex self-similarity behavior of the physical dynamic systems. The extension is done as follows. Instead of assuming all points in the set generated by the same mechanism, a measure μ is first defined such that μ obeys the local power law in terms of scale

$$\mu(B_r(x)) \propto r^{\alpha(x)}, \tag{3}$$

where $B_r(x)$ is a closed Borel hyper sphere with center x and radius r, and $\alpha(x)$ is the Hölder exponent of x that characterizes the local power law around x under the measure μ and can be estimated by the local density function [27]

$$\alpha(x) = \lim_{r \to 0} \frac{\log \mu(B(x,r))}{\log r}. \tag{4}$$

In numerical implementation, the density $\alpha(x)$ can also be estimated by the least square fitting in the log–log coordinate system with a finite sequence of ordered positive radius $r_0 > r_1 > \cdots > r_z$.

The multi-fractal analysis is defined as a function $f(\widehat{\alpha})$ that collects the fractal dimensions of each point set in which all points have the same Hölder exponent

$$f(\widehat{\alpha}) = \beta(E_{\widehat{\alpha}}), \tag{5}$$

where $E_{\widehat{\alpha}} = \{x : \alpha(x) = \widehat{\alpha}\}$ is the point set whose elements share the same local Hölder exponent. In other words, the multi-fractal analysis is about fractal dimensions of multiple point sets partitioned based on their local multi-scale behaviors on some measure μ.

3. Dynamic fractal analysis for dynamic textures

Assuming DT sequences generated by some non-linear stochastic dynamic systems with certain inherent multi-scale self-similarities as shown in the previous studies [22,23,21], we propose to robustly characterize such self-similarities via multifractal analysis on local DT features. In this section, we develop the dynamic fractal analysis for exploring the fractal structures in DT sequences, and derive a DT descriptor called dynamic fractal spectrum (DFS) which encodes rich discriminative information regarding multi-scale self-similarities for DT classification.

3.1. Spatio-temporal measures of pixels

As presented in Section 2, multi-fractal analysis is conducted on the measure μ which determines how pixels are categorized. For notational convenience, we define

$$\mu(p, t; r_s, r_t) = \mu(B_{(p,t;)}(r_s, r_t)), \tag{6}$$

where $B_{(p_0,t_0)}(r_s, r_t)$ denotes a 3D cube centering at (p_0, t_0) with spatial radius r_s and temporal radius r_t. An accepted measure should partition pixels into different categories based on the intrinsic physical meaning of pixels and the resulting pixel partition should corresponds to different types of repetitive patterns and be robust to environmental changes. In our dynamic fractal analysis, the following five measures are chosen to examine DT from different perspectives.

Pixel intensity: Given a gray-scale DT sequence $I(\cdot, t)$ for $t = 1, 2, \ldots$, let $I(p, t)$ denote the intensity value of the pixel p in the sequence $I(\cdot, t)$. A straightforward measure is the *intensity*:

$$\mu_I(p_0, t_0; r_s, r_t) = \int_{B_{(p_0,t_0)}} \int_{(r_s,r_t)} I(p, t)\, dp\, dt, \tag{7}$$

which measures the overall intensity in a space–time neighborhood of the point (p_0, t_0).

Temporal brightness gradient: Besides the spatial measure, the temporal measure also provides an essential cue for describing the DT. Thus, the second measure used in our method is the *temporal brightness gradient*:

$$\mu_B(p_0, t_0; r_s, r_t) = \int_{B_{(p_0,t_0)}} \int_{(r_s,r_t)} \frac{\partial I(p, t)}{\partial t}\, dp. \tag{8}$$

Intuitively, μ_B measures the summation of the temporal intensity changes of DT around the point (p_0, t_0).

Normal flow: Another measure related to temporal information is the *normal flow*:

$$\mu_F(p_0, t_0; r_s, r_t) = \int_{B_{(p_0,t_0)}} \int_{(r_s,r_t)} \frac{\partial I(p, t)/\partial t}{\| \nabla I(p) \|}\, dp. \tag{9}$$

The normal flow is different from the temporal gradient in the sense that it measures the motion of the pixels along the direction perpendicular to the brightness gradient. Thus, it is a measure about edge motion. It is noted that although optical flow is more informative for point-wise motion, it is not used in our analysis because it is a hard task to reliably estimate the optical flow field for chaotic motions.

Laplacian: Besides the first-order information summarized by the gradient measure, the second-order behaviors of DTs are also taken into account. For this purpose, the fourth measure is the *Laplacian*:

$$\mu_L(p_0, t_0; r_s, r_t) = \int_{B_{(p_0,t_0)}} \int_{(r_s,r_t)} \Delta I(p, t)\, dp, \tag{10}$$

which encodes the information of the local co-variance of the pixel intensity at (p_0, t_0) in the spatial–temporal domain.

Principal curvature: The last measure adopted in our dynamic fractal analysis is the *principal curvature*:

$$\mu_C(p_0, t_0; r_s, r_t) = \int_{B_{(p_0,t_0)}} \int_{(r_s,r_t)} \kappa(I, p, t)\, dp, \tag{11}$$

where $\kappa(I, p, t)$ is the principle curvature of the surface I at point (p, t). Principal curvature is often used to measure how a surface bends by different amounts in different directions at a certain point, and it can signify a peak, a valley, or a saddle point, depending on the sign. A positive principal curvature indicates a valley or peak; a negative one indicates a saddle point; and a zero one implies that the surface is flat in at least one direction. Thus, the measure μ_C summarizes the shape information of the local surface of image sequence.

The five measures proposed above conclude the local information of pixels as well as the cues from local DT structures in the spatio-temporal domain from different perspectives, and explore different underlying physical implications. Specifically, the pixel intensity measure μ_I regards pixel brightness, the temporal brightness gradient measure μ_B summarizes the changes of brightness over time, the normal flow measure μ_F encodes reliable temporal changes of edge points, the Laplacian measure μ_L measures the second-order information on brightness changes in the spatio-temporal domain, and the principle curvature measure μ_C considers the local shape of spatio-temporal surface. Fig. 1 shows some examples of these five measures.

