Exploiting Chaotic Feature Vector for Dynamic Textures Recognition

Yong Wang[1] and Shiqiang Hu[1]
[1] School of Aeronautics and Astronautics, Shanghai Jiao Tong University
Shanghai, 200240 - China
[e-mail: wysjtu2008@gmail.com]
[e-mail: sqhu@sjtu.edu.cn]
*Corresponding author: Shiqiang Hu

Received May 14, 2014; revised July 27, 2014; revised September 4, 2014; accepted September 10, 2014; published November 30, 2014

Abstract

This paper investigates the description ability of chaotic feature vector to dynamic textures. First a chaotic feature and other features are calculated from each pixel intensity series. Then these features are combined to a chaotic feature vector. Therefore a video is modeled as a feature vector matrix. Next by the aid of bag of words framework, we explore the representation ability of the proposed chaotic feature vector. Finally we investigate recognition rate between different combinations of chaotic features. Experimental results show the merit of chaotic feature vector for pixel intensity series representation.

Keywords: Chaotic feature vector, pixel intensity series, bag of words, dynamic textures recognition

http://dx.doi.org/10.3837/tiis.2014.11.027

1. Introduction

Dynamic textures (DTs) are sequences of images of moving scenes that exhibit certain stationary properties in time [1], such as smoke floating in the wind, boiling water and so on. Great attention has been paid since the potential application of DTs, e.g. remote surveillance of natural disaster.

Fig. 1(a) illustrates two types of DTs, water and sea. From **Fig. 1(a)** people can not figure out the size of the textures is 10 square cm large or 10 square m, without something or some tool next to it. This is usually called self-similarity. That is, the objects have the same structure at all scales. The natural scenes such as coastline possess the common characteristic of self-similarity. **Fig. 1 (b)** illustrates one pixel intensity series in the DT of sea and the position is (5, 5). The x-axis is the frame number and the y-axis is the value of pixel intensity. Pixel intensity as an image feature has been achieved great success [2, 3, 28, 29]. DTs are image sequences, thus pixel intensity series should be paid attention to. The popular methods in DTs recognition is modeling a DT as linear dynamic systems (LDSs) [1]. In DTs analysis, the self-similarity property leads each pixel intensity series possessing fractal property. Stationary property indicates that similarity exists among pixel intensity series. Chaos theory is developed to solve the fractal problems and is introduced in computer vision recently [4, 5, 6]. Chaotic features can be extracted from time series and have achieved great success in action recognition, dynamic scene recognition and anomaly detection. Motivated by these work, we explore the representation ability of chaotic feature vecotr in DTs.

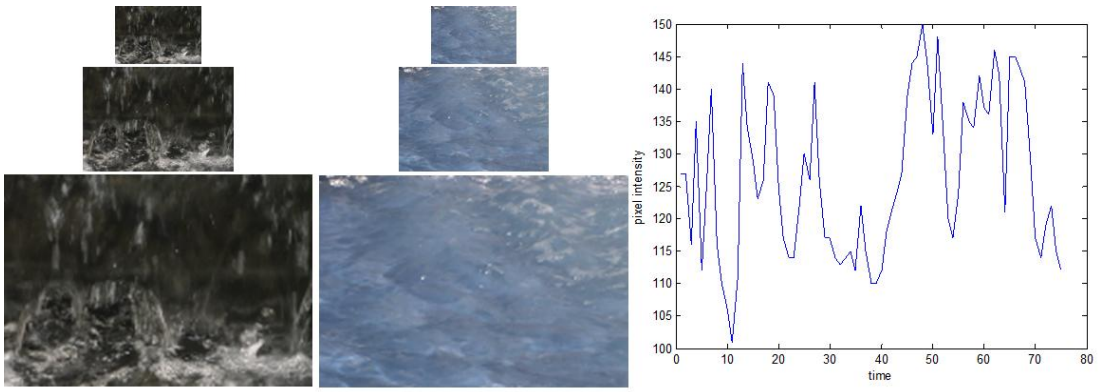

Fig. 1. One frame from DTs

In this paper, each pixel intensity series is used to compute the chaotic feature and other features. These sets of features are then combined to a feature vector, named chaotic feature vector. The chaotic feature vector is used to represent the pixel intensity series in video. A DT video can be represented by a feature vector matrix. Then we use the well-known bag of words (BoWs) approach which has been adopted by many computer vision researchers [7, 8]. A codebook is learned by clustering all the chaotic feature vectors in the training 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.

The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 describes the chaotic feature vector used in our paper. Section 4 demonstrates the framework of our approach in the analysis of DTs recognition. Experimental results are presented in section 5. Section 6 contains conclusions.

2. Related Work

There exists a rich literature for DTs recognition. Traditional image based approach based on frame-to-frame estimation to extract optical flow features [9, 10] which are computation efficient and natural way to depict the local DTs. The main drawback of this approach is the flow features (e.g. optical flow) are computed based on assumption of local smoothness and brightness constancy. The non-smoothness, discontinuities DTs are difficult to process.

Recently, the DTs are modeled as LDSs in many work [1, 11, 12-14]. LDSs are learned by system identification to model DTs and classified 50 different DTs [1]. 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 [12]. Then LDSs model is extended with a nonlinear observation to recognize DTs [13]. A probabilistic kernel is derived to describe the spatial-temporal process [14]. BoWs approach is used in [11] to model each DT video with LDSs. The key idea in using BoWs approach is codebook-based modeling of videos and each video can be considered as a bag containing some codewords from the codebook. However, DTs are generated by complex time varying dynamical systems, the LDSs model is constraint by linearly assumption that makes it restrictive for modeling DTs.

Pixel intensity series is a nonlinear time series. Chaos theory is developed to deal with nonlinear systems [15]. To characterize the structure of pixel intensity series, it is necessary to reconstruct a phase space. The process of reconstructing the phase space is commonly referred to as embedding. Chaotic features capture the structure of the time series and are invariant under phase space reconstruction. Recently, several chaotic features, including Largest Lyapunov exponent (LLE), correlation dimension and correlation integral have been used to represent time series for recognition purpose. M. Perc [16] analyzes recording of human gait which is used to obtain reconstructed phase space and the LLE is calculated. The results indicate human gait possesses properties of chaotic systems. Trajectories from six landmarks (two hands, two feet, the head, and the body center) on human body are molded in [4] to reconstruct phase space. Each trajectory is then used to compute chaotic features that include Lyapunov exponent, correlation integral and correlation dimension. Chaotic features are combined to a feature vector for dynamic scene recognition [5]. Trajectories are treated as time series in [6]. Chaotic features are calculated to detect and locate anomalies. The successes of the work mentioned above motivate us to explore the representation of chaotic feature vector to DTs.

