

Dynamic Texture Classification Using Dynamic Fractal Analysis *

Yong Xu¹, Yuhui Quan¹, Haibin Ling² and Hui Ji³

¹School of Computer Science & Engineering, South China Univ. of Tech., Guangzhou 510006, China

²Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, U.S.A.

³Department of Mathematics, National University of Singapore, Singapore 117542

{yxu@scut.edu.cn, yuhui.quan@mail.scut.edu.cn, hbling@temple.edu, matjh@nus.edu.sg}

Abstract

In this paper, we developed a novel tool called dynamic fractal analysis for dynamic texture (DT) classification, which not only provides a rich description of DT but also has strong robustness to environmental changes. The resulting dynamic fractal spectrum (DFS) for DT sequences consists of two components: One is the volumetric dynamic fractal spectrum component (V-DFS) that captures the stochastic self-similarities of DT sequences as 3D volume datasets; the other is the multi-slice dynamic fractal spectrum component (S-DFS) that encodes fractal structures of DT sequences on 2D slices along different views of the 3D volume. Various types of measures of DT sequences are collected in our approach to analyze DT sequences from different perspectives. The experimental evaluation is conducted on three widely used benchmark datasets. In all the experiments, our method demonstrated excellent performance in comparison with state-of-the-art approaches.

1. Introduction

Dynamic textures (DT) are video sequences of moving scenes that exhibit certain stationary properties in time domain ([3, 6]). Such video sequences are pervasive in real world, like sequences of rivers, water, foliage, smoke, clouds, fire, swarm of birds, humans in crowds and etc. The applications concerning such video sequences are plenty, including surveillance, detection of the onset of emergencies, and foreground and background separation (e.g. [7, 13, 23]). In recent years, the study of dynamic textures has started to attract attention of the computer vision community, with related topics ranging from DT modeling and

synthesis to recognition and classification. In this paper, we focus on the development of effective classification techniques for dynamic textures.

Different from static textures, dynamic textures not only vary on the spatial distribution of texture elements, but also vary on their organization and dynamics over time. One main challenge in the study of dynamic textures is how to reliably capture the motion behavior of texture elements, i.e., the properties of dynamics of texture elements over time. Existing approaches model the dynamics either by treating videos as samples of stochastic dynamical systems or by directly measuring the motion field of videos. These approaches work well for dynamic textures with regular motion. 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.

An interesting observation regarding the motion patterns of many DT sequences is: Although their motion patterns could be highly irregular with certain chaos, 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 usually referred as *power law* or *fractal structure* ([16]). The existence of fractal structures in a large spectrum of dynamic nature images has been observed by many researchers. For example, it is shown in [1, 5, 14] that the amplitude of temporal frequency spectra $A(f)$ of many video sequences, including camera movements, weather and biological movements by one or more humans, indeed fits $1/f^\beta$ power-law models:

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

where f denote the frequency.

Motivated by the existence of stochastic self-similarities in a wide range of dynamic textures, we propose to model dynamic textures by using dynamic systems with self-similarities, i.e., dynamic textures are likely to be generated by some mechanism with similar stochastic behavior

*Y. Xu was partially supported by Program for New Century Excellent Talents in University (NCET-10-0368), the Fundamental Research Funds for the Central Universities (SCUT 2009ZZ0052) and National Nature Science Foundations of China 60603022 and 61070091. H. Ling was supported in part by NSF Grants IIS-0916624 and IIS-1049032. H. Ji was partially supported by AcRF Tier 1 R-146-000-126-112.

operating at various spatial and temporal scales. Based on fractal geometry theory, here we introduce a novel method called dynamic fractal analysis that provides rich discriminative information of such self-similarities of the underlying system. 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 proposed DFS descriptor is very robust to environmental changes such as cluttering, occlusions and view changes.

1.1. Previous work

Most DT recognition and classification methods can be roughly categorized as either generative or discriminative methods. The generative methods [2, 8, 12, 22, 24, 28] 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. For example, in [24], each pixel is expressed as a linear combination of the neighboring pixels in the spatio-temporal domain. A linear dynamic system (LDS) is used in [22] to model DT processes and DT recognition is done through an analysis on the resulting Stiefel manifold. The features proposed in [8] are based on the parameters of a stationary multiscale autoregressive system. A different distance measure is presented in [28] for comparing LDSs to achieve shift invariance. [2] brought a non-linear model of DT by using the kernel principal component analysis. [12] introduced a phase-based DT model for several DT-related tasks.