3.2. Dynamic fractal spectrum

After defining the above five spatio-temporal measures (i.e. μ_I, μ_B, μ_F, μ_L and μ_C) which describe the local DT structures in multiple views, we are ready to formulate our dynamic fractal spectrum (DFS) descriptor using multi-fractal analysis. In order to encode rich discriminative information of DT, the DFS descriptor consists of two complementary components: one is the *volumetric DFS* (V-DFS) component that characterizes the statistical self-similarities of the given DT sequence by viewing it as points collected in a 3D volume; the other is the *multi-slice DFS* (S-DFS) component that captures the statistical self-similarities and complexities of DT by regarding the spatial distribution of the repetitive patterns lying in the 2D slices of the 3D volume along three orthogonal axes. The proposed DFS descriptor is outlined in Algorithm 1.

Algorithm 1. Dynamic fractal analysis (DFS).

Input: A DT sequence I
Output: The DFS vector d

1. Compute five measures $\mu_I(x)$, $\mu_B(x)$, $\mu_F(x)$, $\mu_L(x)$, $\mu_C(x)$ for each pixel $x = (p, t)$ of I.
2. Compute local density exponent $\alpha(x)$ for each pixel x of I using (4) with respect to each measure.
3. Compute the DFS as follows:

    V-DFS: Classify each pixel x in I into set $E_{[\alpha_i, \alpha_{i+1})}$ if its Hölder exponent $\alpha(x)$ falls into $[\alpha_i, \alpha_{i+1})$. Then for each set $E_{[\alpha_i, \alpha_{i+1})}$, compute its 3D fractal dimension $\beta(E_{\alpha_i})$ in the whole 3D spatio-temporal domain by (2) in $\mathbb{R}^3$. Then the V-DFS vector g is defined as the concatenation of all 3D fractal dimensions, i.e., $[\beta(E_{\alpha_1}), \beta(E_{\alpha_2}), \ldots]$.

    S-DFS: Compute the vector of fractal dimensions for each 2D slice of the volume along the x, y and t axis by using (5) in $\mathbb{R}^2$. Then compute the mean vector of all vectors of the corresponding 2D slices for each axis. The S-DFS vector l is defined as the concatenation of these three mean vectors.
4. Concatenate V-DFS vector g and S-DFS vector l to form the final DFS vector d.

Volumetric DFS (*V-DFS*): We first consider a DT sequence as a single 3D volume data and assume that it to be generated by a certain dynamic process in the spatio-temporal domain $\mathbb{R}^3$ with 3D statistical self-similarities. In this case, the global self-similarities in the 3D volume are represented by a vector of fractal dimensions via the multi-fractal analysis in $\mathbb{R}^3$, and the resulting fractal dimension vector is defined as the V-DFS component in DFS. The procedure of computing V-DFS is as follows. Firstly, all pixels in the video are considered as the points in a 3D volume and are partitioned into many 3D point sets based on their local multi-scale behaviors characterized by (4) in $\mathbb{R}^3$ according to some measure. Secondly, the fractal dimension of each fractal point set is estimated by the least squares fitting in the log–log coordinate system. Lastly, the V-DFS is obtained by organizing the fractal dimensions of all the fractal point sets into a vector. See Fig. 2 (a) for an visual illustration of the procedure.

Multi-slice DFS (*S-DFS*): Aside from the global volumetric self-similarity characterized by V-DFS, the local spatial and temporal analysis provides more discriminative information regarding the fractal structures existing in DT sequences. Thus, we introduce one more component called S-DFS, which examines the

| Input | Intensity | Temporal gradient | Normal flow | Laplacian | Principle Curvature |
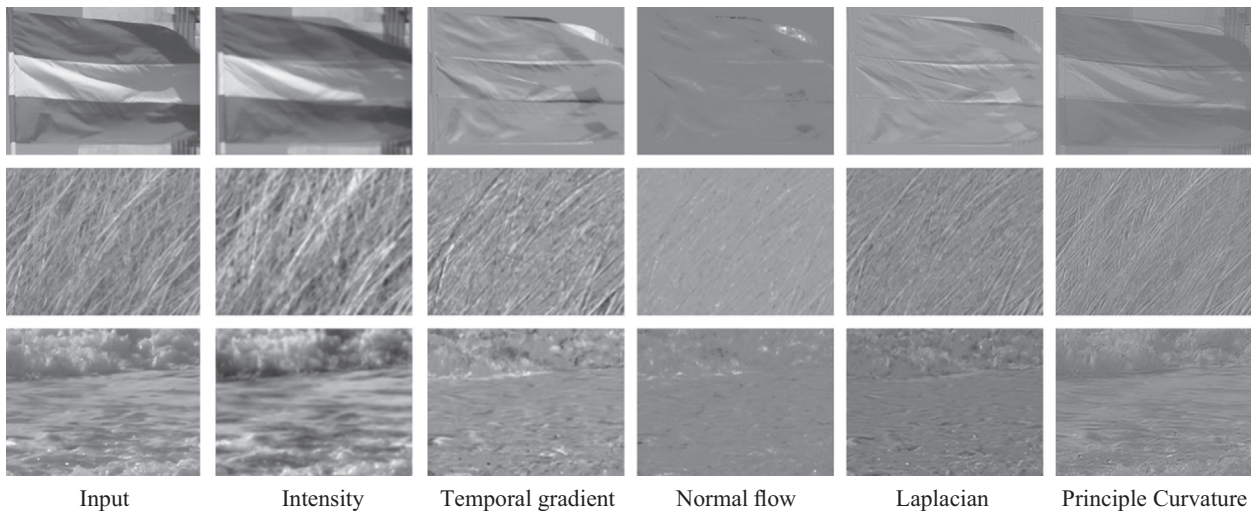
Fig. 1. Examples of the five measures used in the proposed method. The first column shows the frames extracted from three DT videos in DynTex [30] that transformed to gray scale. The second to sixth columns show the corresponding measures (7)–(11).

a

Divide fractal subsets   Estimate fractal dimension   Collect as vector

b

Three orthogonal planes   Divide fractal sets   Compute fractal dimension vectors   Concatenate as output

Fig. 2. Computation of V-DFS and S-DFS.