The aim of this paper is to derive a representation of the DTs from the chaotic feature vector. This is achieved by using the concepts from chaos theory to model and analyze nonlinear dynamics of pixel intensity series. Different from the chaotic features mentioned above, fractal dimension which measures the self-similarity properties of a time series is used in our work. It has been used in image processing for decades [17]. Many natural senses can be modeled by fractal dimension and linear log power spectrum that is related to the fractal dimension exists in natural texture [18, 19]. A modified box counting dimension approach [17] has been

proposed to estimate fractal dimension for image segmentation.

All the previous work suggests that improvement can be made by accurately modeling of pixel intensity series. Thus, we are interested in exploring the use of chaotic feature vector. We present our proposed algorithm in the following section.

3. Chaotic Feature Vector

In this section we present the background material related to the chaos theory. Pixel intensity series is a basic element of composing DTs. Hundreds and thousands of pixel intensity series make up DTs. The similarity among pixel intensity series leads DTs to self-similarity in each scale.

3.1 embedding theory

Embedding is a mapping from one dimensional space to an m-dimensional space. According to Taken's theorem [20] a map exists between the original time series $x(t) = [x_1(t), x_2(t), \cdots, x_n(t)] \in R^n$ and a τ embedding time delay version of $x(t)$, i.e. the vector $[x_0 \quad x_\tau \quad \cdots \quad x_{(m-1)\tau}]$. These vectors are called phase-variable. Here τ is embedding time delay [21] and m is embedding dimension [22].

The embedding delay τ is computed by mutual information algorithm [21]. First, the range of the time series $[\min(x_t), \max(x_t)]$ is divided into equal bins.

$$I(\tau) = \sum_{s=1}^{b} \sum_{q=1}^{b} P_{s,q}(\tau) \log \frac{P_{s,q}(\tau)}{P_s(\tau) P_q(\tau)} \tag{1}$$

where P_s and P_q denotes the probabilities that the variable x_t assumes a value inside the sth and qth bin respectively, and $P_{s,q}$ is the joint probability that x_t is in bin s and $x_{t+\tau}$ is in bin q. The first local minimum of $I(\tau)$ is chosen as the embedding delay.

The embedding dimension d is computed by false nearest neighbors [22]. The idea of the algorithm is if points are sufficiently close in a reconstructed phase space, then they should remain close in the next.

Given a one dimensional time series, $x(t) = [x_1(t), x_2(t), \cdots, x_n(t)]$ for an appropriate embedding dimension m and embedding time delay τ, the time series $x(t)$ can be transform to the m-dimensional space.

$$X = \begin{pmatrix} x_0 & x_\tau & \cdots & x_{(m-1)\tau} \\ x_1 & x_{\tau+1} & \cdots & x_{(m-1)\tau+1} \\ x_2 & x_{\tau+2} & \cdots & x_{(m-1)\tau+2} \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix} \tag{2}$$

3.2 Chaotic Feature

Chaotic features are measures that quantify the properties that are invariant under transformations of the state space.

3.2.1 Information dimension:

The information dimension [15] is one of fractal dimension specifies information scales $I(\varepsilon)$ with the radius ε, which defined as

$$D_i = \lim_{\epsilon \to 0} \frac{I(\varepsilon)}{\ln \varepsilon}, \tag{3}$$

Information dimension shows the inner structure of time series in each scale. When the one dimensional pixel intensity series transformed to an m dimensional phase space, the information dimension can be used to measure the smoothness of the transformed phase space.

The smooth the phase space, the smaller the information dimension is.

3.3. Chaotic Feature Vector

Embedding time delay and embedding dimension are two important parameters to determine the geometry information in the phase space reconstruction. Information dimension depict the fractal information of the pixel intensity series. Mean of pixel intensity series encodes the value of the pixel intensity series that is important for recognition in the video. Thus, the mean value of pixel intensity series is included with chaotic feature in our chaotic feature vector, $F=\{\tau, m, D_i, ,mean\}$.

Given a *W*L*T* sequence, *W*, *L* and *T* are the width, length and time dimension of the sequence respectively. The chaotic feature vector of each pixel intensity series are extracted and the video is represented by a *W*L*4* dimensional feature vector matrix.

4. System workflow

In order to give the quantitate result, the BoWs framework is employed. **Fig. 2** illustrates flowchart of our work.

In the BoWs representation, to learn the vocabulary of codewords, we consider all chaotic feature vector in the training data. The codebook is constructed by clustering using the k-means algorithm and Euclidean distance as the clustering metric. The center of each cluster is defined to be a codeword. Thus, each chaotic feature vector can be assigned a unique cluster membership, i.e., a codeword. Then, a video is encoded as a histogram of the number of occurrences of codewords count according to

$$h(d) = \left(h_i(d)\right)_{i=1\ldots N}, \text{ with } h_i(d) = n(d, v_i) \tag{4}$$

where $n(d, v_i)$ denotes the number of occurrences of chaotic feature vector v_i in video d. The effect of the codebook size is explored in our experiments, and the results are shown in **Fig. 10**.