Alternatively, discriminative methods [21, 27, 31] have been proposed for DT classification without explicitly modeling the underlying dynamic system. In [27] spatiotemporal filters are constructed and specifically tuned up for certain local DT structures with a few image patterns and motion patterns. The descriptor proposed in [31] extends the local binary pattern (LBP) of 2D image to the 3D spatial-temporal volume. [21] combined local LDS model with the bag-of-words model. Compared to generative methods, discriminative methods tend to perform better in the task of DT classification, as shown in experiments of some recent works. The main advantage of discriminative methods lies in their robustness to environmental changes and view changes. However, the merits of existing discriminative methods are quite limited for DT with complex motion, as they are not capable of reliably capturing inherent stochastic stationary properties of such video sequences.

In addition to the approaches mentioned above, some methods are proposed to directly use the motion field information for DT classification. The flow-based method proposed by [3] is to convert the analysis on DT sequences to the analysis on sequences of the static information by using the instantaneous motion patterns estimated from se-

quences. In [18] and [20], DT analysis is done using statistical measurements on optical flow information of DT sequences. A metric of video sequences is defined in [15] using the velocity and acceleration fields estimated at various spatio-temporal scales. Some methods rely on the information extracted from certain transforms such as wavelet and 3D-surfacelet, e.g. [23] used spatio-temporal wavelet transformations to decompose DT into different spatio-temporal scales and measure outputs of each wavelet sub-band.

1.2. Our approach

The approach we proposed can be viewed as a discriminative method with generative motivation, as we assume DT sequences are generated by some non-linear stochastic dynamic systems with certain inherent multi-scale self-similarities as shown in previous studies [1, 5, 14]. Fractal geometry theory [16] is known to be a powerful tool to robustly capture such similarities from local features. Motivated by these observations, we developed a discriminative method called dynamic fractal analysis to measure stochastic self-similarities of DT using local features. As a result, the proposed dynamic fractal analysis actually has the merits of both categories of approaches: The discriminative power of generative methods for modeling stochastic behavior of DT and the robustness of discriminative methods to environmental changes.

Dynamic fractal analysis is built on the concept of the so-called *fractal dimension*, which measures the statistical self-similarity of a point set in a multi-scale fashion. The basic idea is to partition the pixels of all frames into different sets based on their local multi-scale behaviors under some measures, such as intensity or normal flow. Then the stochastic behavior of each pixel set is measured by its fractal dimension from different perspectives. Detailed explanation of dynamic fractal analysis is given in the following sections.

It is noted that fractal dimension and fractal analysis have been proposed in the literatures for static texture analysis. For example, fractal dimension was first proposed by Pentland [17] for texture analysis, and later on the similar concept is applied on static texture classification by replacing fractal dimension using more advanced multi-fractal analysis [25, 29, 30].

2. Basics on fractal analysis

In this section, we give a brief review on the theory of fractal analysis and its numerical implementation. Interested readers are referred to [9, 16, 29] for more details. Fractal analysis is built on the concept of *fractal dimension* which was first proposed by Mandelbrot [16] as the measurement of power law existing in many natural phenomena. 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 [9]:

$$\beta(E) = \lim_{m \rightarrow \infty} \frac{\log \#(E, \frac{1}{m})}{-\log \frac{1}{m}}, \quad (1)$$

where $\#(E, \frac{1}{m})$ is the number of mesh hypercubes that intersect E for $m = 1, 2, \dots$. 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-similar behaviors of the physical dynamic systems. The extension is done as follows. Instead of assuming all points generated by the same mechanism, a measure function μ is first defined such that μ obeys the local power law in terms of scale:

$$\mu(B_r(x)) \propto r^{\alpha(x)}, \quad (2)$$

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 of the measurement μ . The $\alpha(x)$ can be estimated by the local density function [9]:

$$\alpha(x) = \lim_{r \rightarrow 0} \frac{\log \mu(B(x, r))}{\log r}. \quad (3)$$

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 $r_0 > r_1 > \dots > r_z$.

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

$$f(\hat{\alpha}) = \beta(E_{\hat{\alpha}}), \quad (4)$$

where $E_{\hat{\alpha}} = \{x : \alpha(x) = \hat{\alpha}\}$ is the point set with 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 function μ .