Fig. 3. Three 2D slices of sample sequences from DynTex [30]. The first three columns show three sample 2D slices of each sequence along three orthogonal axes. The last column shows the corresponding three fractal dimension vectors.
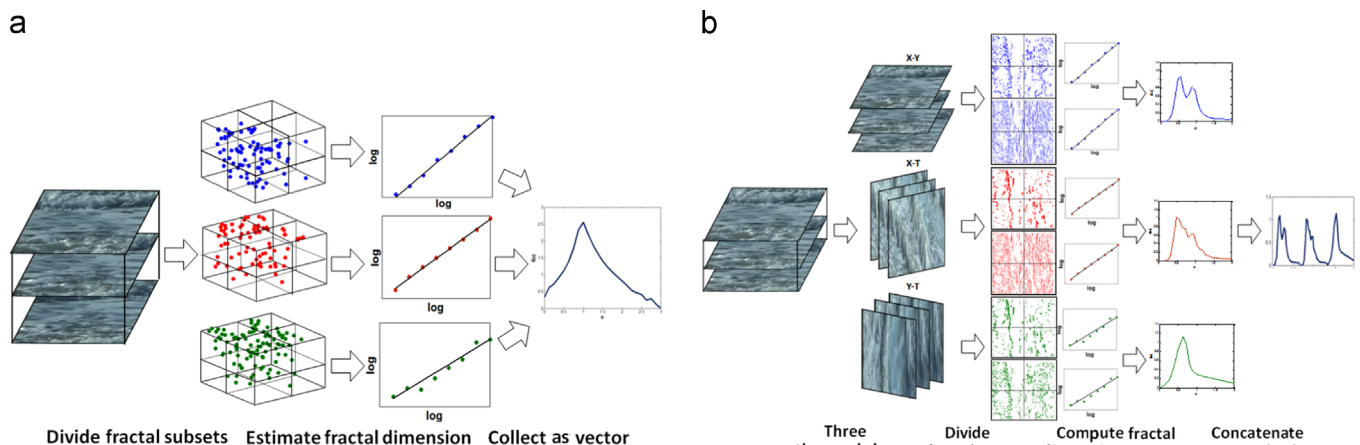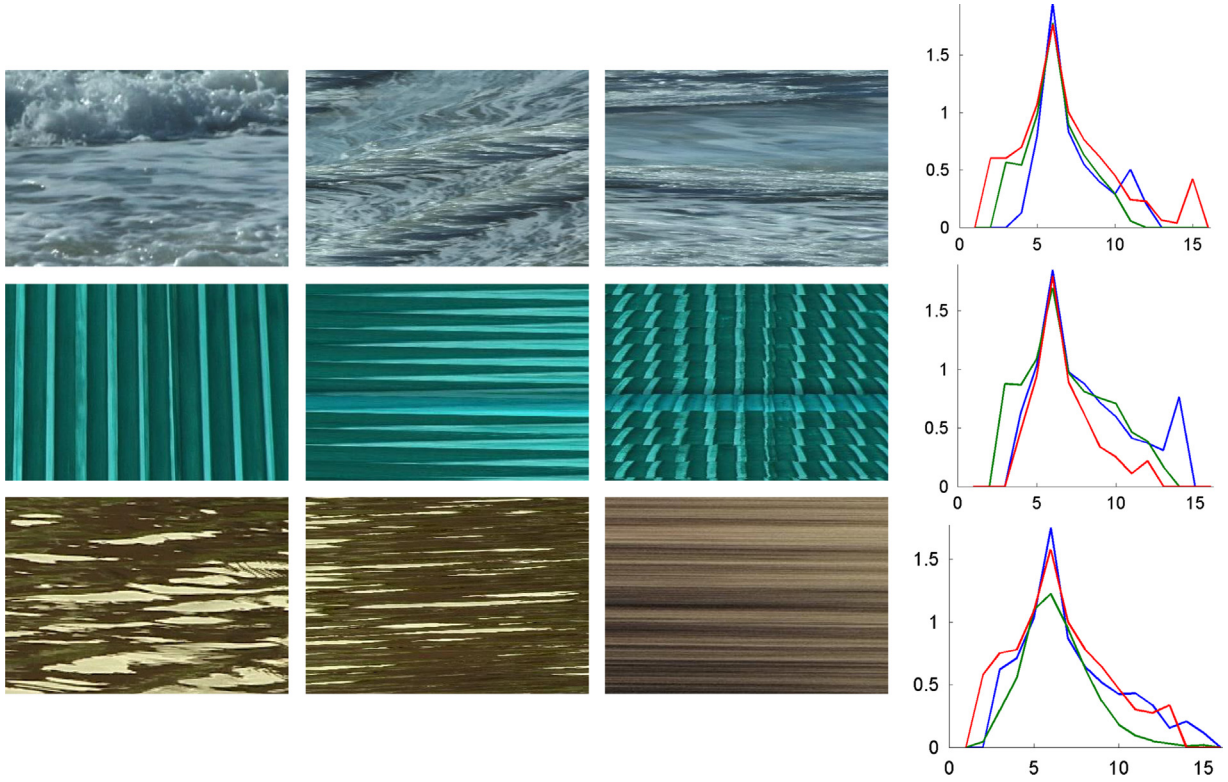
self-similarity behavior of 2D slices cut along three orthogonal axes in a DT volume. The detailed procedure of computing S-DFS is described as follows. Firstly, we compute a vector of fractal dimensions for each slice of the DT volume along the x, y and t axes, which is obtained by calculating the fractal dimensions of all 2D fractal point sets formed by partitioning all pixels on the slice based on their Höder exponents defined in (4). Then for each axis, the mean of the computed fractal dimension vectors is calculated over all slices along this axis. The reason of using the mean is to achieve stability, as we found that the slices along the same axes exhibit quite similar self-similarities. At last, the S-DFS vector is constructed by concatenating the three mean fractal dimension vectors with respect to two spatial axes and one temporal axis. See Fig. 2(b) for an visual illustration of the procedure. The slices along three axes and their corresponding fractal dimension vectors are shown in Fig. 3 for three DT samples. It is seen that strong fractal structures indeed exist in the 2D slices of DT sequences. Also, the slices from different axes exhibit different types of fractal structures, which implies that S-DFS does capture fractal structures of DT from different perspectives.

