

# Chaotic features for dynamic textures recognition

Yong Wang · Shiqiang Hu

Published online: 21 February 2015  
© Springer-Verlag Berlin Heidelberg 2015

**Abstract** This paper presents a novel framework for dynamic textures (DTs) modeling and recognition, investigating the use of chaotic features. We propose to extract chaotic features from each pixel intensity series in a video. The chaotic features in each pixel intensity series are concatenated to a feature vector, chaotic feature vector. Then, a video is modeled as a feature vector matrix. Next, two approaches of DTs recognition are investigated. A bag of words approach is used to represent each video as a histogram of chaotic feature vector. The recognition is carried out by 1-nearest neighbor classifier. We also investigate the use of earth mover's distance (EMD) method. Mean shift clustering algorithm is employed to cluster each feature vector matrix. EMD method is used to compare the similarity between two videos. The output of EMD matrix whose entry is the matching score can be used to DTs recognition. We have tested our approach on four datasets and obtained encouraging results which demonstrate the feasibility and validity of our proposed methods.

**Keywords** Chaotic features · Bag of words (BoWs) · Earth mover's distance (EMD) · Dynamic textures recognition

---

Communicated by V. Loia.

---

This paper is jointly supported by the National Natural Science Foundation of China No. 61374161, China Aviation Science Foundation 20142057006.

---

Y. Wang (✉) · S. Hu  
School of Aeronautics and Astronautics, Shanghai Jiao Tong University,  
Shanghai 200240, China  
e-mail: wysjtu2008@gmail.com

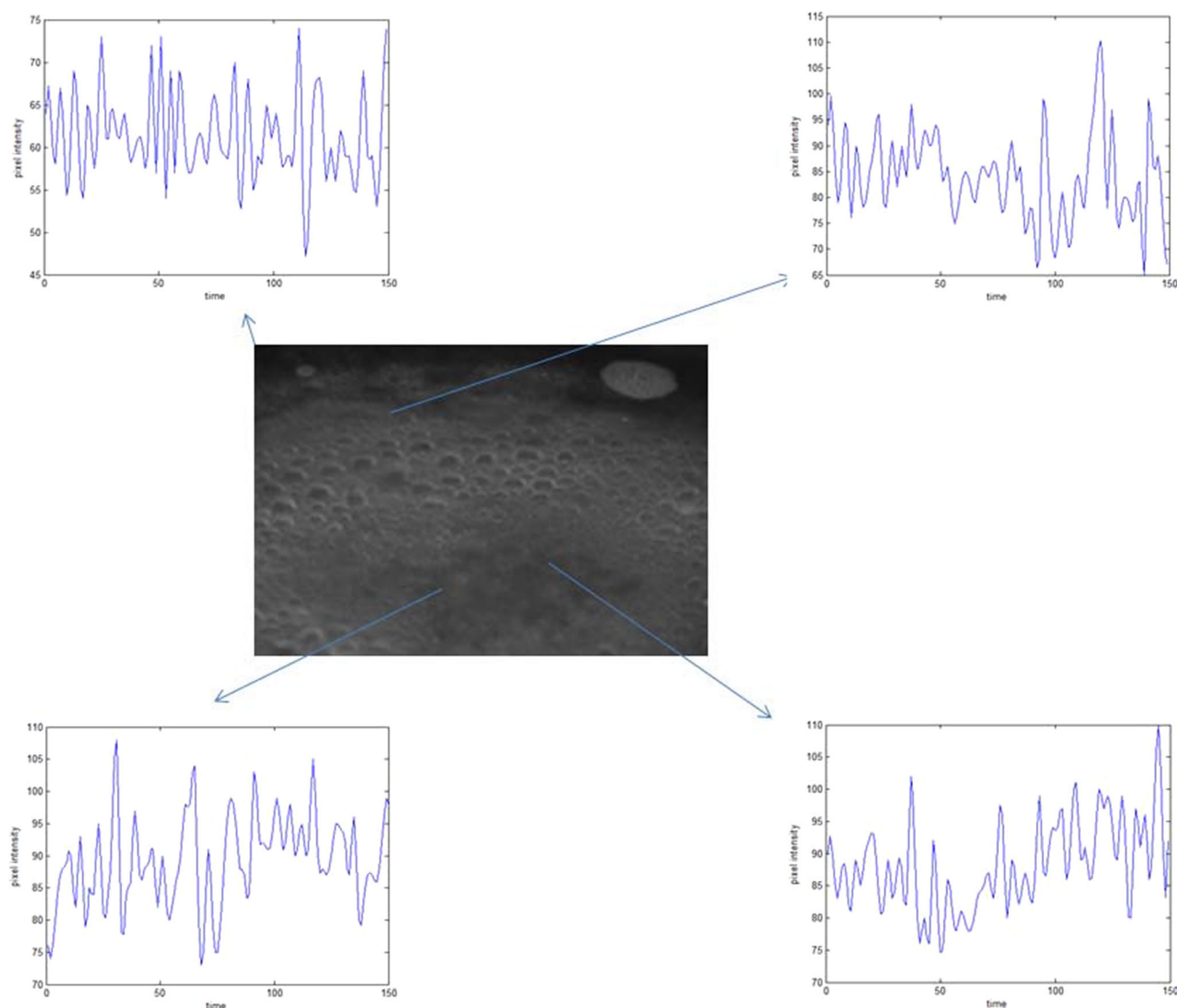
S. Hu  
e-mail: sqhu@sjtu.edu.cn

## 1 Introduction

Dynamic textures (DTs) are video sequences of moving scenes that exhibit certain stationary properties in time domain (Doretto et al. 2003) which can be observed everywhere in our daily life, such as dancing grass, turbulent water, a crowd of people and so on. The applications of DTs are widely spread in many research fields, to name a few, layer segmentation, texture synthesis for realistic rendering and texture segmentation for localizing textures. Therefore, many methods are proposed to model DTs (Doretto et al. 2003; Chan and Vasconcelos 2005a, b, 2007; Ravichandran et al. 2012; Peteri and Chetverikov 2005; Chetverikov and Pteri 2005; Fazekas and Chetverikov 2005; Szummer and Picard 1996; Bar-Joseph et al. 2001; Fitzgibbon 2001; Wang and Zhu 2003; Sivic and Zisserman 2003; Laptev and Lindeberg 2003). Recently, many work advocates the use of linear dynamical systems (LDSs) or a variety of LDSs for DTs recognition. While DTs are generated by a complex time varying dynamical system, the LDSs model is constrained to linear assumption which makes it restrictive for modeling DTs in reality. Take boiling water for example, it presents a chaotic characteristic and these dynamical systems are difficult to be described by linear system.

Motivated by these challenges, we propose a chaotic feature vector for DTs recognition. Unlike the traditional methods for static textures which only pay attention to the spatial relation between pixels, the temporal property of the pixel intensity series is an important clue for DTs modeling. In DTs, each pixel intensity series can be treated as a chaotic time series.

Figure 1 illustrates DTs of boiling water from dataset (Doretto et al. 2003). The central part of the figure shows one frame from the video of boiling water. It also shows four pixel intensity series. The  $x$ -axis is frame number and the  $y$ -



**Fig. 1** Pixels intensity in a video changed over time

axis is the gray value. From Fig. 1, people cannot figure out the size of the boiling water is  $10^2$  cm large or  $10^2$  m, without something or some tool next to it. This is usually called self-similarity. That is, the object has the same structure at all scales. The natural scenes such as coastline possess the common characteristics of self-similarity. Fractal dimension is developed to measure the self-similarity property. Many physical processes produce self-similarity property and natural scene can be modeled by fractal dimension (Pentland 1984). Natural textures have a linear log power spectrum which is related to the fractal dimension and is suitable to characterize textures (Field 1987). Since the stationary property of DTs, we conjecture that self-similarity exists in each pixel intensity series.