3. Dynamic fractal analysis for DT

Based on fractal analysis, we developed the so-called dynamic fractal analysis for DT and derived a descriptor called dynamic fractal spectrum (DFS) which encodes strong discriminative information regarding multi-scale self-similarities existing in DT.

3.1. Spatio-temporal measurement of pixels

It is seen from (2) and (3) that fractal analysis is conducted on the measurement function μ which determines how pixels are categorized. An accepted measurement should partition pixels into different categories based on the intrinsic physical meaning of pixels and the resulting pixel partition should be robust to environmental changes. In our dynamic fractal analysis, the following four measures are chosen to examine DT from different perspectives.

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

$$\mu_I(p_0, t_0) = \iint_{B(p_0, t_0, r_s, r_t)} I(p, t) dp dt, \quad (5)$$

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 . The measure μ_I measures the overall intensity in a space-time neighborhood of the point (p_0, t_0) .

Temporal brightness gradient. Besides the spatial measurement, the temporal measure also plays an essential role when describing DT. Thus, the second measure used in our method is the *temporal brightness gradient*:

$$\mu_B(p_0, t_0) = \int_{B(p_0, t_0, r_s)} \frac{\partial I(p, t)}{\partial t} dp, \quad (6)$$

where $B(p_0, t_0, r_s)$ is the spatial square centering at (p_0, t_0) with spatial radius r_s (same as in μ_I). 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*:

$$\mu_F(p_0, t_0) = \int_{B(p_0, t_0, r_s)} \frac{\partial I(p, t) / \partial t}{\|\nabla I(p)\|} dp. \quad (7)$$

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 measurement 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 optical flow field for chaotic motions.

Laplacian. The last measure we adopted in our dynamic fractal analysis is the *Laplacian*:

$$\mu_L(p_0, t_0) = \int_{B(p_0, t_0, r_s)} \Delta I(p, t) dp, \quad (8)$$

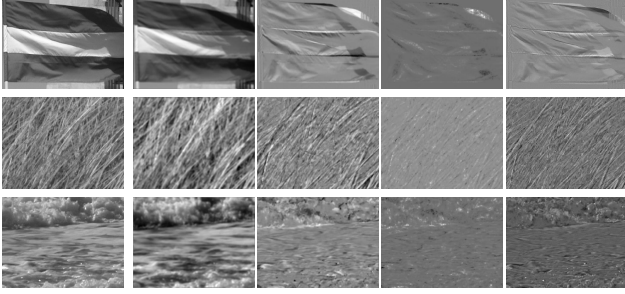


Figure 1. Examples of four types of measures. The first column shows the frames extracted from three DT videos in DynTex [19] that transformed to gray scale. The second to fifth columns show the corresponding measures (5) – (8).

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

The four measures quantify the local information of the pixel in the spatio-temporal domain from different perspectives, which leads to different pixel categorizations with different underlying physical implications. Both the intensity measure μ_I and the temporal gradient measure μ_B directly measure the 3D volume data from the spatio-temporal point of view. The measure μ_I encodes the brightness information and μ_B encodes the changes of brightness over the time. The normal flow measure μ_F is a known quantity in vision society that encodes reliable temporal changes of edge points. The Laplacian measure μ_L encompasses the second-order derivative information of brightness in the spatio-temporal domain. See Fig. 1 for the illustrations of these measures.

3.2. Dynamic fractal spectrum

After defining the four spatio-temporal measurements, μ_I , μ_B , μ_F and μ_L , we are ready to formulate the descriptor for DT using dynamic fractal analysis, which is called DFS (dynamic fractal spectrum). There are two components in DFS: 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 the distribution of the repetitive patterns in 2D slices of the 3D volume along three orthogonal axes. The proposed method is outlined in Algorithm 1.

Volumetric DFS (V-DFS). A DT sequence can be viewed as a 3D volume dataset and its self-similarity in the 3D volume can then be measured by the vector of fractal dimensions in \mathbb{R}^3 . In other words, DT is viewed as the volume data generated by some dynamic process in the spatio-temporal domain \mathbb{R}^3 with 3D statistical self-similarities, and the self-similarities are characterized by the multi-fractal analysis in \mathbb{R}^3 , denoted by V-DFS. The procedure

Algorithm 1 Dynamic fractal analysis (DFS)

Input: A sequence of image frames I

Output: DFS vector d

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

V-DFS: Classify each pixel x in the sequence 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 in the whole 3D spatio-temporal domain by (1) in \mathbb{R}^3 . Then the V-DFS vector g is defined as the concatenation of all 3D fractal dimensions.