3.3. Implementation details

3.3.1. Acceleration via integral videos

Recall that the adopted five measures are defined by the summation of a special scalar function over many 3D cubes $B_{(p_0,t_0)}(r_s, r_t)$. Such computations are quite expensive when dealing with large-scale data. To alleviate this problem, the *integral video* technique [31,32] is used in our implementation to speed up the computation. Take the pixel intensity measure μ_I for instance. In discrete setting, it is easy to show that for a DT sequence $I(x,y,t)$, the intensity summation $\rho(B)$ on the cube grid $B_{(p_0,t_0)}(r_s, r_t)$ can be

expressed as follows[1]:

$$
\begin{aligned}
\rho(B) =\ & \widehat{I}(x_2,y_2,t_2) - \widehat{I}(x_1-1, y_2, t_2) \\
& - \widehat{I}(x_2, y_1-1, t_2) - \widehat{I}(x_2, y_2, t_1-1) \\
& + \widehat{I}(x_2, y_1-1, t_1-1) + \widehat{I}(x_1-1, y_2, t_1-1) \\
& + \widehat{I}(x_1-1, y_1-1, t_2) - \widehat{I}(x_1-1, y_1-1, t_1-1),
\end{aligned}
$$

where $\widehat{I}(x,y,z) = \sum_{x'=1}^{x} \sum_{y'=1}^{y} \sum_{t'=1}^{t} I(x', y', t')$ is defined as the integral video of I, $(x_1, y_1) = p_0 - r_s$, $(x_2, y_2) = p_0 + r_s$, $t_1 = t - r_t$ and $t_2 = t + r_t$. This formulation enables a constant-time computation of μ_I after one-pass computation of integral video. The same technique is also used in the computation of fractal dimension, as counting the nonempty boxes in the computation of DFS, i.e. computing $\#(E, \frac{1}{m})$ in (2), is equivalent to counting the 3D cubes or 2D rectangles with positive sums.

3.3.2. Enhancing robustness by soft assignment

In Step 3 of Algorithm 1, the pixels in a given DT sequence are partitioned into different sets $E_{[\alpha_i, \alpha_{i+1}]}$ according to the local density values α. In existing fractal-based methods [29,33,15], the partition is implemented by a "hard scheme" i.e., a pixel p is assigned to $E_{[\alpha_i, \alpha_{i+1}]}$ iff $\alpha(p) \in [\alpha_i, \alpha_{i+1})$, meaning $E_{\alpha_i} \cap E_{\alpha_{i+1}} = \varnothing$. Such a scheme is vulnerable to quantization errors, especially for the pixels with fractal dimension close to the end points of the interval, we take a "soft" assignment strategy. To overcome this weakness, we take a "soft" assignment strategy. Specifically, for a set $E_{[\alpha_i, \alpha_{i+1}]}$, its soft assignment function $m_i(p)$ is defined as

$$
m_i(p) = \begin{cases}
1 & \text{if } \alpha(p) \in [\alpha_i, \alpha_{i+1}) \\
\text{tansig}\left(\dfrac{|\alpha(p) - \alpha_i|}{\tau} \right) & \text{if } \alpha(p) \in A_{\alpha_i, \tau} \\
0 & \text{otherwise},
\end{cases} \tag{12}
$$

---

[1] Note that in discrete setting B is a cube grid with positive integer coordinates.

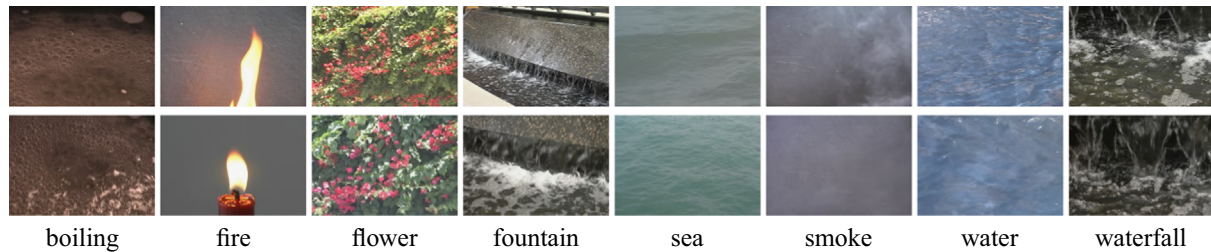boiling   fire   flower   fountain   sea   smoke   water   waterfall

Fig. 4. Example snapshots of eight classes used in our experiment from the UCLA dataset.

where $A_{\alpha_i,\tau} = [\alpha_i - \tau, \alpha_i) \cup [\alpha_{i+1}, \alpha_{i+1} + \tau)$ and τ is a predefined threshold. The soft alignment function (12) allows intersection between two point sets with adjacent Hölder exponent intervals. The threshold τ was fixed to be a small value in implementation. We empirically found that the soft assignment improves the robustness of the fractal dimension vector against quantization errors.

4. Experiments and discussion

In this section, we evaluated our DFS descriptor in terms of the DT classification accuracies on four public DT datasets, including the *UCLA dataset* [2], the *DynTex dataset* [16], the *DynTex++ dataset* [34] and the *DynTex New dataset* [35]. We reported the results in comparison with state-of-the-art DT classification approaches. Note that the DynTex, DynTex++, and DynTex New datasets are collected from the same pool of DT sequences but with different sizes and protocols. The DynTex dataset is the predecessor of the DynTex New dataset with the same creators, and the samples of DynTex++ dataset are collected and panned from the DynTex New dataset by another creator. All these datasets are challenging in different aspects and hence able to provide rich evaluation for the experiments.

We computed V-DFS and S-DFS using all five measures (7)–(11). For V-DFS, a 25-dimensional vector is computed on each measure, and the dimension of the resulting V-DFS vector is 125. For S-DFS, it generates a 75-dimensional vector (25 for each axis) for each measure, and the dimension of the resulting S-DFS vector is 375. The final DFS descriptor is the concatenation of the V-DFS and the S-DFS vectors, with a total dimension of 500. The parameters are set as the following: $r_t = 2$ for the pixel intensity measure and $r_t = 1$ for the other four measures on all the datasets, $r_s = 5$ for the UCLA and DynTex++ datasets, and $r_s = 6$ for the DynTex dataset.[2] We noted experimentally that the DFS descriptor is insensitive to small perturbations of these parameters. To analyze the contribution of each component in DFS, we also reported the results generated by using V-DFS and S-DFS individually. The contribution of each measure used for computing DFS is analyzed in Section 4.5.

4.1. Evaluation on the UCLA dataset

The UCLA DT dataset has been widely used for DT classification evaluation [9,2,18,34,36], which originally contains a total of 200 gray-scale sequences from 50 classes. Each class has 4 sample sequences captured from different viewpoints. Fig. 4 shows some samples from the dataset. There are the following different breakdowns for this dataset when it is used for evaluating DT classification approaches.

50 Class [18,34]: The classification on the original 50 classes was performed.

Table 1
The classification accuracies (%) on the UCLA dataset. *Note*: Superscripts "S", "N" and "M" are for results using SVM, 1NN, and maximum margin learning (followed by 1NN) [34] respectively; "–" means "not available".