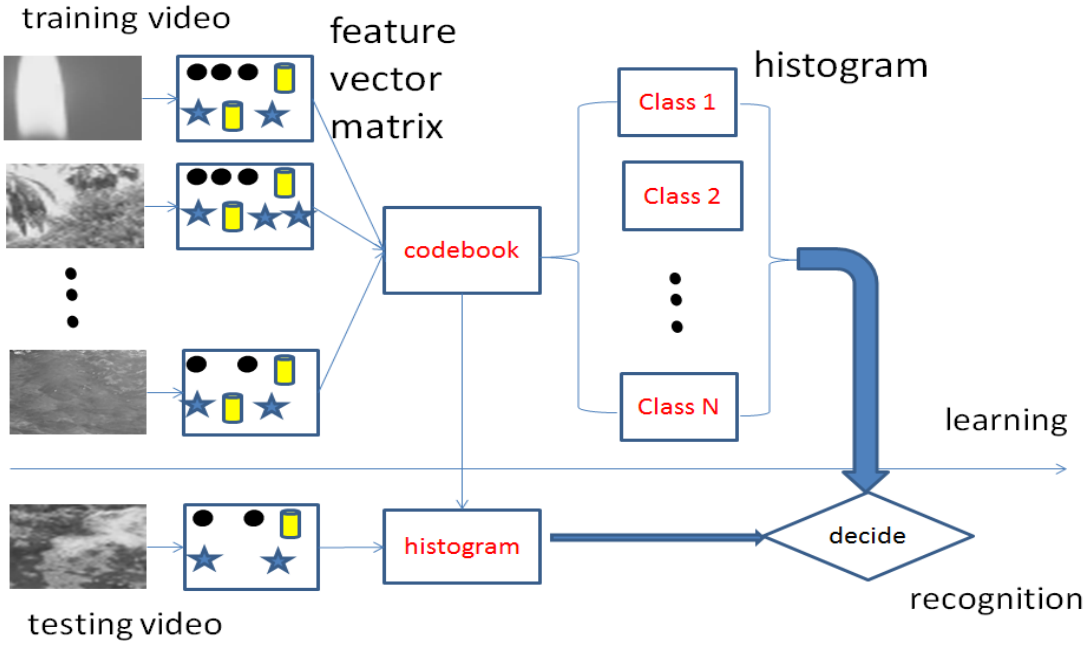

Fig. 2. Flowchart of the proposed algorithm

BoWs recognition framework is composed by two parts, training stage and testing stage. In training stage, all training feature vector matrix are clustered by k-means clustering (k = 100, 200, …, 1000) algorithm with Euclidean metric and obtain the cluster centers, which form our histogram bins. The horizontal axis is the cluster center and the vertical axis is the number of occurrences of the chaotic feature vector. The number of the clusters is the codebook size. After generation of the codebook, each 4-attribute chaotic feature vector of a DT 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 DT are mapped to the cluster centers, the DT video can be represented by a histogram of the codewords. The goal of the learning step is to achieve a model that best represents the distribution of these codewords in each category of DTs.

In the testing stage, a feature vector matrix of an unknown video is first represented by a histogram of the codewords follow the steps mentioned above. Then the category model is found that fits best the distribution of the codewords of the unknown DTs video.

5. Experiment

In this section, an evaluation of the proposed method is proposed on two diverse datasets: newDT-10 dataset and DynTex++ dataset. In addition, the performances of different chaotic features combination recognition are compared. The goal of these experiments is to determine the representation ability of our proposed chaotic feature vector and the presence or absence of the effect of the features in the recognition experiment.

5.1 Implementation detail

Data set:

We collect 16 river videos with smooth shaking with 75 frames and the dimension is reduced to 48*48 and combined with UCLA dataset [1, 11]. The dataset is classified to 10 classes: boiling water (8), fire (8), flowers (12), fountains (20), plants (108), sea (12), smoke (4), water (12), waterfall (16) and river (16), where the numbers denote the number of video sequences in the dataset. This dataset is used to test the robustness of our algorithm when DTs are taken under different viewpoints, scales and other unconstraint environment. The 10 classes dataset is called newDT-10 dataset.

The second dataset is DynTex++ dataset [23] which contains 36 categories of different DTs and 100 in each category. In this dataset, there contains a total of 3600 videos which provides a richer benchmark.

Fig. 3 shows examples from the newDT-10 dataset and DynTex++ dataset.

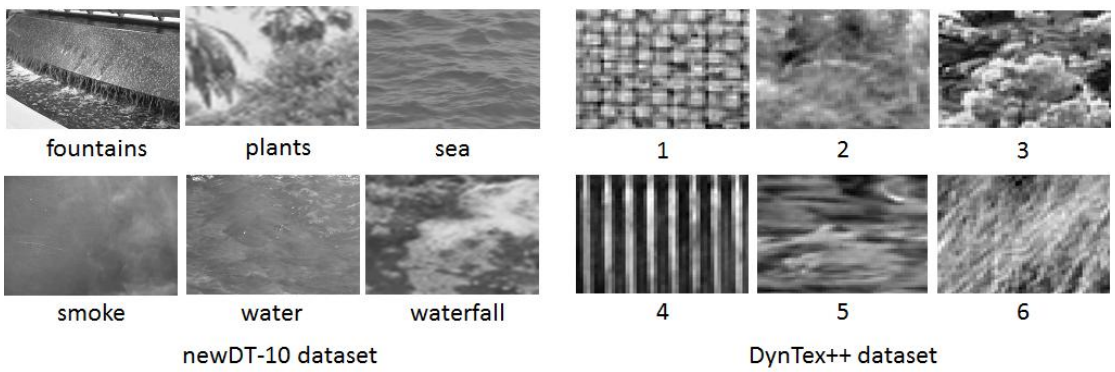

Fig. 3. Examples from the newDT-10 dataset and DynTex++ dataset.

Fig. 4 shows the path of pixel intensity series which are from different classes of newDT-10 dataset evolve over time. Here, the embedding dimension is set to three for clearly show. That is, the transformed phase space is 3. Thus the **Fig. 4** is showed in 3 dimensions. The coordinates in the figure is the coordinates of the transformed phase space. The real reconstructed phase space is more complex than the Fig. 4 shows. It shows that the paths of pixel intensity series in each class of DT vary greatly. The information dimension is a measure to quantify the smoothness of the phase space. The parameters of computing the features are given below. In the mutual information algorithm, the number of partitions for the time series is set to 100. In the false nearest neighbor algorithm, the number of nearest neighbors to compute is set to 50. In computing information dimension, the ε is set to 0.2.