S-DFS: Compute the vector of fractal dimensions for each 2D slice of the volume along the x , y and t axis by using (4) 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 .
-

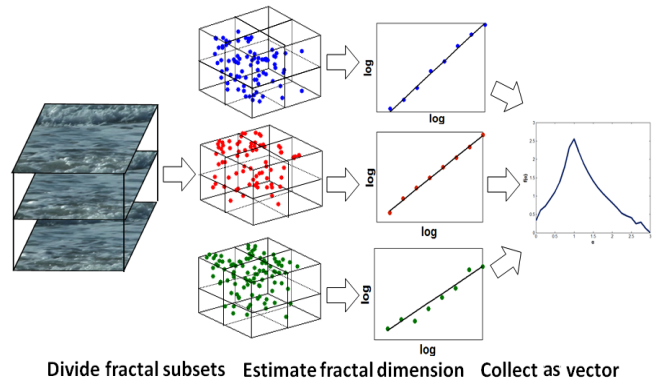


Figure 2. Computation of V-DFS (details in §3.2).

of computing V-DFS is as follows. Firstly, all pixels in the video are considered as points in the 3D volume and are partitioned into many 3D point sets based on their local multi-scale behaviors characterized by (3) in \mathbb{R}^3 . 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 fractal point sets into a vector. See Fig. 2 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 infor-

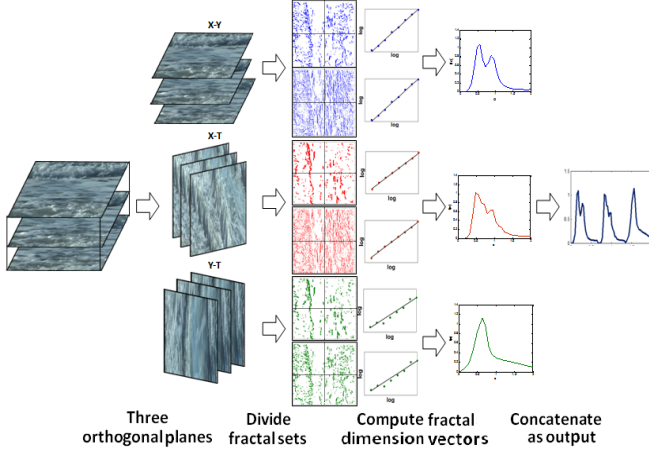


Figure 3. Computation of S-DFS (details in §3.2).

mation regarding the fractal structures existing in DT sequences. Thus, we introduce one more component called S-DFS, which examines the 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 the 2D multi-fractal vectors for all slices along x , y and t axes. For each slice, a vector is obtained by calculating the fractal dimensions of all 2D fractal point sets, which are formed by partitioning all pixels on this slice based on their Hölder exponents. Then for each axis, the mean of the 2D fractal dimension vectors is calculated over all slices along this axis. The reason of using the mean is to achieve the stability. At last, the S-DFS vector is defined by concatenating the three mean fractal dimension vectors with respect to two spatial axes and one temporal axis. See Fig. 3 for an visual illustration of the procedure. The volume slices of three axes and their corresponding fractal dimension vectors are shown in Fig. 4 for three sample DT sequences. 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. The complete S-DFS of four sample DT videos are shown in Fig. 5. It is seen that by using different measures for pixel categorization, the resulting S-DFS is also different.

3.3. Implementation details

Integral images. Recall that all four measurements are defined by the summation of a special scalar function μ over many 3D cubes $B(p_0, t_0, r_s, r_t)$ or 2D rectangles $B(p_0, t_0, r_s)$. Such computations can be costly. In our implementation, the *integral image* technique [26] is used to speed up the computation. The same technique is also used in the computation of fractal dimension, as counting the nonempty box is equivalent to counting the rectangles

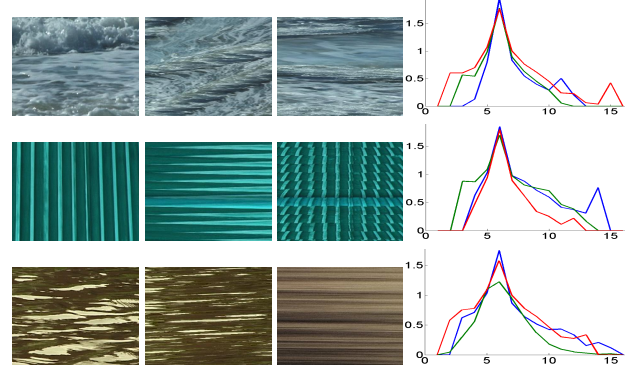


Figure 4. Three 2D slices of sample sequences from DynTex [19]. The first three columns show three sample 2D slices of each sequence along three orthogonal axes. The last column shows the corresponding three fractal dimension vectors.