Suppose there is a collection of  $V = v_1, \dots, v_N, v \in R^{W \times L \times T}$  video sequences, where  $W$ ,  $L$ ,  $T$ , and  $N$  are

width, length, frame number and total number of the sequence, respectively. A one-dimensional pixel intensity series  $\{x_{i,j}(t)\}_{t=1}^T = v_q(i, j, :)$ , where  $i$  and  $j$  are horizontal and vertical coordinate of  $x_{i,j}(t)$  in video, respectively. Each pixel intensity series  $x_{i,j}$  can be represented by a chaotic feature vector. A DT video can be represented by a feature vector matrix. Then, we use two methods for DTs recognition.

For the first method, we follow the well-known bag of words (BoWs) approach which has been adopted by many computer vision researchers (Chen and Prasanna 2013; Fei-Fei and Perona 2005; Lazebnik et al. 2006). A codebook is learned by clustering all the feature vector matrix. During clustering, each chaotic feature vector is assigned to the codeword that is closest to it in terms of Euclidean distance. These representative chaotic feature vectors are called codewords in the context of BoWs approach. After the generation

of the codebook, each feature vector matrix is represented by a histogram based on chaotic feature vectors.

However, the codebook size will affect the result of recognition accuracy rate. The reason is that quantize chaotic feature vector space into fixed-size bins of histogram will lose the structure information. To obtain a more compact and descriptive representation of the distribution of chaotic feature vectors in a video, we perform clustering on each feature vector matrix first. Signatures which summarize the distribution of chaotic feature vectors in each feature vector matrixes provide a measure of similarity between two videos. Then, earth mover's distance (EMD) (Rubner et al. 1998) method is employed to compare similarity between feature vector matrixes.

The contribution of our paper lies in: (1) investigation of the appropriateness of chaotic dynamical system for DTs modeling and recognition, (2) a new chaotic feature vector is proposed to characterize nonlinear dynamics of DTs, (3) experimental validation of the feasibility and potential merits of carrying out DTs using methods from chaotic dynamical system.

The rest of the paper is organized as follows: Sect. 2 discusses related work. The video representation we explore is presented in Sect. 3. Section 4 describes our approach in detail, including the chaotic features, a brief overview of the BoWs approach, EMD method approach, and the specifics of learning and recognition procedures. Recognition results are provided and discussed in Sect. 5. Section 6 concludes the paper.

## 2 Related works

In this section, we review previous work on DTs recognition. DTs recognition has been studied for decades. The existing methods that model DTs can be categorized into three approaches: (1) Physics based approach that derives a model of the DTs. The model is simulated to synthesize the surface of the ocean (Fournier and Reeves 1986). The main disadvantage of this method is that the model is closely tied to specific physical process and thus difficult to generalize to a large class of DTs.

(2) Image-based approach based on frame-to-frame estimation to extract motion field features. Various motion features are proposed, normal flow (Peteri and Chetverikov 2005) and optical flow features (Chetverikov and Pteri 2005; Fazekas and Chetverikov 2005) which are computation efficient and natural way to depict the local DTs. The main drawback of this approach is that the flow features (e.g., optical flow) are computed based on the assumption of local smoothness and brightness constancy. The non-smoothness, discontinuities DTs are difficult to process.

(3) Statistical generative models to jointly capture the spatial appearance and statistical models have been extensively studied. They include auto-regressive models (Szummer and Picard 1996) multi-resolution analysis (Bar-Joseph et al. 2001). Recently, many work models the DTs as LDSs. LDSs are learned by system identification to model DTs (Doretto et al. 2003). UCLA dataset is provided which contains 200 videos and widely used as a benchmark dataset in varies of DTs recognition methods. Gaussian mixture models (GMMs) of LDSs are also used to model DTs. And expectation-maximization algorithm is derived for learning and recognizing a mixture of DTs (Chan and Vasconcelos 2005a). Then, LDSs model is extend with a nonlinear observation to recognize DTs (Chan and Vasconcelos 2007). A probabilistic kernel is derived which is capable to describe both the spatial-temporal process and temporal process (Ravichandran et al. 2012). BoWs approach is used in Ravichandran et al. (2012) to model each DT video with LDSs and recognize DTs. They propose bag-of-systems that is analogous to the BoWs approach for DTs recognition. The method in Ravichandran et al. (2012) obtains promising results that is better than state of the art methods.

Chaotic dynamical system has been studied extensively in physics community (Kantz and Schreiber 1997) and is introduced into computer vision community recently (Ali et al. 2007; Wu et al. 2010; Shroff et al. 2010). Trajectories of reference points are used in Ali et al. (2007) as time series. And chaotic features are extracted and combined to a feature vector to characterize the different motion properties of time series. Experimental results validate the feasibility and merits of using method from chaotic dynamical system. People's tracks are treated as time series in Wu et al. (2010). Chaotic features are calculated to detect and locate anomalies. Other chaotic features are used in image processing. A modified box-count approach is proposed to estimate fractal dimension and experiment of image segmentation is effective (Chaudhuri and Sakar 1995).

Inspired by the work mentioned above, we are interested in exploring the use of typical recognition framework in conjunction with a representation based on chaotic feature vector. We present our proposed algorithm in the following section.

## 3 Chaotic dynamical system

In this section, we present the background material related to the chaotic dynamical system. The dynamical system can be depicted by a state space models  $y(t) = f_m(y(t-1))$ , where  $y(t)$  is the observation at time  $t$ , and  $f_m$  is a mapping function. The mapping function  $f_m$  can be computed by system identification if the system is linear (Doretto et al. 2003). However, when the system is nonlinear, especially in

the natural wild system, it is not an easy work to compute the mapping function  $f_m$ . By virtue of chaotic dynamical system, we calculate two chaotic features of pixel intensity series  $y(t)$  instead of computing mapping function  $f_m$ . We next describe the framework to compute chaotic features.

Takens' theorem (Taken 1981) states that a map exists between the original state space and a reconstructed state space. That is the pixel time series  $\{x_{i,j}(t)\}_{t=1}^T$  can be written into a matrix:

$$x_{ij} = \begin{pmatrix} x_0 & x_{\tau_{ij}} & \dots & x_{(m_{ij}-1)*\tau_{ij}} \\ x_1 & x_{\tau_{ij}+1} & \dots & x_{(m_{ij}-1)*\tau_{ij}+1} \\ x_2 & x_{\tau_{ij}+2} & \dots & x_{(m_{ij}-1)*\tau_{ij}+2} \\ \dots & \dots & \dots & \dots \end{pmatrix} \tag{1}$$

where  $\tau_{ij}$  is embedding time delay and  $m_{ij}$  is embedding dimension.  $\tau_{ij}$  and  $m_{ij}$  can be computed by mutual information algorithm (Fraser et al. 1986) and false nearest neighbor algorithm (Kennel et al. 1992), respectively.

Chaotic features are measures that quantify the properties that are invariant under transformations of the state space. We next introduce chaotic features used in this paper.

### 3.1 Box-count dimension

Box-count dimension (Kantz and Schreiber 1997) measures the degree of a set holds in space. If a point set is covered with a regular grid of boxes of length  $\epsilon$  and  $N(\epsilon)$  is the number of boxes which contain at least one point, then box counting dimension  $D_b$  is

$$D_b = \lim_{\epsilon \rightarrow 0} \frac{\ln N(\epsilon)}{\ln \frac{1}{\epsilon}} \tag{2}$$

### 3.2 Correlation dimension

Correlation dimension  $D_c$  (Peter Grassberger and Itamar Procaccia 1983) characterizes the system complexity and calculated as the slope of  $\text{Inc}(\epsilon)$  versus  $\ln(\epsilon)$ ,

$$D_c = \lim_{\epsilon \rightarrow 0} \frac{\text{Inc}(\epsilon)}{\ln \epsilon} \tag{3}$$

where  $\epsilon$  is radius and  $c(\epsilon)$  is correlation integral (Kantz and Schreiber 1997).

When the one-dimensional pixel intensity series transformed to an  $m$  dimensional phase space, the box-count dimension and the correlation dimension can be used to measure the smoothness of the transformed phase space. The smooth the phase space, the smaller are the two dimensions.