Method	7-Class	8-Class	9-Class	50-Class	SIR
[36]	–	80.00^S	–	–	–
[18]	92.30^N	–	–	81.00^N	60.00^N
[34]	–	–	95.60^M	99.00^M	–
[14]	96.11^N	99.50^S	97.23^N	$99.25^S, 87.10^N$	67.45^N
[15]	96.87^N	96.96^S	97.11^S	$99.75^S, 99.12^N$	61.25^N
V-DFS	86.94^N	88.94^S	82.56^S	90.25^S	55.67^S
S-DFS	98.25^N	97.54^S	97.25^S	99.80^S	71.86^S
DFS	98.57^N	99.20^S	97.50^S	100^S	74.20^N

Shift-invariant recognition (*SIR*)*-class* [18]: To test the shift-invariance of descriptors, each of the original 200 video sequences is cut into non-overlapping left and right halves, and totally 400 sequences are obtained. The "shift-invariant recognition" was implemented to compare the sequences only between different halves to test the shift-invariance of descriptors. As a result, the dataset becomes very challenging.

9-Class [34]: By combining the sequences from different viewpoints, the 50 classes were merged to the 9 classes including boiling water (8), fire (8), flowers (12), fountains (20), plants (108), sea (12), smoke (4), water (12) and waterfall (16), where the numbers denote the number of the sequences in the each class. As a result, the dataset serves as an excellent test bed for evaluating DT classification algorithms under viewpoint changes.

8-Class [36]: The 9 classes used in [34] were further reduced to the 8 classes by removing sequences of "plant", as it contains too many sequences.

7-Class [18]: The "semantic category recognition" was considered on the 400 sequences obtained by cutting 200 video sequences into non-overlapping parts. These 400 sequences were represented into the following semantic categories: flames (16), fountain (8), smoke (8), (water) turbulence (40), (water) waves (24), waterfalls (64) and (windblown) vegetation (240).

Our proposed DFS descriptors were compared with the previously tested methods in [2,9,14,15,18,34,36] using the same experimental setups. The classification accuracies are shown in Table 1 and the confusion matrices are shown in Fig. 5. It is seen that our approach achieves the best performance in all cases except the '8-Class' configuration.

4.2. Evaluation on the DynTex dataset

The DynTex dataset [30] contains various kinds of DT videos, ranging from struggling flames to whelming waves, from sparse curling smoke to dense swaying branches. The sequences in DynTex are taken under different environmental conditions involving scaling and rotation. Each sequence is a color video with dimension 400×300 in space and 250 frames in 10 s, and de-interlaced with a spatio-temporal median filter.

---

[2] As the resolutions of the datasets are different, we assign r_s larger value when the resolution is relatively high.
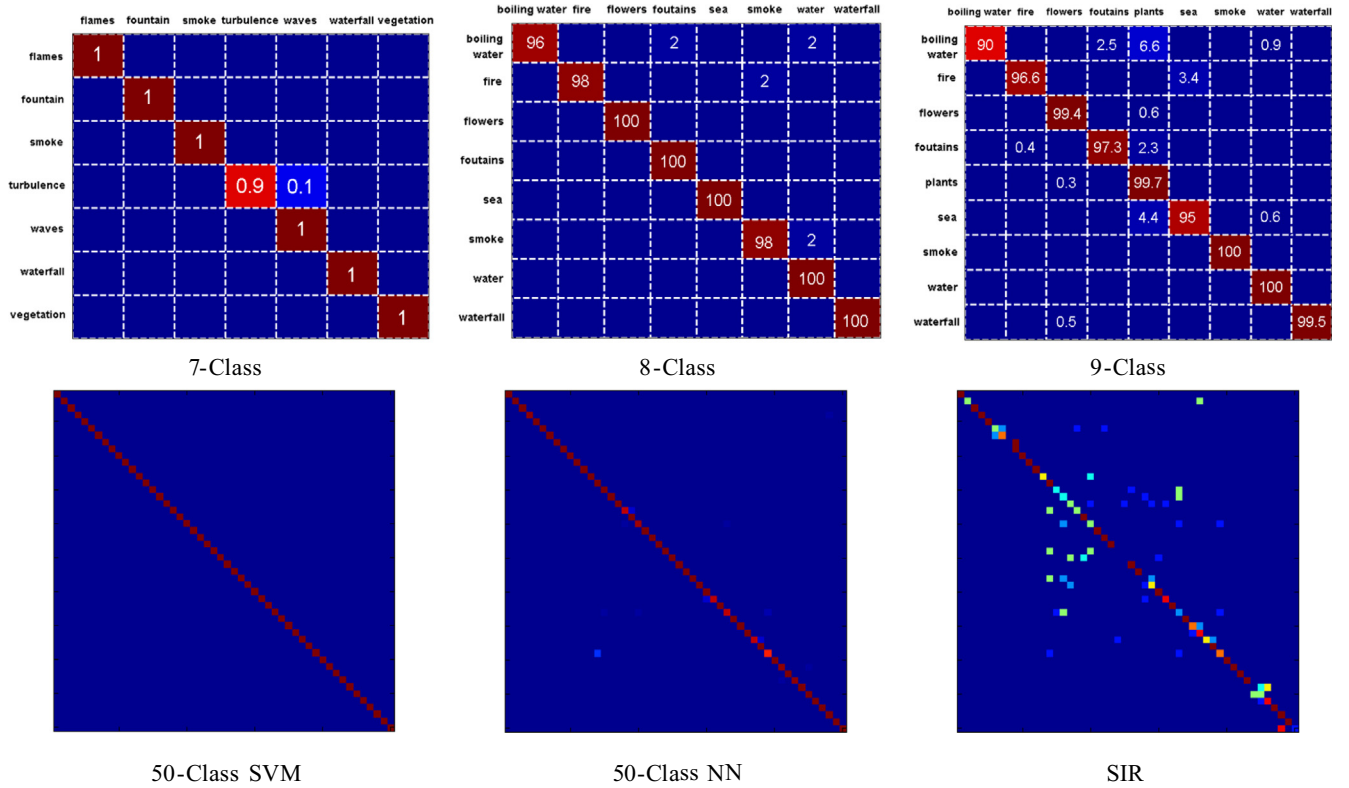
Fig. 5. Confusion matrices by DFS on the UCLA dataset.