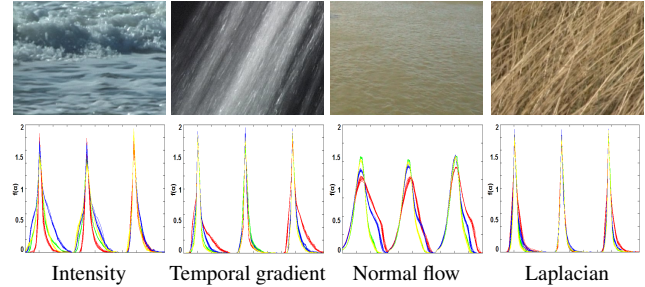


Figure 5. S-DFS of sample sequences. The first row shows one key frame of each video. The second row shows the S-DFS vectors by four types of measures, where the plots in blue, yellow, red and green represent the results of sea-weaves, shower-strong, danube-far and straw respectively.

or cubes with positive sum.

Soft assignment. When we compute DFS for a given sequence, the local density $\alpha(p)$ of each pixel p is first computed w.r.t. each measurement. Then, all pixels are partitioned into different sets $E_{[\alpha_i, \alpha_{i+1}]}$, according to their local density values. In [29], the partition is implemented by a “hard” scheme, that is, a pixel x is assigned to $E_{[\alpha_i, \alpha_{i+1}]}$ iff $\alpha(p) \in [\alpha_i, \alpha_{i+1}]$. Such a scheme is vulnerable to quantization errors, especially for the pixels with fractal dimension close to the end points of the interval. 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

$$m_i(p) = \begin{cases} 1, & \text{if } \alpha(p) \in [\alpha_i, \alpha_{i+1}] \\ \text{tansig}\left(\frac{|\alpha(p) - \alpha_i|}{\tau}\right), & \text{if } \alpha(p) \in A_{\alpha_i, \tau} \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

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 (9) allows intersection between two point sets with adjacent Hölder exponent intervals. It is empirically observed that

the soft assignment improves the robustness of the fractal dimension vector against quantization errors.

In our experiments, we noticed that the results are not very sensitive to the number of levels of α in a reasonable range. The threshold τ has only a little affect on the final results since it is very small in implementation.

4. Experiment

While there exist many static texture datasets, only a limited number of dynamic texture datasets are available due to the difficulties in collecting DT sequences. There are mainly three public DT datasets that have been widely used for DT analysis: the UCLA dataset [6], the DynTex dataset [18] and the DynTex++ dataset [11]. We test the proposed method on all of them in comparison with state-of-the-art DT classification approaches.

In our experiments, the color information is discarded by converting all frames to gray-scale images. For the DFS descriptor, the 16-dim V-DFS vector is computed by only using the measurement (5). The S-DFS uses all four measurements (5)-(8), and each generates a 75-dim vector (25 for each axis). The final DFS descriptor is the concatenation of all these vectors, with the total dimension 316. The parameters are set as the following: $r_t = 2$ for all the datasets, $r_s = 5$ for the UCLA and DynTex++ datasets, and $r_s = 6$ for the DynTex dataset. We noted experimentally that the DFS descriptor is insensitive to small perturbations of these parameters.

4.1. Recognition on the UCLA dataset

The UCLA dynamic texture dataset has been widely used in many previous studies [4, 6, 11, 21, 22]. It originally contains 50 DT classes, each with four grayscale video sequences captured from different viewpoints. Fig. 6 shows some samples from the dataset. There are several different breakdowns when the dataset is used for evaluating DT classification algorithms:



boiling fire flower fountain sea smoke water waterfall

Figure 6. Example snapshots of eight classes used in our experiment from the UCLA dataset.

50-Class: The classification rates for 50 classes are implemented in [4, 11].