	X-Y	Embedding Delay	Embedding Dimension	Box-count Dimension	Correlation Dimension	Mean	Standard Variance
Boiling	10, 10	2	4	0.14	0.07	78.22	6.0
candle	20,11	2	4	0.12	0.21	229.86	15.22
Flowers	20,10	4	7	0.06	0.026	111.78	17.35
Fountain	20,20	2	9	0.02	0.013	134.27	24.73
plant	30,20	4	6	0.44	0.266	215.29	15.26
Sea	30,20	2	9	0.034	0.052	120.47	13.14
Smoke	30,30	2	7	0.27	0.087	125.14	14.86
water	30,40	2	8	0.036	0.017	141.34	10.58
Waterfall	45,40	4	8	0.007	0.003	102.35	26.07

Fig. 2 Features computed from videos

### 3.3 Chaotic feature vector

Given a video  $v_q$ , embedding time delay and embedding dimension are two important parameters to determine the geometry information in the phase space reconstruction. Box-count dimension and correlation dimension provide complementary information to characterize self-similarity property. Our chaotic feature vector is  $cf_{ij} = \{\tau_{ij}, m_{ij}, D_{b_{ij}}, D_{c_{ij}}\}$ . A video  $v_q$  can be transformed to a feature vector matrix and each pixel intensity series  $x_{i,j}(t)$  is represented by feature vector  $cf_{ij}$ . The video  $v_q$  is represented by a  $W * L * 4$  dimensional features vector matrix. In Fig. 2, we give the results of features computed in several videos. Details of the datasets will be given in Sect. 5.  $X - Y$  denotes the horizontal and vertical coordinates of pixel intensity series.

## 4 Recognition algorithm

To investigate whether the feature vector is suitable for DTs recognition, we present two recognition methods to validate our conjecture.

### 4.1 BoWs approach

After the chaotic feature vector is obtained, DTs are represented as a collection of codewords in a pre-defined codebook. In the BoWs approach, a text document is encoded as a histogram of the number of occurrences of each codeword. Similarly, a video can be characterized by a histogram of codewords count according to

$$h(v_q) = (h_i(v_q))_{i=1,\dots,K}, \quad \text{with } (h_i(v_q)) = n(v_q, cw_i) \tag{4}$$

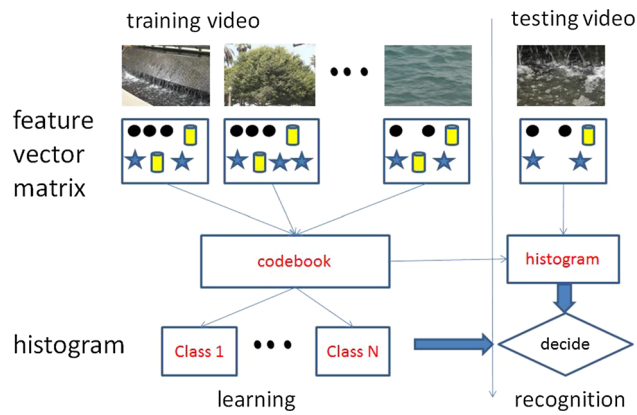


Fig. 3 Flowchart of BoWs approach

where  $n(v_q, cw_i)$  denotes the number of occurrences of codeword  $cw_i$  in video  $v_q$  and  $K$  is the codebook size. A codebook consists of a set of representative chaotic feature vectors learned from training samples.

Figure 3 shows an overview of BoWs approach in both learning and recognition. In learning step, each video is represented by a feature vector matrix which is mentioned in Sect. 3. The  $K$ -means clustering algorithm is employed to obtain the cluster centers which generate the codebook. Each class can be represented by a histogram of the codebook after all feature vector matrixes are mapped to the cluster centers using nearest neighbor algorithm. The goal of learning is to achieve a model that best represents each DTs' class. In recognition, we find the unknown histogram of a video that fits best the model of a particular class via 1-nearest neighbor (1-NN).

In Fig. 4, we show ten examples from testing videos with their corresponding DTs codewords histograms to demonstrate discrimination of the distribution of the learned DTs codewords. It is clear to see that DTs from each category have different dominant peaks. Meanwhile, different categories have some overlap bins. That is the reason of confusing different classes.

## 4.2 EMD-based approach

### 4.2.1 Overview of the framework

Figure 5 shows a summary of EMD method approach. Chaotic feature vectors in each video are first computed to form a feature vector matrix as stated in Sect. 3. The mean shift algorithm is employed to group the chaotic feature vectors in each video into clusters. Then, the EMD method (Rubner et al. 1998) is used to handle the degree of similarity between the feature vector matrix. The output is an EMD cost matrix which can be used for learning and recognition.

Details about the generation of chaotic feature vector clustering and the chaotic feature vector matching are presented in the following.

### 4.2.2 Chaotic feature vector clustering and matching

Mean shift clustering algorithm, unlike  $K$ -means and GMMs need to define the number of clusters ahead, is a non-parametric clustering algorithm. It is suitable to cluster non-Gaussian feature space. Therefore, we use mean shift algorithm (Comaniciu and Meer 2002) for chaotic feature vector clustering. Other clustering methods that do not require a priori knowledge about the number of clusters can also be used.

To compute similarities between videos that are represented by cluster centers, we need to define an appropriate similarity measure. EMD method is appropriate to compute the cluster centers' similarities as signature represents a set of chaotic feature vectors. Matching cluster centers can be naturally cast as a transportation problem (Dantzig 1951) by defining one cluster center as the supplier and the other as the consumer, and by setting the cost for a supplier-consumer pair to equal the ground distance between an element in the first cluster center and an element in the second (Fig. 6).

Let  $P = \{(p_i, wp_i) \mid 1 \leq i \leq m\}$  and  $Q = \{(q_j, wq_j) \mid 1 \leq j \leq n\}$  be two cluster centers, where  $p_i$  and  $q_j$  are the mean chaotic feature vector,  $wp_i$  and  $wq_j$  are the weight of cluster centers, and  $m$  and  $n$  are the number of the chaotic feature vector. The distance is as follows:

$$EMD(P, Q) = \frac{\sum_{i=1}^m \sum_{j=1}^n d_{ij} f_{ij}}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}} \tag{5}$$

where  $D = \{d_{ij}\}$  is the distance between two cluster centers  $p_i$  and  $q_j$ .  $F = [f_{ij}]$  is the flow between  $p_i$  and  $p_j$ . Equation (5) is governed by the following constraints:

$$f_{ij} \geq 0, \quad 1 \leq i \leq m, \quad 1 \leq j \leq n, \tag{6}$$

$$\sum_{j=1}^n f_{ij} \leq wp_i, \quad 1 \leq i \leq m, \tag{7}$$

$$\sum_{i=1}^m f_{ij} \leq wq_j, \quad 1 \leq j \leq n, \tag{8}$$

$$\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \min \left( \sum_{i=1}^m wp_i, \sum_{j=1}^n wq_j \right) \tag{9}$$

The EMD cost matrix is then used in a form of Gaussian kernel as follows:

$$\text{Kernel}(P, Q) = \exp \left( -\rho EMD(P, Q)^2 \right) \tag{10}$$

where  $\rho$  is the kernel parameter. The transformed EMD cost matrix is then used for DTs classification.