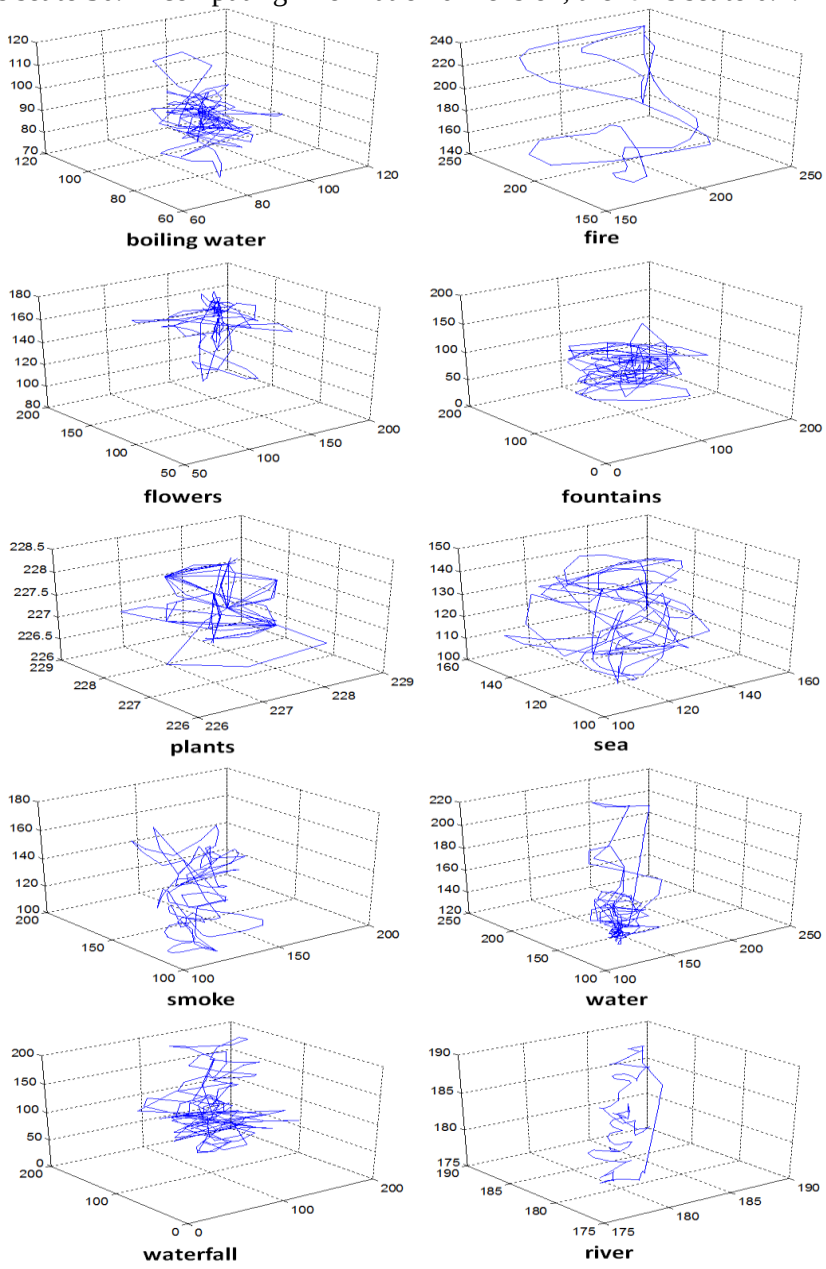

Fig. 4. System state evolve over time

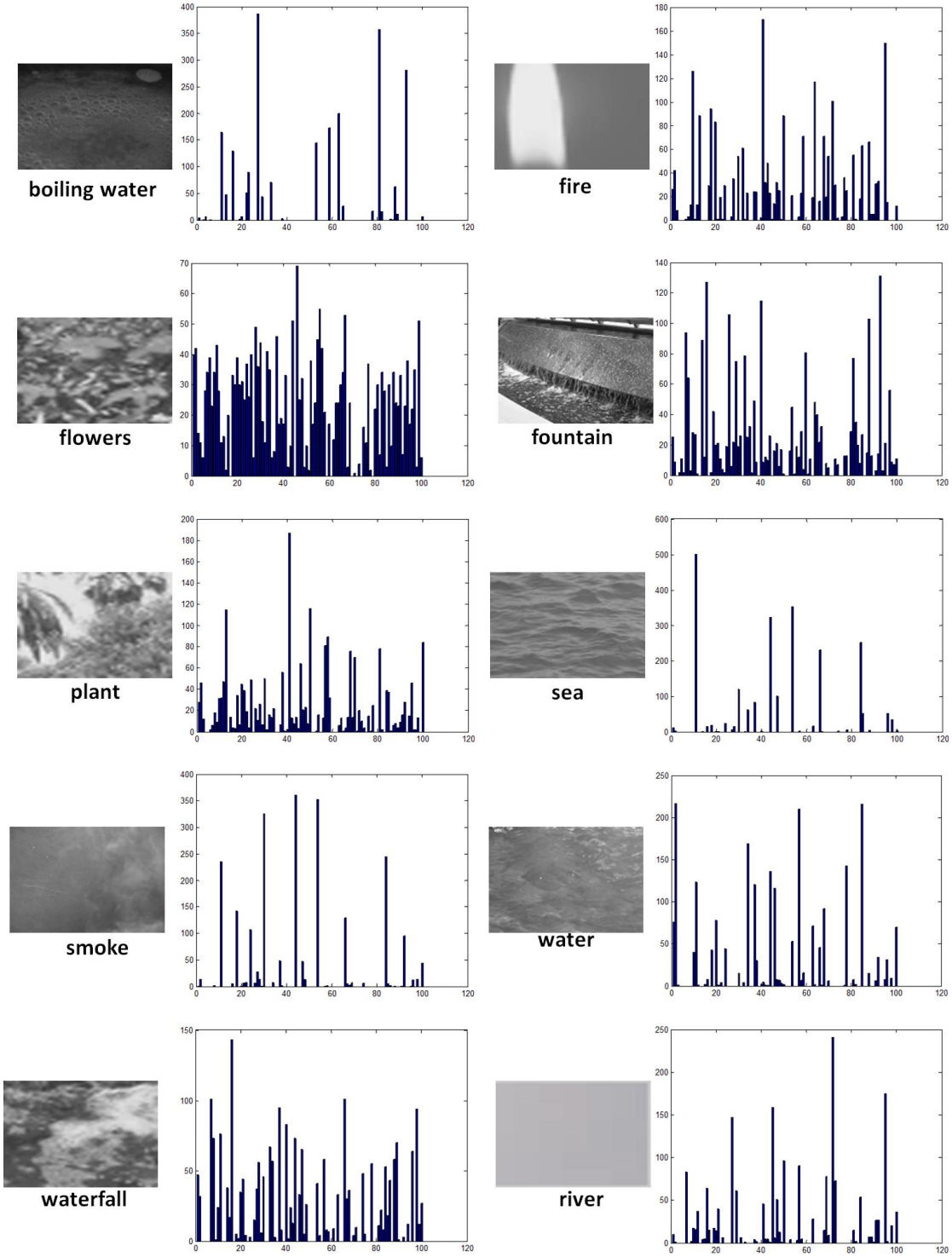

Fig. 5. Shows the histogram of chaotic feature vector in codebook size k=100.

The generation of histograms using BoWs approach is shown in **Fig. 5**. It shows ten examples of testing videos from each category with their corresponding DTs histograms to demonstrate discrimination of the distribution of the learned DTs codewords. It is clear to see

DTs from each categories have different dominate peaks. For example, the peak in the boiling water is in the index of 30 and 80 while the peak in the fire is in the index of 40 and 100. Meanwhile different categories have some overlap bins. For example, the peak in fountain is around 20, 40, 60 and 80 and so is waterfall. That is the reason of confusing different classes.

Recognition method：
Nearest neighbor (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.