Although the DynTex dataset has been used for DT classification evaluation in many studies [37,12,6], different experimental configurations (e.g. different subsets and categories) were usually used. For the purpose of fair comparison, we follow the work in [6] since it not only gives a detailed description of the configuration but also achieves very good results. The experimental settings is as follows. Firstly, a version of the DynTex dataset containing 35 DT categories is used. Then, each DT sequence is divided into eight non-overlapping subsequences with random meaningful sizes along all dimensions. In addition, two subsequences are generated from the original sequence by randomly cutting on the temporal axis. Consequently, each original sequence creates 10 sample subsequences with various dimensions. These samples share the same class label with the original sequence. Finally, all such samples are used in the DT classification task.

The evaluation is conducted using the leave-one-group-out scheme and the average performance over 2000 runs is reported. For each run, one sample per class is selected to form the testing set and the rest samples are used as the training set. Each class is then represented by the mean feature vector over the samples in the training set. After that, each test sample is classified according to the class that has the smallest ℓ_1 distance in the feature space. Finally the average classification accuracy over all runs is reported.

Similar to [6], we also tested different weights for each feature dimension to improve the performance. The results are shown in Table 2. It can be seen that our method performs very well, with recognition accuracies of 95.92% and 97.16% for non-weighting and best-weighting respectively. Both scores outperform the best results reported in [6,14]. The confusion matrix is shown in Fig. 6. It is worth mentioning that our descriptors require much fewer parameters than those in [6]. Only two simple parameters rather easy to be determined in practice are considered: the radius

Table 2
The classification accuracies (%) on the DynTex dataset.

Weighting	LBP–TOP [6]	3D–OTF [14]	V–DFS	S–DFS	DFS
Non-weighting	95.71	95.89	73.14	91.62	95.92
Best-weighting	97.14	96.70	74.68	93.38	97.16

r_s shared in (7)–(10) and the radius r_t in (7) for estimating the local density function.[3]

4.3. Evaluation on the DynTex++ dataset

The DynTex++ dataset proposed in [34] is a challenging dataset composed of 36 classes of dynamic texture, each of which contains 100 sequences of a fixed size $50 \times 50 \times 50$. The dataset is designed carefully to provide a rich and reasonable benchmark for DT recognition. We used the same experimental setting as that in [34] for evaluation. One half of samples from each class are used for training, and the other half are used for testing. A SVM classifier with the RBF kernel is trained for prediction. We applied our DFS descriptor on DynTex++ and obtained an average recognition accuracy of 91.70%, which significantly outperforms previously tested methods [14,34]. The classification accuracies for each class are shown in Fig. 7 and the confusion matrix is shown in Fig. 6.

4.4. Evaluation on the DynTex New dataset

With the development of DT analysis and DT classification, the performances on the original DynTex dataset become saturate.

---

[3] The number of scales used for calculating the box counting fractal dimension in (2) and the local density exponent (4) does not influence the performance of DFS within reasonable ranges.
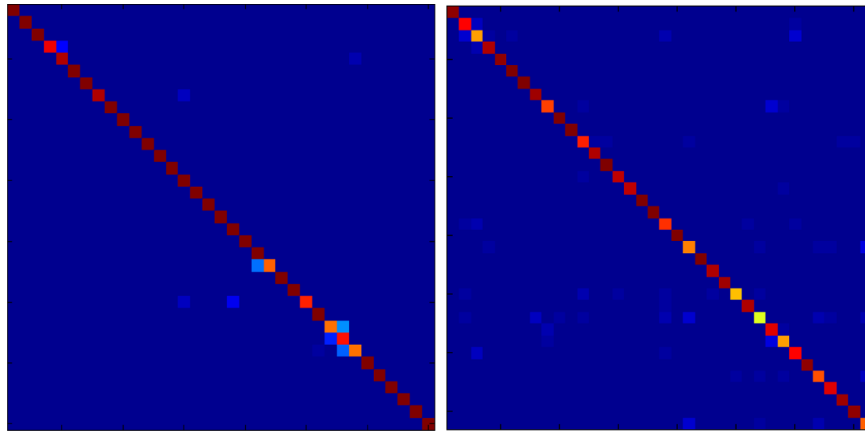
Fig. 6. Confusion matrices by DFS on the DynTex (left) and DynTex++ (right) datasets.

100	100	92.96	98.84	84.40	92.30	96.20	97.76	82.60
98.84	84.62	78.16	96.54	99.30	99.44	99.62	97.90	80.72
100	86.46	96.88	99.18	90.92	92.92	97.54	98.88	84.64
100	81.88	96.22	96.16	70.78	97.14	62.98	92.00	76.54

Fig. 7. The classification accuracy (%) by DFS on each class of the DynTex++ dataset.

Thus, the creators of the DynTex dataset rebuild a new version of DynTex dataset, denoted by DynTex New in this paper. The DynTex New dataset [35] inherits most of the samples from the original one and is enriched with many new samples and categories. It contains more than 650 video sequences that were captured under different conditions like scaling and rotation and captured by static cameras as well as moving ones. For classification, the DynTex New dataset is suggested to be divided and rearranged into three benchmark datasets, named the Alpha dataset, the Beta dataset, and the Gamma dataset:

- *Alpha*: The Alpha dataset is composed of 60 DT sequences divided into three classes, including sea, grass, and trees. Each category contains 20 sequences.
- *Beta*: The Beta dataset is composed of 162 DT sequences divided into 10 classes, including sea, vegetation, trees, flags,

Table 3
The classification accuracy (%) on the DynTex++ dataset.

Dataset	DL-PEGASOS [34]	3D-OTF [14]	V-DFS	S-DFS	DFS
DynTex++	63.70	89.17	68.22	88.71	91.70

Table 4
The classification accuracies (%) by DFS on the DynTex New dataset.

Dataset	LBP-TOP [6]	OTF [14]	WMFS [15]	V-DFS	S-DFS	DFS
Alpha	83.34	83.61	84.83	67.34	83.62	85.24
Beta	73.44	73.22	75.21	59.75	4.14	76.93
Gamma	72.03	72.53	73.32	58.62	72.12	74.82

Table 5
The classification accuracies (%) by different components of DFS on several test dataset.