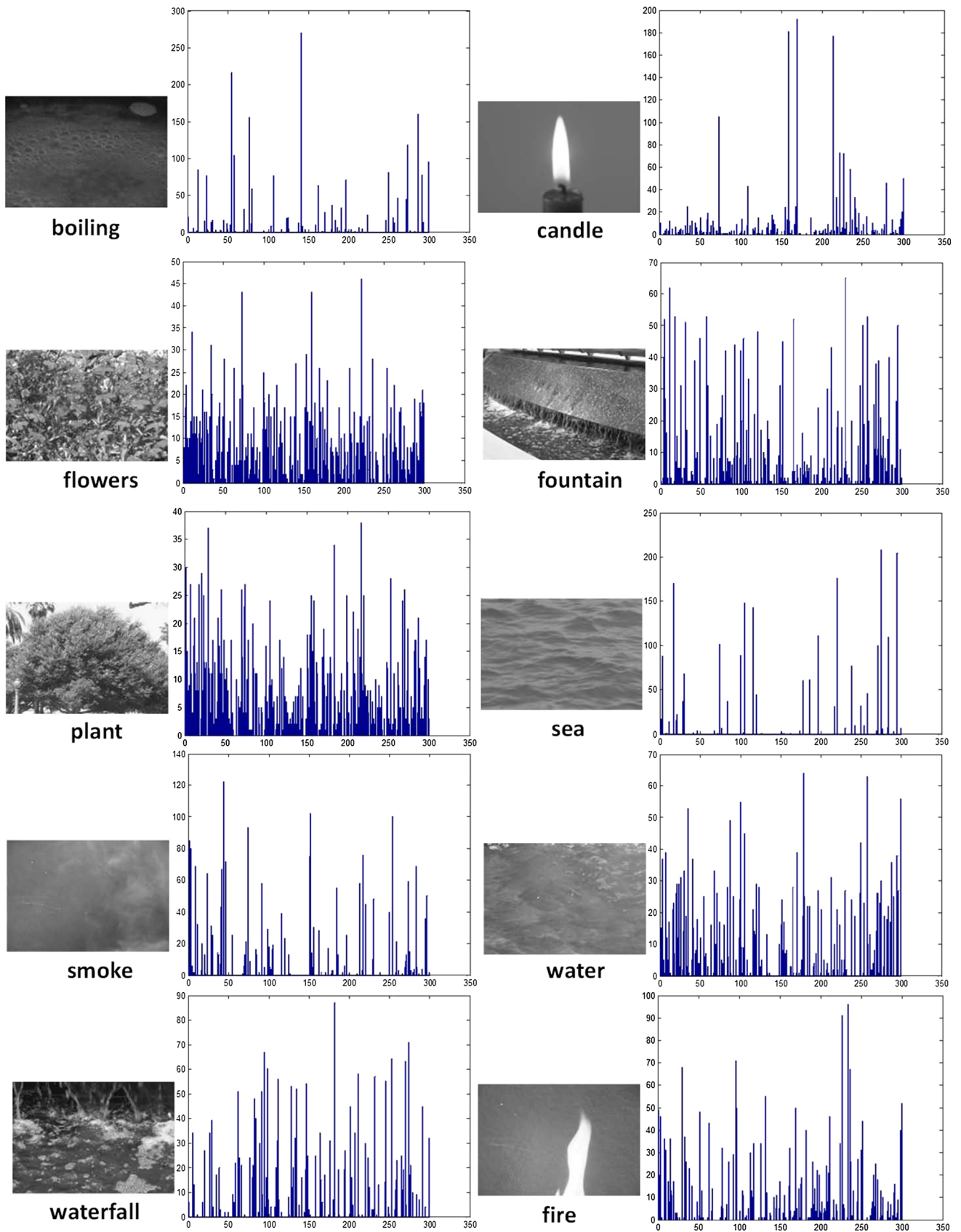
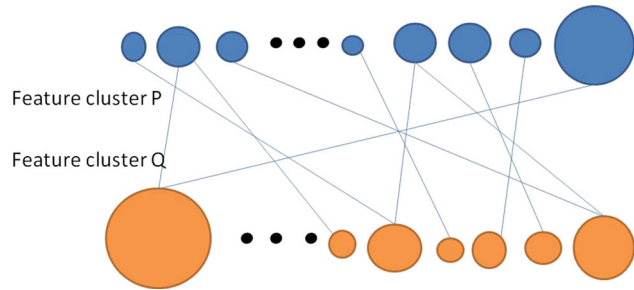
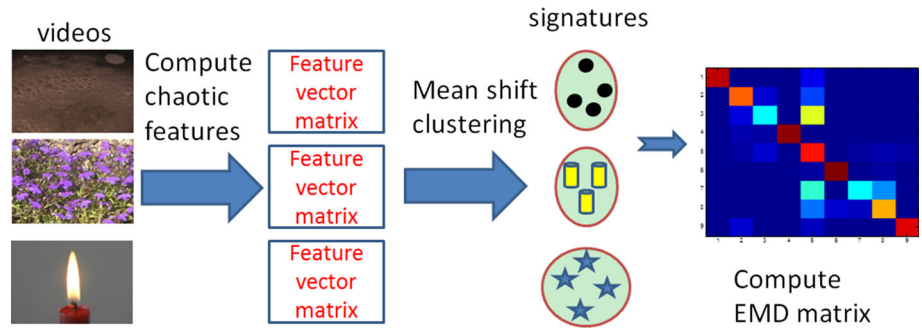


Fig. 4 Example histograms of the codebook (codebook size,  $K = 300$ ) for ten selected testing DTs

**Fig. 5** Process flow of EMD-based approach



**Fig. 6** Example of EMD-based matching between two feature clusters P and Q; *lines* indicate flows between two clusters

4.3 Spatial–temporal feature recognition algorithm

Spatial–temporal feature which is widely used in action recognition (Dollar et al. 2005) and DTs recognition (Ravichandran et al. 2012) is employed in our EMD-based approach as a baseline method. The BoWs approach of spatial–temporal feature has been used in Ravichandran et al. (2012). The feature produces dense spatial–temporal features that can improve the recognition performance. It includes two separate filters in spatial and temporal directions: 2-D Gaussian filter in space and 1-D Gabor filter in time. The response function at position  $(x, y, t)$  is as follows:

$$R = \{I * g_{\sigma}(x, y) * h_{ev}(t)\}^2 + \{I * g_{\sigma}(x, y) * h_{od}(t)\}^2 \quad (11)$$

where  $g_{\sigma}(x, y)$  is the 2-D Gaussian spatial Gaussian filter, and  $h_{ev}$  and  $h_{od}$  are a quadrature pair of 1D Gabor filter in time domain, which are defined as:

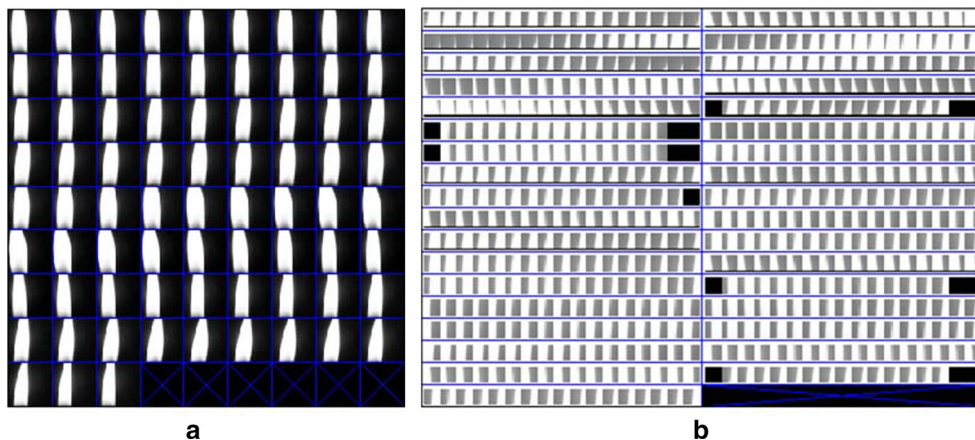
$$h_{ev}(t) = -\cos(2 * \pi * t * \omega) \exp(-t^2/\tau^2) \quad (12)$$

$$h_{od}(t) = -\sin(2 * \pi * t * \omega) \exp(-t^2/\tau^2) \quad (13)$$

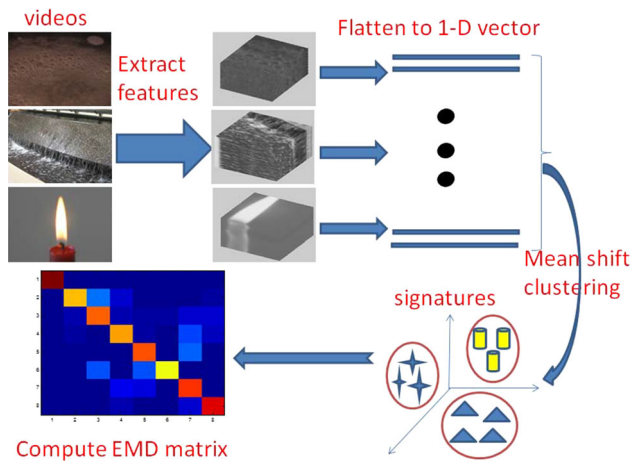
where  $\omega = 4/\tau$ . Figure 7a shows the original video of each frame, and Fig. 7b shows the cuboid. For 3D video cuboids, we concatenate each column to flatten them into 1D vector and follow the similar step of EMD-based approach as mentioned above. The recognition process is shown in Fig. 8.