5.2 newDT-10 dataset

We compare the performance of our approach with two baselines: single LDS approach and spatial temporal feature approach. Traditional methods develop features in image analysis (SIFT, Harris, GIST and so on) to three dimensional features in video analysis by adding time dimension in the features. Therefore, SIFT3D [32], Harris3D [31], spatial temporal feature [25], spacetime texture [30] and GIST3d are used in video analysis. We use spatial temporal feature as an example in our paper to compare the recognition results with our method. Spatial temporal feature is also used in [11] for a baseline method. In this paper, we use spatial temporal feature to represent these features mentioned above. These features based methods extract features between frames. Only information of two frames is considered. In this work, we treat pixel intensity series as an integral to obtain more information.

Linear dynamic system method is a classical method in DTs recognition. And many methods are based on it [11-14]. LDSs model the pixel between frames as a linear dynamic system. While our method consider pixel intensity series as an integral and chaotic feature is extracted. Since most of the methods are based on the two approaches (3d feature based and LDSs based), we use these two methods as baseline methods.

We briefly explain the two methods and give some implementation details.
Baseline methods:
Single LDS Approach [24]: In our first baseline method we model the entire DTs video using a single LDS. Given a testing DT video, we compute the Martin distance and Fisher distance between the testing LDS and each of the LDS models of the training set. Based on these distances, we use a NN classifier. As for the system order, we test all system orders in the range [2, 4, 6, 8], and consider the best results out of these as the single LDS baseline. This approach is identical to the one originally proposed in [24].
Spatial temporal feature [25]: Our second baseline method is BoWs approach. We extract spatial temporal features from DTs videos. And reduce the dimensionality of the feature vector to a 100-dimensional vector using PCA. We used the original code provided by the authors at http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html.

Fig. 6 shows the confusion matrix for pixel intensity series approach on the newDT-10 dataset corresponding to the recognition rate 68.77%. **Fig. 7** shows the confusion matrix for our proposed chaotic feature vector approach on the newDT-10 dataset corresponding to the recognition rate 90%. The number of codebook size is 100. Each row in the confusion matrix corresponds to the ground truth class, and each column corresponds to the assigned label.

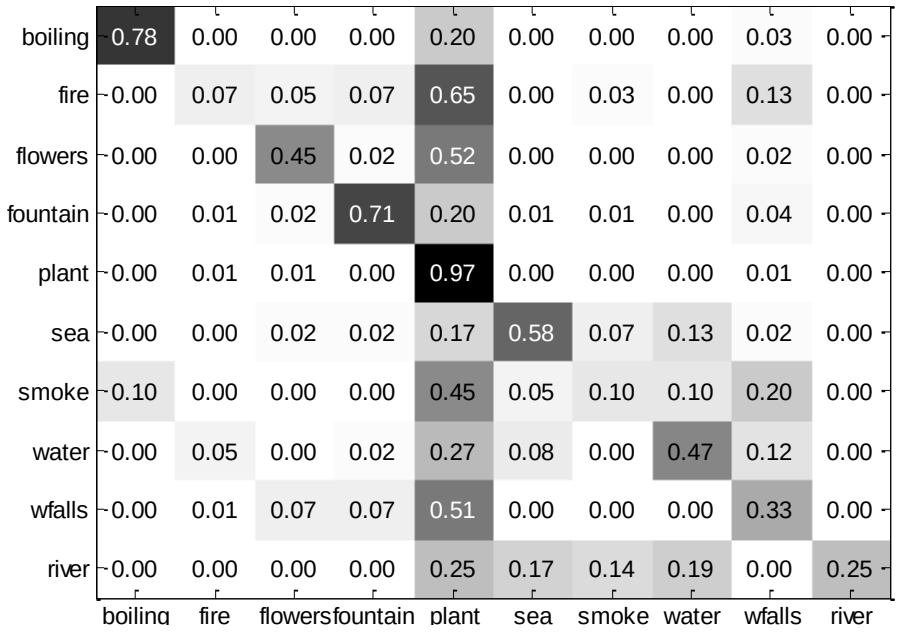

Fig. 6. Confusion matrix of pixel intensity series approach on newDT-10 dataset. The overall recognition performance is 68.77%.

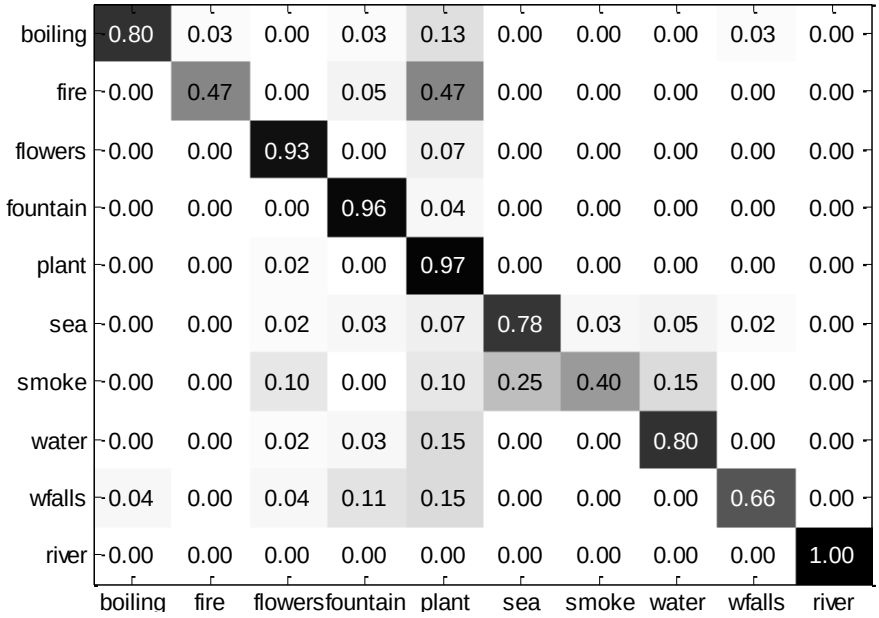

Fig. 7. Confusion matrix of our proposed chaotic feature vector on newDT-10 dataset. The overall recognition performance is 90%.

Boiling water class is misclassified to flowers, fountain and waterfall by using pixel intensity series. River class is misclassified to sea, smoke and water since the appearance is similar. The misclassification rate is much lower by using chaotic feature vector for boiling water class. For other categories, the recognition rate of using chaotic feature vector is higher than that of using pixel intensity series. The recognition rate of using single LDS and Spatial temporal feature is 63% and 78.33% respectively.

5.3 DynTex++ dataset

Single LDS approach [24] is employed as baseline method. And the parameter setting is the same as section 5.2.