Dataset	V-DFS					S-DFS					DFS				
	μ_I	μ_B	μ_N	μ_L	μ_C	μ_I	μ_B	μ_N	μ_L	μ_C	μ_I	μ_B	μ_N	μ_L	μ_C
UCLA-SIR	48.7	50.7	49.9	42.1	45.6	61.8	64.2	63.3	53.7	57.9	65.1	67.7	66.7	56.8	60.9
DynTex	67.6	70.5	69.3	58.4	63.4	83.7	87.8	85.8	72.3	78.3	89.1	92.6	91.7	77.1	83.3
DynTex++	58.4	60.7	59.8	50.1	52.4	74.1	77.2	75.4	63.5	69.1	75.1	78.1	76.9	64.8	70.2
DynTex Alpha	55.9	55.7	54.6	55.5	59.6	76.2	76.9	73.8	75.2	80.9	78.9	78.9	76.4	77.8	83.0
DynTex Beta	48.9	51.6	45.9	41.3	44.2	62.9	71.1	61.4	57.1	59.5	65.5	74.1	64.0	59.5	62.0
DynTex Gamma	44.7	45.8	41.2	41.4	42.8	60.8	62.4	55.8	54.8	57.8	63.6	65.4	59.4	58.3	61.5

clam water, fountains, smoke, escalator, traffic and rotation. The number of video sequences per category varies from 7 to 20.

- *Gamma*: The Gamma dataset is composed of 275 DT sequences divided into 10 classes, including flowers, sea, naked trees, foliage, escalator, calm water, flags, grass, traffic and fountains. The number of video sequences per category varies from 7 to 38.

As there is no previous experimental result reported on these three benchmarks, we initiate an experimental protocol as follows. A SVM with the RBF kernel is trained on 5 randomly selected samples per category. The remaining samples are used for testing. The average performance over 100 runs is reported.

We compared our method with [6,14,15]. The parameters of these methods are either set according to suggestions from the original work or carefully tuned up for optimal results. The classification accuracies of the compared methods are summarized in Table 4. It can be seen that our method again performs the best.

4.5. Analysis and discussion on the components of DFS

From Tables 1 to 4 we can see that the S-DFS descriptor performs consistently better than the V-DFS descriptor across all the datasets. This does not surprise us as the slice-wise DT characteristics captured by S-DFS is richer and more informative than the global DT behaviors characterized by V-DFS in a 3D volume. Though V-DFS is less discriminative than S-DFS, from Tables 1 to 4 it can also be seen that V-DFS is complementary to S-DFS in DT classification, as DFS as the combination of V-DFS and S-DFS has noticeable improvement over single S-DFS. This demonstrated that although the self-similarities on the entire DT volume and those on the DT slices are related to each other, they provide different cues which can be integrated for improved discriminative power for DT classification.

To analyze the contribution of each measure used in DFS, we showed in Table 5 the classification results generated by using individual measure in computing the V-DFS, S-DFS, and DFS descriptors on the test datasets. It is seen from Table 5 that temporal brightness gradient measure is the most discriminative among all the five measures through all the test datasets, and all the measures are complementary to each other.

5. Conclusion

We presented a powerful DT descriptor using dynamic fractal analysis developed in this paper. The proposed DFS descriptor consists of two components which capture the 3D fractal structures in DT sequences from different perspectives. Based on the developed five spatio-temporal measures of DT pixels, the DFS descriptor effectively captures the stochastic self-similarities existing in a wide range of DT sequences regarding different local DT features. The speeded-up and robust implementation of DFS

has also been presented. Experiments on four benchmark datasets showed noticeable performance improvement of the proposed method over the state-of-the-art methods in DT classification.

There are several limitations of the proposed method. Firstly, real scenes often contain DTs of multiple types (e.g. hills behind lakes), but the proposed method does not work well in such cases without pre-segmentation. Secondly, the robustness to cluttering and occlusion of DFS is not guaranteed. Such robustness is crucial in recognizing the scenes composed of dynamic textures and non-textures. Finally, the dimension of the DFS descriptor used in the experiment is relatively large in comparison with many existing LBP-based methods, which limits the practicability of the proposed method.

In the future, we would like to study new fractal-based methods to overcome the aforementioned drawbacks. In particular, we would like to utilize DFS as a local descriptor for DT segmentation and dynamic scene classification.

Conflict of interest

None declared.

References

[1] Dmitry Chetverikov, Renaud Peteri, A brief survey of dynamic texture description and recognition, in: Computer Recognition Systems, Springer, Berlin, Germany, 2005, pp. 17–26.
[2] Gianfranco Doretto, Alessandro Chiuso, Ying Nian Wu, Stefano Soatto, Dynamic textures, Int. J. Comput. Vis. 51 (2) (2003) 91–109.
[3] Gianfranco Doretto, Daniel Cremers, Paolo Favaro, Stefano Soatto, Dynamic texture segmentation, in: IEEE International Conference on Computer Vision, IEEE, Nice, France, 2003, pp. 1236–1242.
[4] Bernard S. Ghanem, Dynamic textures: models and applications (Ph.D. thesis), University of Illinois at Urbana-Champaign, 2010.
[5] John R. Smith, Ching-Yung Lin, Milind Naphade, Video texture indexing using spatio-temporal wavelets, in: IEEE Conference on International Conference on Image Processing, vol. 2, IEEE, Rochester, New York, USA, 2002, pp. II–437.
[6] Guoying Zhao, Matti 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.
[7] Martin Szummer, Rosalind W. Picard, Temporal texture modeling, in: International Conference on Image Processing, vol. 3, IEEE, Lausanne, Switzerland, 1996, pp. 823–826.
[8] Gianfranco Doretto, Eagle Jones, Stefano Soatto, Spatially homogeneous dynamic textures, in: European Conference on Computer Vision, Springer, Prague, Czech Republic, 2004, pp. 591–602.
[9] Payam Saisan, Gianfranco Doretto, Ying Nian Wu, Stefano Soatto, Dynamic texture recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, IEEE, Kauai, HI, USA, 2001, pp. II–58.
[10] Franco Woolfe, Andrew Fitzgibbon, Shift-invariant dynamic texture recognition, in: European Conference on Computer Vision, Springer, Graz, Austria, 2006, pp. 549–562.
[11] Antoni B. Chan, Nuno Vasconcelos, Classifying video with kernel dynamic textures, Comput. Vis. Pattern Recognit. (2007) 1–6.
[12] Bernard Ghanem, Narendra Ahuja, Phase based modelling of dynamic textures, in: International Conference on Computer Vision, IEEE, Rio de Janeiro, Brazil, 2007, pp. 1–8.

[13] Richard P. Wildes, James R. Bergen, Qualitative spatiotemporal analysis using an oriented energy representation. in: European Conference on Computer Vision, Springer, Dublin, Ireland, 2000, pp. 768–784.

[14] Yong Xu, Sibin Huang, Hui Ji, Cornelia Fermüller, Scale-space texture description on sift-like textons, Comput. Vis. Image Underst. 116 (9) (2012) 999–1013.