5 Experiment

In this section, we present an evaluation of the proposed two algorithms on four diverse datasets: UCLA-8 dataset, UCLA-9 dataset, newDT-10 dataset, and DynTex++ dataset. The comparison is performed with other methods that have reported on these datasets.



**Fig. 7** Spatial–temporal feature



**Fig. 8** Recognition process

## 5.1 Implementation detail

### 5.1.1 Datasets

The UCLA dataset is used in [Chan and Vasconcelos \(2005b\)](#), [Ravichandran et al. \(2012\)](#) and [Saisan et al. \(2001\)](#) which contains different DTs, such as boiling water, candle, fire, flowers and so on. The video sequences are gray scale in each class with 75 frames and the dimension is  $48 * 48$ . Figure 9 shows nine examples from the UCLA dataset.

Most of the LDSs-based methods use UCLA dataset as a test bed. Thus, we adopt the UCLA dataset to test our proposed method. The UCLA dataset can be classified to 9 class datasets which are boiling water (8), fire (8), flowers (12), fountains (20), plants (108), sea (12), smoke (4), water (12)



**Fig. 9** Examples from the UCLA dataset

and waterfall (16), where the numbers denote the number of sequences in the dataset. The dataset can be further reduced to 8 class by removing sequences of 'plants' since the number of sequences of plant far outnumber the number of the other classes. These four datasets are a challenging test bed to address the DTs categorization problem.

To test our method in unconstrained conditions (e.g., camera motion), we collect 16 river videos and combine the videos with UCLA-9 dataset to a new dataset, named newDT-10 dataset. The new videos which are captured with smooth shaking are gray scale in each class with 75 frames and the dimension is  $48 * 48$ . This dataset is a challenging testing bed to address the DTs categorization problem.

The fourth dataset is DynTex++ dataset ([Ghanem and Ahuja 2010](#)) which contains 36 categories of different DTs and 100 in each category. In this dataset, there contains a total of 3,600 videos which provide a richer benchmark.

### 5.1.2 Codebook formation

Our chaotic feature vector is a 4-attribute, which consists of embedding time delay, embedding dimension, box-count dimension and correlation dimension. The chaotic feature vectors are normalized to have values between 0 and 1. For generating the codebook, we use *K*-means clustering algorithm directly on the Euclidian distance of the 4-attribute across the entire feature vector matrix and obtain the cluster centers, which form our histogram bins. The number of the clusters *K* is the codebook size which varies from  $K = 100, 200, \text{ to } 1,000$ . After formation of the codebook, each 4-attribute chaotic feature vector of a feature vector matrix is mapped to a certain cluster center, which should be the nearest neighbor of that chaotic feature vector. After all chaotic feature vectors of a feature vector matrix are mapped to the cluster centers, the feature vector matrix can be represented by a histogram of the codebook.

### 5.1.3 Recognition method

1-NN classifier is chosen as the classifier with 50% of the dataset for training and the rest for testing. The results reported in this paper have been averaged over 10 times.

Features used in this paper:

We compare the performance of our approach with four feature-based methods: single LDS approach ([Saisan et al. 2001](#)), 3D SIFT ([Scovanner et al. 2007](#)), Spatial temporal feature and pixel intensity series. We briefly explain these features and give some implementation details.

**Single LDS Approach:** We model the entire DTs video using a single LDS. Given a test DT video, we compute the Martin distance ([Cock and Moor 2000](#)) and Fisher distance between the testing LDS and each of the LDS models of the training set.



3D SIFT that represents the 3D nature of videos is employed in the BoWs approach. We used the original code provided by the authors at <http://crcv.ucf.edu/source/3D>.

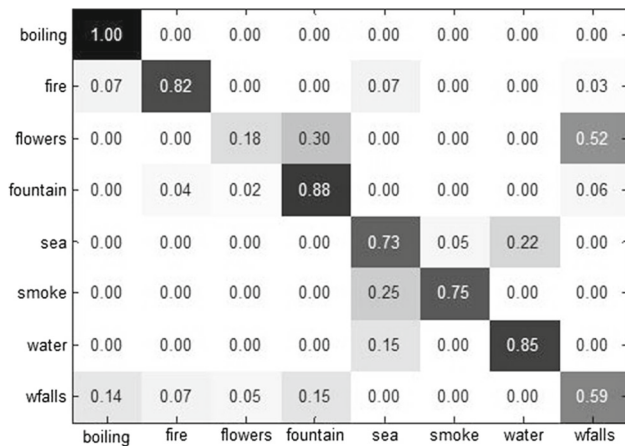
Spatial-temporal feature parameters:

The spatial-temporal feature parameters are set to  $\sigma = 1.5$  and  $\tau = 2.5$ .

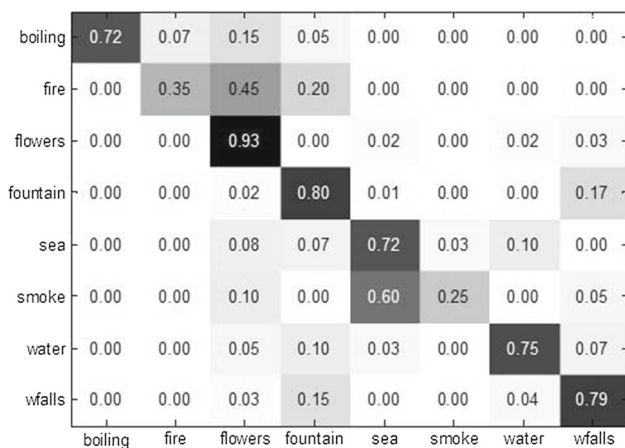
Pixel intensity series: The pixel intensity series are treated as a basic feature and implemented by the BoWs approach and EMD approach, respectively, for comparison with chaotic feature vector.

### 5.2 UCLA-8 dataset

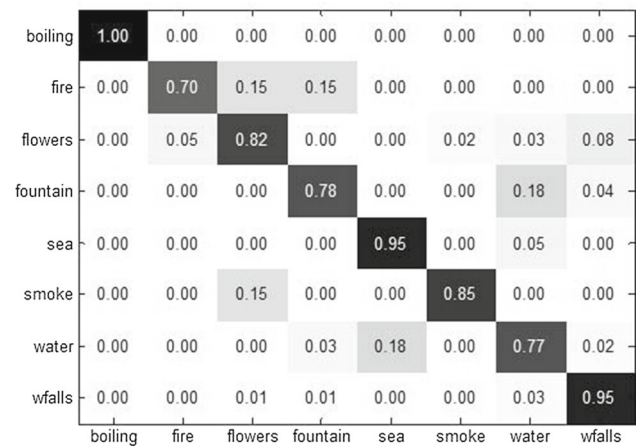
Figure 10 shows the confusion matrix for spatial-temporal feature approach on the UCLA-8 dataset corresponding to the recognition rate 71.52%. Figures 11 and 12 show the confusion matrix for BoWs approach and EMD-based approach on the UCLA-8 dataset corresponding to the recognition rate 72.83 and 85% respectively. Several methods