Fig. 8 shows the confusion matrix for pixel intensity series approach on the DynTex++ dataset corresponding to the recognition rate 49.67%. **Fig. 9** shows the confusion matrix for our proposed chaotic feature vector approach on the DynTex++ dataset corresponding to the recognition rate 64.5%. The number of codebook size is 100. The recognition rate of using single LDS is 47.2%. The best performance in [23] is 63.7%.

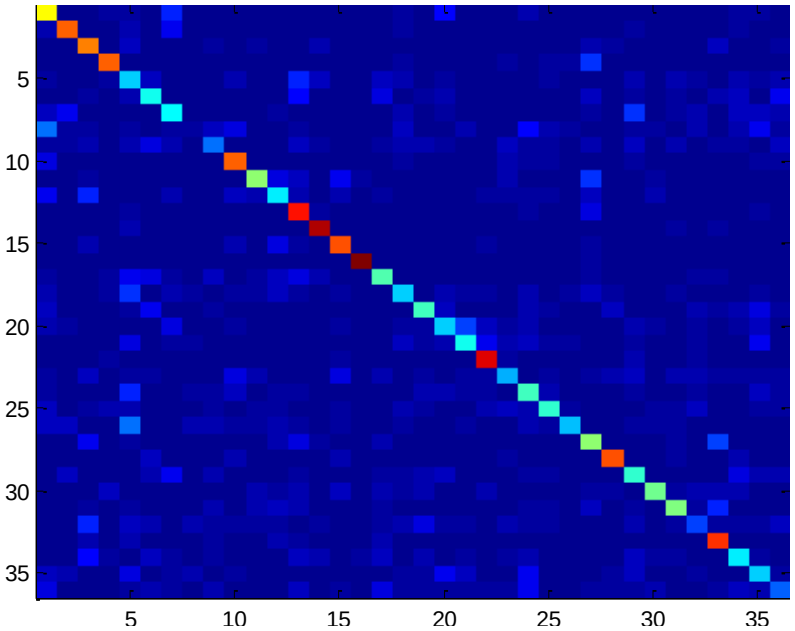

Fig. 8. Confusion matrix of pixel intensity series approach on DynTex++ dataset. The overall recognition performance is 49.67%.

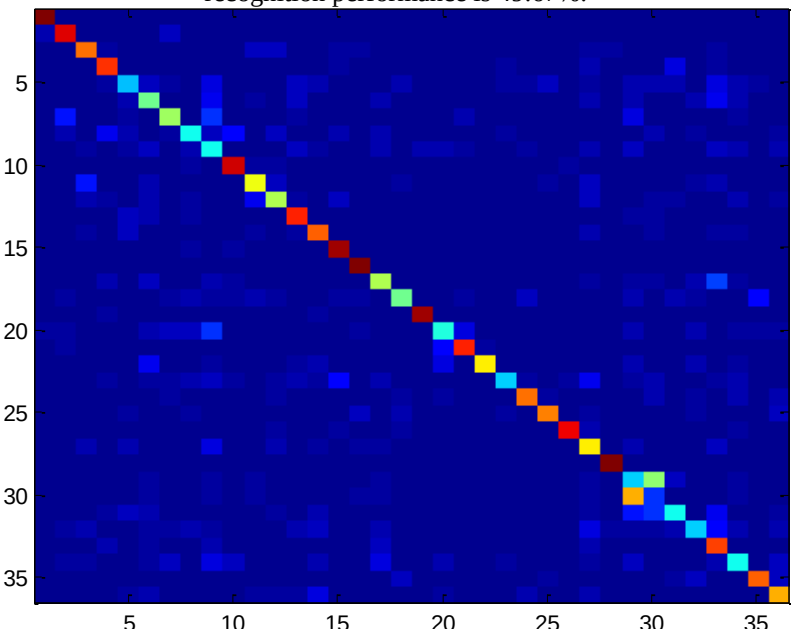

Fig. 9. Confusion matrix of our proposed chaotic feature vector on DynTex++ dataset. The overall recognition performance is 64.5%.

5.4 Codebook size

The number of codewords on recognition accuracy on newDT-10 dataset and DynTex++ dataset is illustrated in **Fig. 10**. The x axis is the number of codebook size and the y axis is the recognition rate. It shows some dependency of the recognition accuracy on the codebook size. And recognition accuracy is not increased as the increasing of the codebook size. ”CFV1” and “CFV2” stand for the recognition results of our proposed chaotic feature vector method for newDT-10 dataset and DynTex++ dataset respectively. ”PIS1” and “PIS2” denote the recognition results of pixel intensity series as feature for newDT-10 dataset and DynTex++ dataset respectively. In most of the time, the recognition rate of chaotic feature vector is higher than that of pixel intensity series.

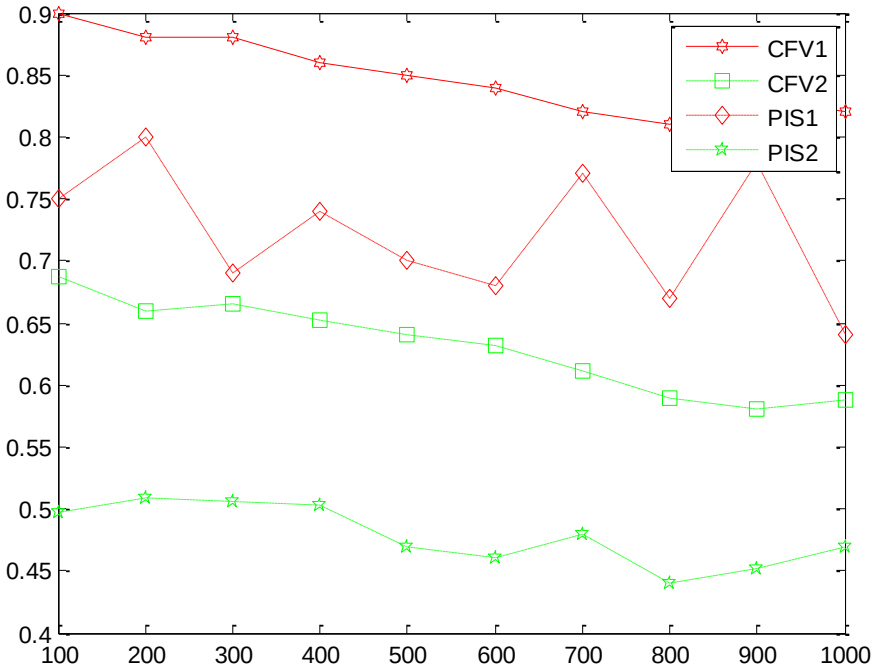

Fig. 10. Recognition performance on newDT-10 dataset and DynTex++ dataset using different codebook size.