Shift-invariant recognition(SIR)-Class: In [4], 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

Table 1. The classification results (in %) on UCLA dataset. Note: Superscripts “S”, “N” and “M” are for results using SVM, 1NN, and maximum margin learning (followed by 1NN) [11] respectively; “-” means “not available”.

Method	7-Class	8-Class	9-Class	50-Class	SIR
[21]	-	80 ^S	-	-	-
[4]	92.3 ^N	-	-	81 ^N	60 ^N
[11]	-	-	95.6 ^M	99 ^M	-
DFS	98.5 ^N	99 ^S	97.5 ^S	100 ^S , 89.5 ^N	73.8 ^N

the end. The “shift-invariant recognition” [4] was implemented to compare the sequences only between different halves to test the shift-invariance of the descriptors.

9-Class: In [11], 50 UCLA DT classes were clustered to 9 classes by combining the sequences from different viewpoints, which were 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 dataset. The dataset is therefore very challenging and serves as an excellent test bed for evaluating DT classification algorithms under viewpoint change.

8-Class: In [21], 9 classes used in [11] are further reduced to 8 classes by removing sequences of “plants”, since it contains too many sequences.

7-Class: In [4], the “semantic category recognition” was also 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 (wind-blown) vegetation (240).

We compare our DFS descriptors with previously tested methods in [4, 6, 11, 21, 22] and use the same experimental setups. The classification accuracies are shown in Table. 1 and the confusion matrices are shown in Fig. 7. It is seen that our approach achieves the best performance in all cases.

4.2. Recognition on the DynTex dataset

The DynTex dataset [19] is a large dataset devoted to the study of DT. The dataset 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 seconds, and de-interlaced with a spatio-temporal median filter.

The DynTex dataset has been used for DT classification experiments in previous study, *e.g.* [10, 12, 31]. However, these studies often use different experimental configurations, *e.g.* different subsets and categories. We follow the

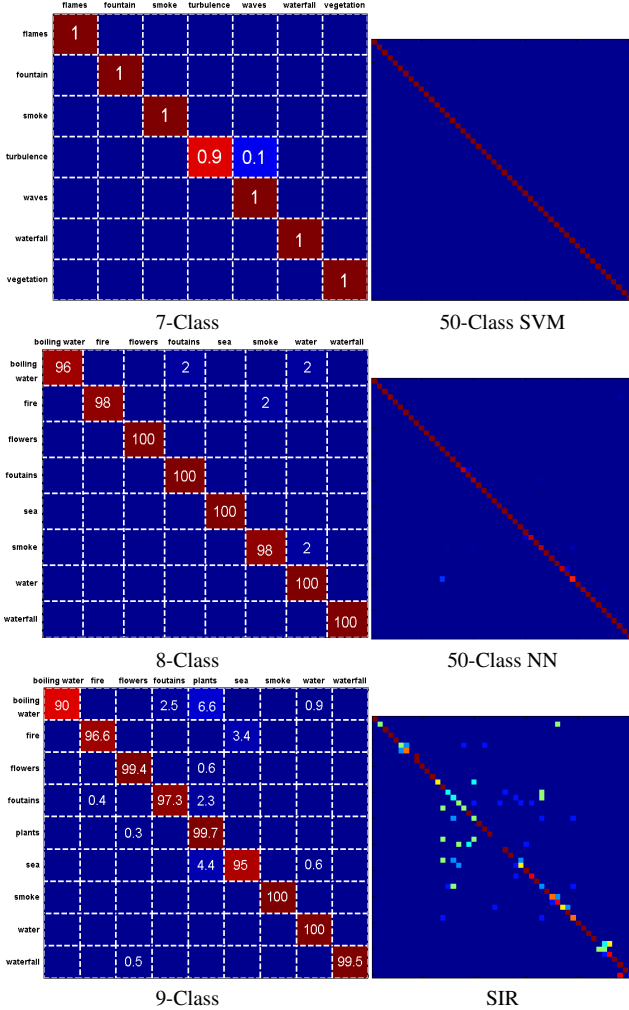


Figure 7. Confusion matrices of DFS on the UCLA dataset.

work in [31] since it not only gives a detailed description of the configuration but also achieves very good recognition performances on DT by using the LBP-TOP.