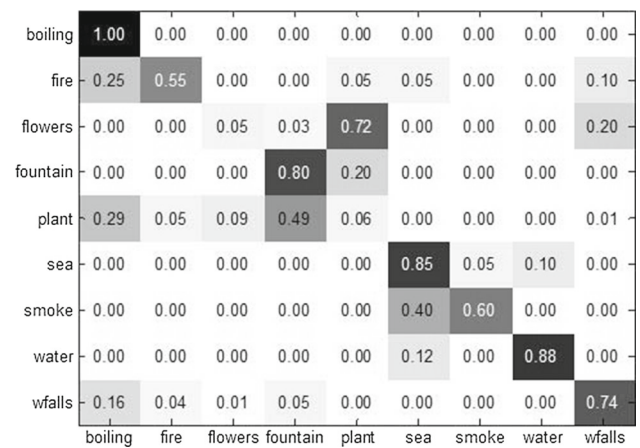
**Fig. 10** Confusion matrix of spatial-temporal feature approach on UCLA-8 dataset. The overall recognition performance is 71.52%



**Fig. 11** Confusion matrix of BoWs approach on UCLA-8 dataset. The overall recognition performance is 72.83%



**Fig. 12** Confusion matrix of EMD-based approach on UCLA-8 dataset. The overall recognition performance is 85%



**Fig. 13** Confusion matrix of spatial-temporal feature approach on UCLA-9 dataset. The overall recognition performance is 35.3%

such as spatial-temporal feature with BoWs approach have been used (Ravichandran et al. 2012). LDSs with BoWs approach are implemented and the best recognition rate is 84% (Ravichandran et al. 2012). The recognition rate of using single LDS, 3D SIFT and pixel intensity series (100 codewords) is 59.78, 50 and 53.48% respectively. The recognition rate of EMD-based approach using pixel intensity series is 32.61%.

### 5.3 UCLA-9 dataset

Figure 13 shows the confusion matrix for spatial-temporal feature approach on the UCLA-9 dataset corresponding to the recognition rate 35.3%. Figures 14 and 15 show the confusion matrix for BoWs approach and EMD-based approach on the UCLA-9 dataset corresponding to the recognition rate 83.3 and 85.1% respectively. LDSs with BoWs approach are implemented and the best recognition rate is 78% (Ravichandran et al. 2012). The recognition rate of using single LDS,

3D SIFT and pixel intensity series (100 codewords) is 67.5, 69 and 73.2% respectively. The recognition rate of EMD-based approach using pixel intensity series is 14.2%.

The confusion matrixes show the confusion between “fountain” and between “waterfall” (in Fig. 11), “fountain” and between “water” (in Fig. 12), and between “flowers” and “plant” (in Figs. 14, 15). This is consistent with our intuition that similar DTs are more easily confused with each other.

In Figs. 11, 12, 14 and 15, the results show that the proposed chaotic feature vector achieves high performances. In the BoWs approach, the results of chaotic feature vector on the two datasets are better than that in Ravichandran et al. (2012). In the same EMD-based approach, the performance of chaotic feature vector is better than the performance of spatial temporal feature. In Fig. 10, several categories such as boiling water, fire and fountain show higher recognition results while the rest give higher error rate. The reason is that the spatial–temporal feature response strongly to motion

information and the fire class provides the motion characteristics.

### 5.4 NewDT-10 dataset

Figure 16 shows the confusion matrix for spatial–temporal feature approach on the newDT-10 dataset using 100 codewords corresponding to the recognition rate 41.39%. Figures 17 and 18 show the confusion matrix for BoWs approach and EMD-based approach on the newDT-10 dataset corresponding to the recognition rate 74.14 and 75.93% respectively. The recognition rate of using single LDS, 3D SIFT and pixel intensity series (100 codewords) is 67.13, 65 and 68.77% respectively. The recognition rate of EMD-based approach using pixel intensity series is 14.2%. The recognition rate of EMD-based approach using pixel intensity series is 21.48%.

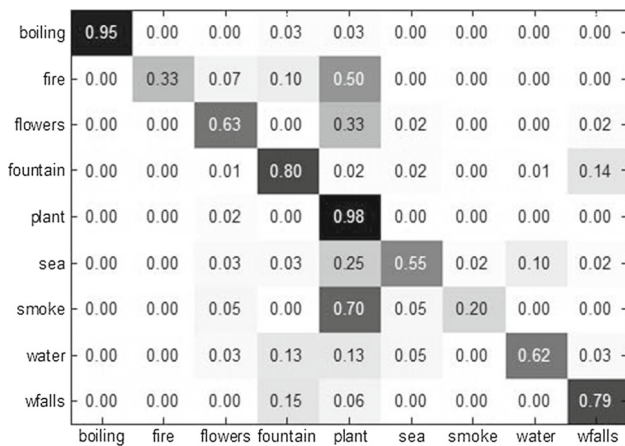


Fig. 14 Confusion matrix of BoWs approach on UCLA9 dataset. The overall recognition performance is 83.3%

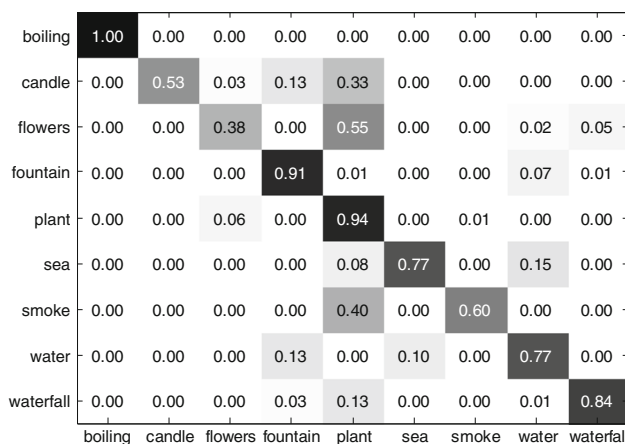


Fig. 15 Confusion matrix of EMD-based approach on UCLA9 dataset. The overall recognition performance is 85.1%

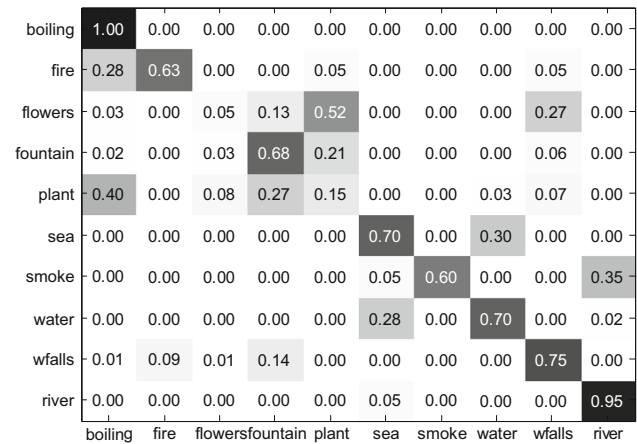


Fig. 16 Confusion matrix of spatial–temporal feature approach on newDT-10 dataset. The overall recognition performance is 41.39%

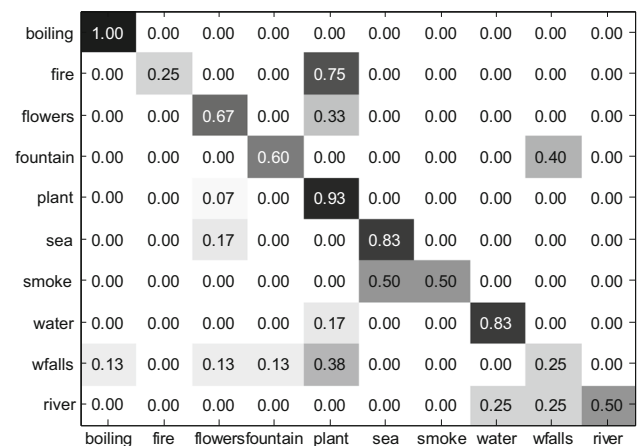
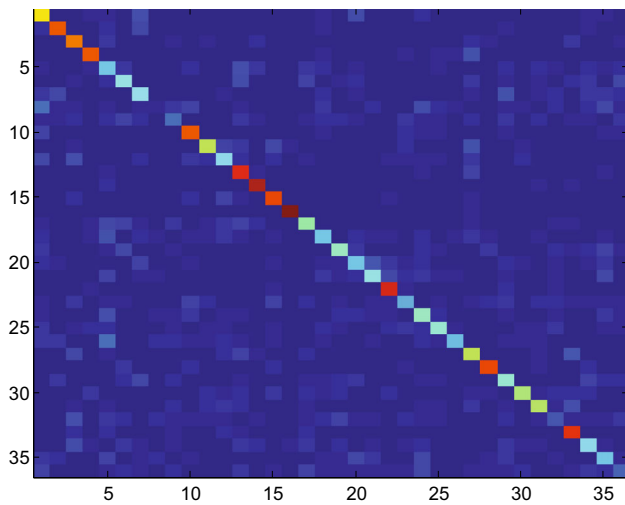