5.5 Feature combination

In **Fig. 11**, the bar chart shows the best performance of the BoWs method for different combinations of features on newDT-10 dataset and DynTex++ dataset. The label on each bar corresponds to the feature vector used for the experiments. They are CFV = [embedding time delay, embedding dimension, information dimension, mean], FV2 = [embedding dimension, information dimension, mean], FV3 = [embedding time delay, information dimension, mean], FV4 = [embedding time delay, embedding dimension, mean], FV5 = [larget Lyapunov exponent, correlation integral, correlation dimension, variance], FV6 = [larget Lyapunov exponent, correlation dimension, mean], PIS = [pixel intensity series] respectively. We compare with the feature vector used in [4] and [6] as showed for label 5 and label 6 respectively.

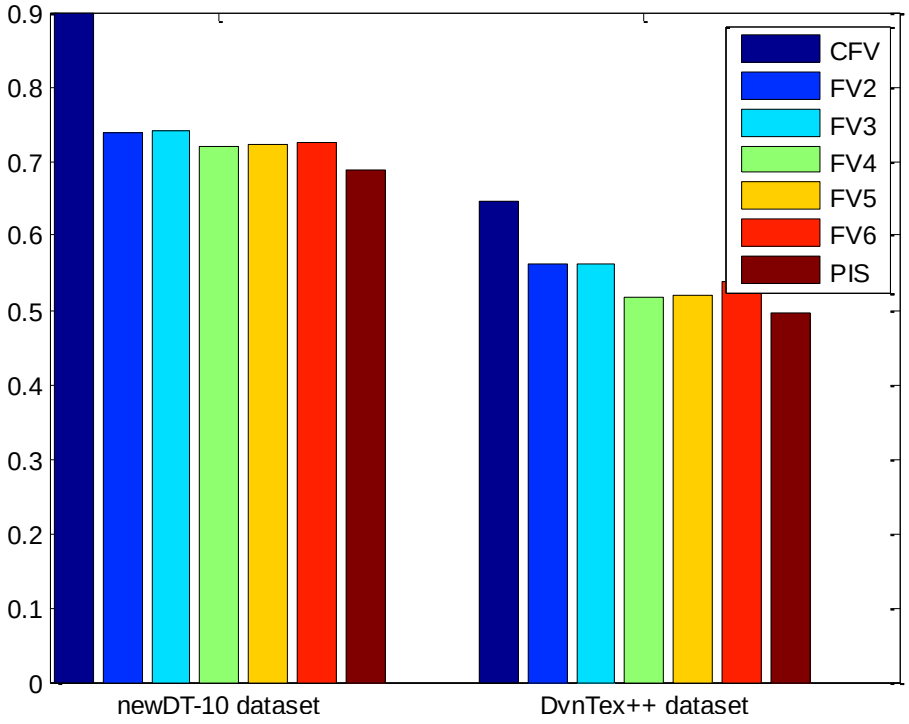

Fig. 11. Feature combination recognition results

Our proposed chaotic feature vector performs best in all the feature vectors. The best recognition result of using chaotic feature vector on newDT-10 dataset and DynTex++ dataset are 90% and 64.5% respectively. The best recognition result of using pixel intensity series on newDT-10 dataset and DynTex++ dataset are 68.77% and 49.67% respectively.

Most of the performances of feature combination are better than the pixel intensity series which indicate that chaotic feature vector is proper for DTs recognition. The recognition rates mainly belong to 70% to 80% and 50% to 60% for newDT-10 dataset and DynTex++ dataset respectively. Embedding time delay and embedding dimension which are important features for the structure of pixel intensity series. Label 2 and label 3 show that the recognition rates drop dramatically if we delete embedding time delay and embedding dimension. Label 1, label 5 and label 6 show that our proposed chaotic feature vector performs better than [4] and [23]. This indicates that the information dimension captures the self-similarity of pixel intensity series.

5.6 discussions

A few interesting observations can be made from the experimental results:

Pixel intensity series is longer than chaotic feature vector. It is easy to over fit when the training data is not big enough. In pattern recognition research, data dimension reduction is the first step to extract features which is represented common features of that category, such as [26, 27]. In our work, chaotic feature vector is used to represent pixel intensity series. The length of the feature vector is reduced from 75 to 4. In this point of view, our work that uses chaotic feature vector is a similar way to dimension reduction.

Given two pixel intensity series that are similar in shape but one pixel intensity series lag a time to another one. If alignment is first implemented, the distance of the two pixel intensity

series can be smaller than a threshold. Otherwise, the distance between the two pixel intensity series is large. Our proposed chaotic feature vector encodes the shape and fractal properties of the pixel intensity series. The distance between the chaotic feature vectors of the two pixel intensity series is small. This is the advantage of the chaotic feature vector that is insensitive to the initial value of the pixel intensity series.

In feature combination experiment results, it shows that different chaotic features represent different properties of time series. Information dimension is more suitable than other chaotic features to capture the self-similarity of pixel intensity series. There are many works in image classification that based on BoWs framework [33, 34]. In future, we will use the techniques in our work to improve the DTs recognition rate.

Since our method based on pixel intensity series, the proposed approach is effective under the following conditions: first, the DT should always exist in the video and occupy the major area. Second, the location of the DT should be fixed. If the position of the DT changed rapidly, the pixel intensity series will change the structure. This work can be extended to DTs segmentation and DTs localization since the chaotic feature vector is an appropriate feature for each pixel intensity series.

6. Conclusions

The main contribution of this paper is the novel application of a chaotic feature vector representation of pixel time series to the field of DTs recognition. The chaotic feature vector based representation is shown to be effective and broadly applicable in a range of representative DTs. In empirical comparisons to alternative commonly employed features and state-of-the-art methods, the proposed approach has been shown to yield exceptionally strong performance in response to such challenges. We believe DTs in our work is an example and our work bridge the gap between chaos theory and engineering applications. There is more natural scenes that can be treated as a time series and represented by chaotic feature vector.

Acknowledgements

This work was partly supported by the National Natural Science Foundation of China "61374161"and "61074106". The authors would like to thank Dr. G. Doretto for sharing the datasets that were used in this paper.