First, 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 ten sample subsequences with various dimensions. These samples share the same class label of 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 picked to form the testing set and leave the rest samples as the training set. Each class is then represented by the mean feature vector over the samples in the training set. After that, each

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

	LBP-TOP [31]	DFS
non-weighting	95.71	96.28
best-weighting	97.14	97.63

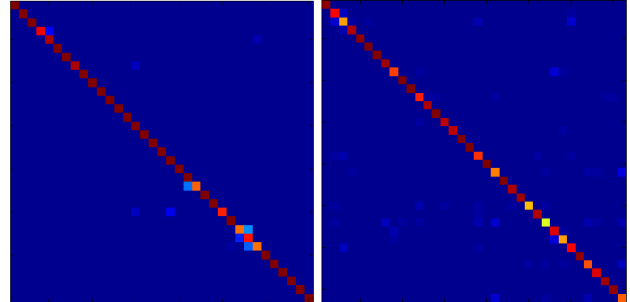


Figure 8. Confusion matrices by our method on the DynTex (left) and DynTex++ (right) datasets.

test sample is classified according to the class that has the smallest ℓ_1 distance in the feature space. Finally the average classification rate over all runs is reported.

The classification rate of the proposed method is summarized in Table 2. Similar to [31], we also tested different weights for each feature dimension to improve the performance. It can be seen from Table 2 that our method performs very well, with recognition rates of 96.28% and 97.63% for non-weighting and best-weighting respectively. Both scores outperform the best results reported in [31]. The confusion matrix is shown in Fig. 8.

It is worth noting that our descriptors require much fewer parameters than those in [31]. Only three simple parameters are considered: the radius r_s shared in (5) - (8), the radius r_t in (5) for estimating the local density function, and the number of levels z for counting the fractal dimension. In practice, we found them rather easy to be determined and the performance is insensitive within reasonable ranges.

4.3. Recognition on the DynTex++ dataset

The DynTex++ dataset proposed in [11] is a challenging dataset comprised 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 [11] in the evaluation. Using SVM as the classifier, we train on 50% of the dataset and test on the rest. We applied our DFS descriptor on DynTex++ and obtained an average recognition rate of 89.9%, which significantly outperforms previously tested methods (Table 3). The classification rates for each class are shown in Fig. 9 and the confusion matrix is shown in Fig. 8.

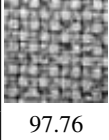
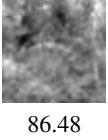



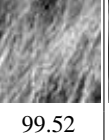

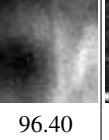
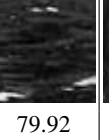
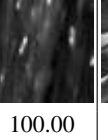



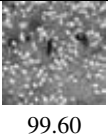




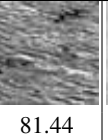

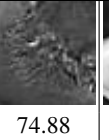
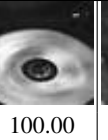










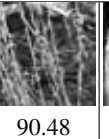
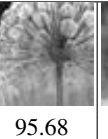
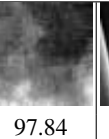

											
97.76	86.48	71.20	94.56	98.40	99.52	100.00	96.40	79.92	100.00	100.00	83.36
											
94.24	99.60	93.36	92.24	98.56	98.48	81.44	100.00	74.88	100.00	95.28	95.60
											
68.56	95.12	58.00	89.36	71.76	86.56	97.52	78.80	90.48	95.68	97.84	77.76

Figure 9. Classification rate (%) on each class of the DynTex++ dataset.

Table 3. Results on the DynTex++ dataset

Method	DL-PEGASOS [11]	DFS
Classification accuracy (%)	63.7	89.9

5. Conclusion

We presented a powerful DT descriptor using dynamic fractal analysis developed in this paper. The proposed DFS descriptor consists of two components: V-DFS component and S-DFS, which capture the 3D fractal structures in DT from different perspectives. DFS effectively captures the stochastic self-similarities existing in a wide range of DT sequences. Experiments on the UCLA, DynTex and DynTex++ datasets demonstrated that our proposed descriptor compares favorably against existing state-of-the-art methods.