Fig. 17 Confusion matrix of BoWs approach on newDT-10 dataset. The overall recognition performance is 74.14%

boiling	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
fire	0.00	0.25	0.00	0.00	0.25	0.00	0.00	0.00	0.50	0.00
flowers	0.00	0.00	0.83	0.00	0.17	0.00	0.00	0.00	0.00	0.00
fountain	0.00	0.10	0.00	0.30	0.10	0.00	0.00	0.00	0.50	0.00
plant	0.00	0.00	0.00	0.00	0.89	0.00	0.00	0.11	0.00	0.00
sea	0.00	0.00	0.00	0.00	0.67	0.33	0.00	0.00	0.00	0.00
smoke	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.50	0.00
water	0.00	0.00	0.00	0.00	0.33	0.00	0.00	0.50	0.17	0.00
wfalls	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
river	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
	boiling	fire	flowers	fountain	plant	sea	smoke	water	wfalls	river

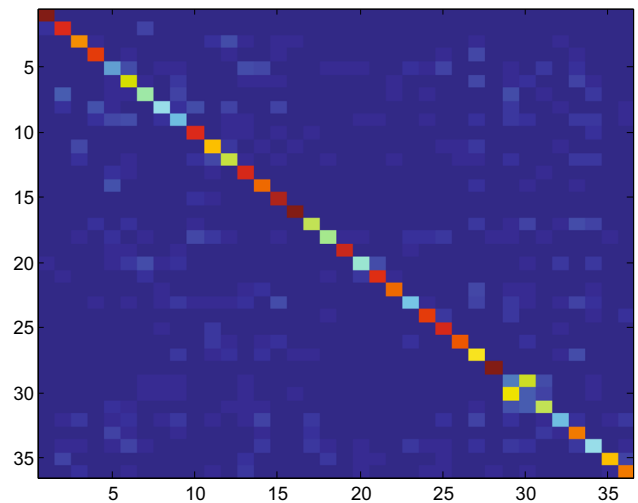
**Fig. 18** Confusion matrix of EMD approach on newDT-10 dataset. The overall recognition performance is 75.93 %



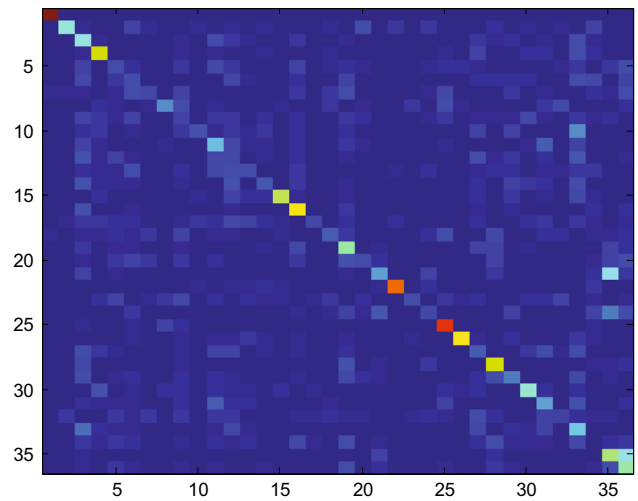
**Fig. 19** Confusion matrix of BoWs approach for pixel intensity series approach on DynTex++ dataset. The overall recognition performance is 49.67 %

5.5 DynTex++ dataset

Figure 19 shows the confusion matrix of BoWs approach for pixel intensity series approach on the DynTex++ dataset corresponding to the recognition rate 49.67 %. Figure 20 shows the confusion matrix of BoWs approach for chaotic feature vector approach on the DynTex++ dataset corresponding to the recognition rate 64.22 %. Figure 21 shows the confusion matrix of EMD approach for pixel intensity series approach on the DynTex++ dataset corresponding to the recognition rate 32.39 %. Figure 22 shows the confusion matrix of EMD approach for chaotic feature vector approach on the DynTex++ dataset corresponding to the recognition rate 59.33 %. The recognition rate of using single LDS is 47.2 %. The best performance in Ghanem and Ahuja (2010) is 63.7 %.



**Fig. 20** Confusion matrix of BoWs approach for chaotic feature vector approach on DynTex++ dataset. The overall recognition performance is 64.22 %

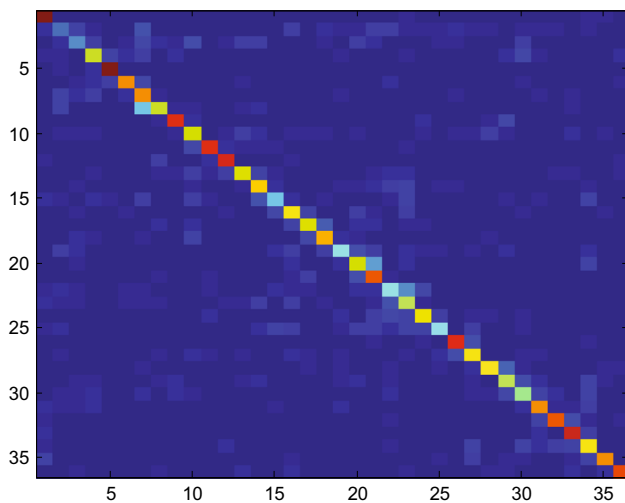


**Fig. 21** Confusion matrix of EMD approach for pixel intensity series approach on DynTex++ dataset. The overall recognition performance is 32.39 %

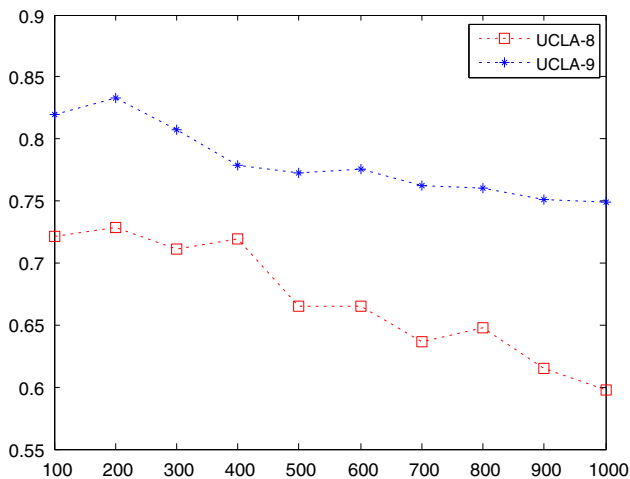
5.6 Codebook size

The purpose of this experiment is to validate the effect of different codebook size towards DTs recognition. Figure 23 shows the results obtained by different codebook size using BoWs approach on UCLA-8 dataset and UCLA-9 dataset, respectively.

In Fig. 23, the recognition rate is between 60 and 85 %. It shows some dependency of the recognition accuracy on the size of the codebook. The experiment coincides our conjecture that the recognition rate varies with the codebook size.