References

[1] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, “Dynamic texture,” *International Journal of Computer Vision*, vol. 51, no. 2, pp. 91-109, 2003. Article (CrossRef Link)

[2] Rongrong Ji, Yue Gao, Richang Hong, Qiong Liu, Dacheng Tao, and Xuelong Li, “Spectral-Spatial Constraint Hyperspectral Image Classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 3, pp. 1811-1824, march 2013. Article (CrossRef Link)

[3] Rongrong Ji, Hongxun Yao, Qi Tian, Pengfei Xu, Xiaoshuai Sun, and Xianming Liu, “Context-Aware Semi-Local Feature Detector,” *ACM Transactions on Intelligent System and Technology* , vol. 3, no. 3, pp. 44-71, 2012. Article (CrossRef Link)

[4] S. Ali, A. Basharat, and M. Shah, “Chaotic invariants for human action recognition,” in *Proc. of IEEE International Conference on Computer Vision*, pp. 1-8, October 14-20, 2007.

[5] N. Shroff, P. Turaga, and R. Chellappa, “Moving Vistas: Exploiting Motion for Describing Scenes,” *in Proc. of IEEE conference on Computer Vision and Pattern Recognition*, pp. 1911-1918, June

13-18, 2010.
[6] S. Wu, B. Moore, and M. Shah, "Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2054-2060, June 13-18, 2010.
[7] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2169-2178, 2006. Article (CrossRef Link)
[8] Fei-Fei, L. and Perona, P, "A Bayesian hierarchical model for learning natural scene categories," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 524-531 June, 2005. Article (CrossRef Link)
[9] S.Fazekas, and D.Chetverikov, "Normal Versus Complete Flow in Dynamic Texture Recognition: A Comparative Study," in *Proc. of 4th International Workshop on Texture Analysis and Synthesis*, pp.37–42, 2005. Article (CrossRef Link)
[10] D.Chetverikov, and R.Péteri, "A Brief Survey of Dynamic Texture Description and Recognition," *in Proc. of 4th Int. Conference on Computer Recognition Systems*, Poland, pp.17-26, 2005. Article (CrossRef Link)
[11] A. Ravichandran, R. Chaudhry, and R. Vidal, "Categorizing Dynamic Textures using a Bag of Dynamical Systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 342-353, 2013. Article (CrossRef Link)
[12] A. B. Chan and N. Vasconcelos, "Mixtures of dynamic textures," *in Proc. of IEEE International Conference on Computer Vision*, vol. 1, pp. 641–7, 2005. Article (CrossRef Link)
[13] A. B. Chan and N. Vasconcelos, "Classifying video with kernel dynamic textures," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, June, 2007. Article (CrossRef Link)
[14] A. B. Chan and N.Vasconcelos, "Probabilistic kernels for the classification of auto-regressive visual processes," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, June, 2005. Article (CrossRef Link)
[15] Kantz H and Schreiber T, "Nonlinear Time Series Analysis," (Cambridge: Cambridge University Press), 1997.
[16] M. Perc, "The Dynamics of Human Gait," *European Journal of Physics*, vol. 26, pp. 525-534, 2005. Article (CrossRef Link)
[17] Chaudhuri BB, Sakar N, "Texture segmentation using fractal dimension," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp:72-77, 1995. Article (CrossRef Link)
[18] A.P.Pentland, "Fractal Based Description of Natural Scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.6, no. 6, pp. 661-674, 1984. Article (CrossRef Link)
[19] D. J. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *Journal Optical Society America*, 1987, vol. A4, pp. 2379-2394. Article (CrossRef Link)
[20] F. Taken, "Detecting Strange Attractors in Turbulence," *Lecture Notes in Mathematics*, ed D. A.Rand & L. S. Young, 1981.
[21] A. M. Fraser and H. L. Swinney, "Independent Coordinates for Strange Attractors from Mutual Information," *Physical Review A*, vol. 33, no. 2, pp. 1134-1140, February, 1986. Article (CrossRef Link)
[22] M. B. Kennel, R. Brown and H. D. I. Abarbanel, "Determining Embedding Dimension for Phase Space Reconstruction using A Geometrical Construction," *Physical Review A*, vol. 45, no. 6, pp. 3403-3411, June, 1992. Article (CrossRef Link)
[23] B. Ghanem and N. Ahuja, "Maximum margin distance learning for dynamic texture recognition," *in Proc. of European Conference on Computer Vision*, pp. 223-236, 2010. Article (CrossRef Link)
[24] Saisan, P., Doretto, G., Wu, Y. N., and Soatto, S., "Dynamic texture recognition," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 58-63, 2001. Article (CrossRef Link)
[25] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," *Visual Surveillance and Performance Evaluation of Tracking and*

Surveillance, pp. 65-72, 2005. Article (CrossRef Link)
[26] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000. Article (CrossRef Link)
[27] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000. Article (CrossRef Link)
[28] Yasmin Mussarat, Sharif Muhammad, Mohsin Sajjad and Irum Isma, "Content Based Image Retrieval Using Combined Features of Shape, Color and Relevance Feedback," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 12, pp. 3149-3165. December, 2013. Article (CrossRef Link)
[29] Huy Hoang Nguyen, GueeSang Lee, SooHyung Kim and Hyung Jeong Yang, "An Effective Orientation-based Method and Parameter Space Discretization for Defined Object Segmentation," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 12, pp. 3180-3199, December, 2013. Article (CrossRef Link)
[30] Derpanis K G, Wildes R P, "Spacetime texture representation and recognition based on a spatiotemporal orientation analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 6, pp. 1193-1205, 2012. Article (CrossRef Link)
[31] C. Schuldt and I. Laptev, "Recognizing human actions: A local SVM approach," in *Proc. of the International Conference on Pattern Recognition*, vol. 3, pp. 32-36, 2004. Article (CrossRef Link)
[32] Scovanner P, Ali S, Shah M, "A 3-dimensional sift descriptor and its application to action recognition," in *Proc. of the 15th international conference on Multimedia*, ACM, pp. 357-360, 2007. Article (CrossRef Link)
[33] Li, T., Mei, T., Kweon, I. S., and Hua, X. S, "Contextual bag-of-words for visual categorization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 4, pp. 381-392, 2011. Article (CrossRef Link)
[34] Li, T., Yan, S., Mei, T., Hua, X. S., and Kweon, I. S., "Image decomposition with multilabel context: Algorithms and applications," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2301-2314, 2011. Article (CrossRef Link)

Yong Wang is a Ph.D. candidate in control science and engineering in the School of Aeronautics and Astronautics at Shanghai Jiao Tong University. His research interests include visual tracking, pattern recognition, and machine learning.

Shiqiang Hu is a Professor and the vice-president of the School of Aeronautics and Astronautics at Shanghai Jiao Tong University. He received his M.S. (1998) and Ph.D. (2002) degrees at Beijing Institute of Technology both in electronics and information technology. His research areas include intelligent information processing, image understanding, and nonlinear filtering.