[15] Hui Ji, Xiong Yang, Haibin Ling, Yong Xu, Wavelet domain multifractal analysis for static and dynamic texture classification, IEEE Trans. Image Process. 22 (1) (2013) 286–299.

[16] Renaud Péteri, Dmitry Chetverikov. Dynamic Texture Recognition Using Normal Flow and Texture Regularity, 2005, pp. 223–230.

[17] Ramprasad Polana, Randal Nelson, Temporal texture and activity recognition, Springer, Berlin, Germany, 1997.

[18] Konstantinos G. Derpanis, Richard P. Wildes, Dynamic texture recognition based on distributions of spacetime oriented structure, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, San Francisco, CA, USA, 2010, pp. 191–198.

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

[20] Konstantinos G. Derpanis, Matthieu Lecce, Kostas Daniilidis, Richard P. Wildes, Dynamic scene understanding: the role of orientation features in space and time in scene classification, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Providence, RI, USA, 2012, pp. 1306–1313.

[21] J.H. van Hateren, Processing of natural time series of intensities by the visual system of the blowfly, Vis. Res. 37 (23) (1997) 3407–3416.

[22] Vincent A. Billock, Gonzalo C. de Guzman, J.A. Scott Kelso, Fractal time and 1/f spectra in dynamic images and human vision, Phys. D: Nonlinear Phenom. 148 (1) (2001) 136–146.

[23] Dawei W. Dong, Joseph J. Atick, Statistics of natural time-varying images, Network: Comput. Neural Syst. 6 (3) (1995) 345–358.

[24] Yuhui Quan, Yong Xu, Yuping Sun, A distinct and compact texture descriptor, Image Vis. Comput. 32 (4) (2014) 250–259.

[25] Yuhui Quan, Yong Xu, Yuping Sun, Yu Luo, Lacunarity analysis on image patterns for texture classification, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Columbus, OH, USA, 2014, pp. 160–167.

[26] Yong Xu, Yuhui Quan, Haibin Ling, Hui Ji, Dynamic texture classification using dynamic fractal analysis, in: International Conference on Computer Vision, IEEE, Barcelona, Spain, 2011, pp. 1219–1226.

[27] J. Kenneth, Falconer, K.J. Falconer, Techniques in Fractal Geometry, vol. 16, Wiley Chichester, West Sussex, UK, 1997.

[28] Benoit B. Mandelbrot, The Fractal Geometry of Nature/Revised and Enlarged Edition, New York, WH Freeman and Co., New York, USA, 1983, 495 p. 1.

[29] Yong Xu, Hui Ji, Cornelia Fermüller, Viewpoint invariant texture description using fractal analysis, Int. J. Comput. Vis. 83 (1) (2009) 85–100.

[30] Renaud Péteri, Sandor Fazekas, Mark J. Huiskes, Dyntex: a comprehensive database of dynamic textures, Pattern Recognit. Lett. 31 (12) (2010) 1627–1632.

[31] Paul Viola, Michael J Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2) (2004) 137–154.

[32] Yan Ke, Rahul Sukthankar, Martial Hebert, Efficient visual event detection using volumetric features, in: IEEE International Conference on Computer Vision, vol. 1, IEEE, Beijing, China, 2005, pp. 166–173.

[33] Yong Xu, Xiong Yang, Haibin Ling, Hui Ji, A new texture descriptor using multifractal analysis in multi-orientation wavelet pyramid, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, San Francisco, CA, USA, 2010, pp. 161–168.

[34] Bernard Ghanem, Narendra Ahuja, Maximum margin distance learning for dynamic texture recognition, in: European Conference on Computer Vision, Springer, Crete, Greece, 2010, pp. 223–236.

[35] Renaud Péteri, Sándor Fazekas, Mark J. Huiskes. DynTex: a comprehensive database of dynamic textures, Pattern Recognit. Lett., http://dx.doi.org/10.1016/j.patrec.2010.05.009. 〈http://projects.cwi.nl/dyntex/〉.

[36] Avinash Ravichandran, Rizwan Chaudhry, René Vidal, View-invariant dynamic texture recognition using a bag of dynamical systems, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Miami, FL, USA, 2009, pp. 1651–1657.

[37] Sándor Fazekas, Dmitry Chetverikov, Normal versus complete flow in dynamic texture recognition: a comparative study, in: International Workshop on Texture Analysis and Synthesis, 2005, pp. 37–42.

Yong Xu (M'96–SM'99) received the B.S., M.S. and Ph.D. degrees in Mathematics from Nanjing University, China, in 1993, 1996 and 1999 respectively. From 1999 to 2001, he was a postdoc research fellow in computer science at South China University of Technology and became a faculty member afterward. He is currently a professor with the School of Computer Science & Engineering at South China University of Technology, China. His research interests include image analysis, image/video recognition and image quality assessment. He is the member of IEEE Computer Society and of ACM.

Yuhui Quan received the B.E. degree and the Ph.D. degree in computer science from South China University of Technology in 2013. He is currently the postdoc research fellow in mathematics at National University of Singapore. His research interests include computational vision, sparse representation, and machine learning.

Zhuming Zhang is currently a Ph.D. student in the Department of Computer Science & Engineering at The Chinese University of Hong Kong. Before that, he received the B.E. degree from the Guangdong University of Technology, China, in 2010 and the M.E. degree from the South China University of Technology, China, in 2013. His research interests include image analysis, image/video recognition, and computer graphics.

Haibin Ling received the B.S. degree in mathematics and the M.S. degree in computer science from Peking University, China, in 1997 and 2000, respectively, and the Ph.D. degree from the University of Maryland, College Park, in Computer Science in 2006. From 2000 to 2001, he was an assistant researcher at Microsoft Research Asia. From 2006 to 2007, he worked as a postdoctoral scientist at the University of California Los Angeles. After that, he joined Siemens Corporate Research as a research scientist. In 2008, he joined Temple University as an assistant professor. Currently, he is an associate professor at Temple University. Dr. Ling's research interests include computer vision, medical image analysis, human computer interaction, and machine learning.

Hui Ji received the B.Sc. degree in Mathematics from Nanjing University in China, the M.Sc. degree in Mathematics from National University of Singapore and the Ph.D. degree in Computer Science from the University of Maryland, College Park. In 2006, the joined National University of Singapore as an assistant professor in Mathematics. Currently, he is an associate professor in mathematics at National University of Singapore. His research interests include computational harmonic analysis, optimization, computational vision, image processing and biological imaging.