**Fig. 22** Confusion matrix of EMD approach for chaotic feature vector approach on DynTex++ dataset. The overall recognition performance is 59.33 %



**Fig. 23** Recognition performance on UCLA-8 dataset and UCLA-9 dataset using different codebook size

### 5.7 Discussion

A few interesting observations can be made from the experimental results: Our proposed chaotic feature vector-based approach significantly outperforms the traditional LDSs-based method. Comparing results of the LDSs-based BoWs approach with the results in [Ravichandran et al. \(2012\)](#) on the UCLA-8 and UCLA-9 datasets, our proposed chaotic feature vector-based methods show more than 5 % improvement over its LDSs-based counterpart ([Ravichandran et al. 2012](#)). This can be attributed to the fact that chaotic feature vector being based on chaotic dynamical system depicts fractal dimension of pixel intensity series. The fractal dimension is known to contain useful information for texture modeling.

In the first three datasets, the EMD-based approach produces better recognition results compared to the BoWs approach. This coincides with the former statement that in the EMD-based approach the signature can be more descriptive to summarize the distribution of chaotic feature vectors than histogram. We advocate the use of EMD-based approach because along with superior results it also offers ways to other DTs applications (e.g., DTs segmentation). In the DynTex++ dataset, the BoWs-based approach performs better than the EMD-based method. This can be attributed to the fact that the content of the dataset is simple. Therefore, the mean shift clustering algorithm cannot obtain significant signatures (foreground).

Traditional DTs recognition methods such as LDSs based have been studied and perfected for at least a decade, while our method is built on new techniques that have not previously been applied to DTs analysis. We believe that “mature” methods such as LDSs have been pushed close to the intrinsic limit of their performance, while novel methods such as ours have a much greater potential for improvement in the future.

## 6 Conclusions

We test our algorithm on four DTs datasets. The performances for the four datasets support our conjecture that the proposed approach is appropriate for DTs modeling and recognition. First chaotic features are extracted from each pixel intensity series and concatenated to a chaotic feature vector. Each video is represented by a feature vector matrix. Two recognition schemes are adopted. Following the BoWs approach, we show the histogram of each category. Another scheme is first clustering chaotic feature vectors in each feature vector matrix. Then, EMD method is employed to measure the similarity between two videos. The matching score is used as a kernel for recognition. Utilizing the proposed recognition framework, we have achieved very competitive performances on four diverse datasets. Based on our experiments, we observe that in most of the cases, EMD-based approach is better than BoWs approach using the 1-NN classifier.

Future work includes testing more DTs datasets and investigating how to fuse our proposed features with other features.

## References

- Ali S, Basharat A, Shah M (2007) Chaotic invariants for human action recognition. In: IEEE international conference on computer vision, 2007
- Bar-Joseph Z, El-Yaniv R, Lischinski D, Werman M (2001) Texture mixing and texture movie synthesis using statistical learning. *IEEE Trans Vis Comput Graph* 7(2):120–135

- Chan AB, Vasconcelos N (2005) Mixtures of dynamic textures. *IEEE Int Conf Comput Vis* 1:641–647
- Chan AB, Vasconcelos N, Probabilistic kernels for the classification of auto-regressive visual processes. In: *IEEE conference on computer vision and pattern recognition (CVPR)*, 2005
- Chan AB, Vasconcelos N (2007) Classifying video with kernel dynamic textures. In: *IEEE conference on computer vision and pattern recognition (CVPR)*, June 2007, Minneapolis
- Chen N, Prasanna VK (2013) A bag-of-semantic model for image clustering. *Vis Comput* 29(11):1221–1229
- Chetverikov D, Pteri R (2005) A brief survey of dynamic texture description and recognition. In: *Proceedings of 4th international conference on computer recognition systems, Poland*, pp 17–26, 2005
- Chaudhuri BB, Sakar N (1995) Texture segmentation using fractal dimension. *IEEE Trans Pattern Anal Mach Intell (PAMI)* 17:72–77
- Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. *IEEE Trans Pattern Anal Mach Intell (PAMI)* 24(5):603–619
- Cock KD, Moor BD (2000) Subspace angles between linear stochastic models. In: *IEEE conference on decision and control*. Proceedings, December, pp 1561–C1566
- Dantzig GB (1951) Application of the simplex method to a transportation problem. In: *Activity analysis of production and allocation*. Wiley, pp 359–373
- Dollar P, Rabaud V, Cottrell G, Belongie S (2005) Behavior recognition via sparse spatio-temporal features. In: *VS-PETS*, 2005
- Doretto G, Chiuso A, Wu YN, Soatto S (2003) Dynamic texture. *Int J Comput Vis* 51(2):91–109
- Fazekas S, Chetverikov D (2005) Normal versus complete flow in dynamic texture recognition: a comparative study. *Texture (2005) 4th international workshop on texture analysis and synthesis*. Beijing, pp 37–42
- Fei-Fei L, Perona P (2005) A Bayesian hierarchical model for learning natural scene categories. In: *IEEE conference on computer vision and pattern recognition (CVPR)*, vol 2, pp 524–531, June 2005
- Field DJ (1987) Relations between the statistics of natural images and the response properties of cortical cells. *J Opt Soc Am A* 4:2379–2394
- Fitzgibbon A (2001) Stochastic rigidity: image registration for nowhere-static scenes. *IEEE Int Conf Comput Vis* 1:662–669
- Fournier A, Reeves W (1986) A simple model of ocean waves. In: *Proceedings of ACM SIGGRAPH*, pp 75–84, 1986
- Fraser AM et al (1986) Independent coordinates for strange attractors from mutual information. *Phys Rev* 33(2):1134–1140
- Ghanem B, Ahuja N (2010) Maximum margin distance learning for dynamic texture recognition. *ECCV*, pp 223–236
- Grassberger P, Procaccia I (1983) Characterization of strange attractors. *Phys Rev Lett* 50(5):346–349
- Kantz H, Schreiber T (1997) *Nonlinear time series analysis*. Cambridge University Press, Cambridge
- Kennel MB et al (1992) Determining embedding dimension for phase space reconstruction using a geometrical construction. *Phys Rev A* 45(6):3403–3411
- Laptev I, Lindeberg T (2003) Space time interest points. In: *IEEE international conference on computer vision*, 2003
- Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: *IEEE conference on computer vision and pattern recognition (CVPR)*, 2006
- Pentland AP (1984) Fractal based description of natural scenes. *IEEE Trans Pattern Anal Mach Intell (PAMI)* 6(6):661–674
- Peteri R, Chetverikov D (2005) Flow dynamic texture recognition using normal, regularity texture. In: *Proceedings of Iberian conference on pattern recognition and image analysis (IbPRIA 2005)* Estoril, Portugal, pp 223–230
- Ravichandran A, Chaudhry R, Vidal R (2012) Categorizing dynamic textures using a bag of dynamical systems. *IEEE Trans Pattern Anal Mach Intell (PAMI)* 35(2):342–353
- Rubner Y, Tomasi C, Guibas LJ (1998) A metric for distributions with applications to image databases. In: *IEEE international conference on computer vision*, pp 59–66, January 1998
- Saisan P, Doretto G, Wu YN, Soatto S (2001) Dynamic texture recognition. In: *IEEE conference on computer vision and pattern recognition (CVPR)*, vol 2, pp 58–63
- Scovanner P, Ali S, Shah M (2007) A 3-dimensional sift descriptor and its application to action recognition. In: *Proceedings of the 15th international conference on Multimedia*. ACM, pp 357–360
- Shroff N, Turaga P, Chellappa R (2010) Moving vistas: exploiting motion for describing scenes. In: *IEEE conference on computer vision and pattern recognition (CVPR)*, June 2010
- Sivic J, Zisserman A (2003) Video Google: a text retrieval approach to object matching. In: *Videos proceedings of the international conference on computer vision (2003)*
- Szumner M, Picard RW (1996) Temporal texture modeling. In: *Proceedings of the international conference on image processing*, vol 3
- Taken F (1981) Detecting strange attractors in turbulence. In: Rand DA, Young LS (eds) *Lecture notes in mathematics*. Springer, Berlin, Heidelberg
- Wang Y, Zhu S (2003) Modeling textured motion: particle, wave and sketch. In: *IEEE international conference on computer vision*, pp 213–220, 2003
- Wu S, Moore B, Shah M (2010) Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes. In: *IEEE conference on computer vision and pattern recognition (CVPR)*, 2010