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

References

- [1] V. A. Billock, G. C. Guzman, and J. S. Kelso. "Fractal time and $1/f$ spectra in dynamic images and human vision." *Physics D*, 148:136–146, 2001. 1, 2
- [2] A. Chan and N. Vasconcelos. "Classifying video with kernel dynamic textures." *CVPR*, 2007. 2
- [3] D. Chetverikov and R. Péteri. "A brief survey of dynamic texture description and recognition." *ICCRS*, 2005. 1, 2
- [4] K. G. Derpanis and R. P. Wildes. "Dynamic texture recognition based on distributions of spacetime oriented." *CVPR*, 2010. 6
- [5] D. W. Dong and J. J. Attrick. "Statistics of natural time-varying images." *Network*, 345–358, 1995. 1, 2
- [6] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. "Dynamic texture." *IJCV*, 2003. 1, 6
- [7] G. Doretto, D. Cremers, P. Favaro, and S. Soatto. "Dynamic texture segmentation." *ICCV*, 2003. 1
- [8] G. Doretto, E. Jones, and S. Soatto. "Spatially homogeneous dynamic texture." *ECCV*, 591–602, 2004. 2
- [9] K. Falconer. *Techniques in fractal geometry*. John Wiley, 1997. 2, 3
- [10] S. Fazekas and D. Chetverikov. "Normal versus complete flow in dynamic texture recognition: A comparative study." *Workshop on Texture Analysis and Synthesis*, 37–42, 2005. 6
- [11] B. Ghanem and N. Ahuja. "Maximum margin distance learning for dynamic texture recognition." *ECCV*, 2010. 6, 7, 8
- [12] B. Ghanem and N. Ahuja. "Phase based modelling of dynamic textures." *ICCV*, 2007. 2, 6
- [13] B. Ghanem. "Dynamic textures: Models and applications." Ph.D Thesis, UIUC, 2010. 1
- [14] J. V. Hateren. "Processing of natural time series of intensity by the blowfly visual system." *Vision Research*, 37:3407–3416, 1997. 1, 2
- [15] Z. Lu, W. Xie, J. Pei, and J. Huang. "Dynamic texture recognition by spatio-temporal multi-resolution histogram." *IEEE Wksp. on Motion and Video Computing*, 241–246, 2005. 2
- [16] B. Mandelbrot. *The fractal geometry of nature*. Freeman, 1982. 1, 2
- [17] A. Pentland. "Fractal-based description of natural scenes." *PAMI*, 6(6):661–674, 1984. 2
- [18] R. Péteri and D. Chetverikov. "Dynamic texture recognition using normal flow and texture regularity." *lbPRIA*, 2005. 2, 6
- [19] R. Péteri and M. Huskies. DynTex: A comprehensive database of dynamic texture. <http://www.cwi.nl/projects/dyntex/>, 2005. 4, 5, 6
- [20] R. Polana and R. Nelson. "Temporal texture and activity recognition." *Motion-based recognition*, 1997. 2
- [21] A. Ravichandran, R. Chaudhry, and R. Vidal. "View-invariant dynamic texture recognition using a bag of dynamical systems." *CVPR*, 2009. 2, 6
- [22] P. Saisan, G. Doretto, Y. Wu, and S. Soatto. "Dynamic texture recognition." *CVPR*, 58–63, 2001. 2, 6
- [23] J.R. Smith, C.Y. Lin, and M. Naphade. "Video indexing using spatio-temporal wavelets." *ICIP*, 2002. 1, 2
- [24] M. Szummer and R. W. Picard. "Temporal texture modeling." *ICIP*, 1996. 2
- [25] M. Varma and R. Garg. "Locally invariant fractal features for statistical texture classification." *ICCV*, 2007. 2
- [26] P. Viola and M. Jones. "Robust real-time face detection." *ICCV*, 2001. 5
- [27] R.P. Wildes and J.R. Bergen. "Qualitative spatio-temporal analysis using an oriented energy representation." *ECCV*, 768–784, 2000. 2
- [28] F. Woolfe and A. Fitzgibbon. "Shift-invariant dynamic texture recognition." *ECCV*, 549–562, 2006. 2
- [29] Y. Xu, H. Ji, and C. Fermüller. "Viewpoint invariant texture description using fractal analysis." *IJCV*, 83(1):85–100, 2009. 2, 5
- [30] Y. Xu, X. Yang, H. Ling, and H. Ji. "A new texture descriptor using multifractal analysis in multi-orientation wavelet pyramid." *CVPR*, 2010. 2

- [31] G. Zhao and M. Pietikäinen. "Synamic texture recognition using local binary patterns with an application to facial expression." *PAMI*, 29(6):915-928, 2007. [2](#), [6](#), [7](#)