Chaotic Features for Dynamic Textures Recognition with Group Sparsity Representation

Xinbin Luo[1], Shan Fu[1] and Yong Wang[1]
[1] School of Aeronautics and Astronautics, Shanghai Jiao Tong University
Shanghai, 200240 - China
[e-mail: xbluo_sjtu@126.com]
*Corresponding author: Xinbin Luo

Received April 18, 2015; revised July 12, 2015; ; revised August 24, 2015; accepted August 25, 2015; published November 30, 2015

Abstract

Dynamic texture (DT) recognition is a challenging problem in numerous applications. In this study, we propose a new algorithm for DT recognition based on group sparsity structure in conjunction with chaotic feature vector. Bag-of-words model is used to represent each video as a histogram of the chaotic feature vector, which is proposed to capture self-similarity property of the pixel intensity series. The recognition problem is then cast to a group sparsity model, which can be efficiently optimized through alternating direction method of multiplier algorithm. Experimental results show that the proposed method exhibited the best performance among several well-known DT modeling techniques.

Keywords: Dynamic textures recognition, chaotic feature vector, group sparsity

http://dx.doi.org/10.3837/tiis.2015.11.017 **ISSN : 1976-7277**

1. Introduction

Dynamic textures (DTs) are videos that exhibit certain temporal stationarity. Examples of DTs in nature are fire, river, and boiling water. DT applications range from remote monitoring for prevention of natural disasters, such as forest fires, to various types of surveillance, such as public security and traffic flow. However, DT recognition in a dynamic environment is challenging because of various changes in appearance, such as illumination, scale, and viewpoint changes.

DTs are generated by a complex time-varying dynamical system. A complete description of this system requires enumeration of all independent variables and differential equations controlling the evolution. A set of variables defining the state space is selected to obtain the description of a dynamical system, and a function maps the previous state to the next one. The type of mapping function determines whether the system is linear or non-linear. For instance, DTs can be represented in terms of state variables defined as the pixel intensity or pixel intensity over time, followed by assuming a linear or non-linear dynamical model. However, obtaining a complete analytic description of a dynamical system is extremely difficult in practical scenarios. Classical algorithms of DT recognition based on linear dynamical systems (LDSs) often assume that the model is first-order Markov and linear, thereby restricting nonlinear DT modeling.

Chaos theory was developed to study nonlinear systems [1] and achieved great success in science and technology. Chaotic features that capture motion information have been used in action recognition [2], dynamic scene understanding [3], and anomaly detection [4].

Fig. 1 (a) illustrates DTs of fire from a dataset [5]. People cannot determine whether the fire is a forest fire or a candle fire; this condition is called self-similarity, wherein the object has the same structure at all scales. **Fig. 1 (b)** shows two pixel intensity series in positions (5, 5) and (15, 5). The horizontal row and vertical column denote time and pixel intensity, respectively. Many physical processes produce fractal property, and a natural scene can be modeled by fractal dimension [6]. Several natural textures have a linear log power spectrum, which is related to the fractal dimension and is suitable to characterize textures [7]. Self-similarity is conjectured to exist in each pixel intensity series as the DTs exhibit certain stationary properties in time domain [5]. Therefore, we computed fractal dimension from each pixel intensity series.

A recent study on DT recognition showed no specific preferences on classifier selection. Nearest neighbor (NN) [2, 14] and support vector machine (SVM) [3, 13] are commonly used in object recognition. Sparse representation has received wide attention in visual recognition because of its robustness against occlusions and noise. Facial recognition is formulated as finding a sparse linear combination of dictionary templates [8]. However, this method separately learns sparse representation of training data and ignores the relationships among the training data, which ultimately constrain their representation. Multi-task sparse learning recently aims to extend the l_1 framework to jointly learn the sparse model. Multi-task learning has been applied in image classification [16] and annotation [17]. The present work used two sparse models [9] to capture the relations across features. In particular, the recognition model is formulated to two sparsity norms: a $l_{1,\infty}$ norm and l_{11} norm, which penalizes the sum of maximum absolute values of each row to capture the shared and non-shared features, respectively. To efficiently solve the model, we utilized the alternating direction method of multiplier algorithm for optimization with guaranteed fast convergence rate. The classification is ruled in favor of the class that has the lowest total reconstruction error.

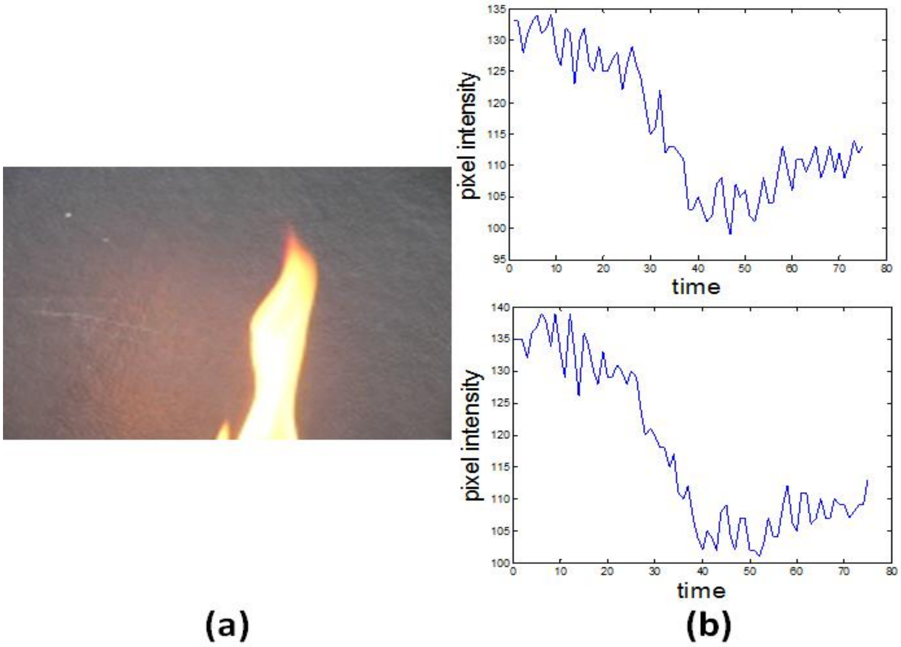


Fig. 1. (a) One frame from fire video. (b) Pixel intensity series in positions (5, 5) and (15, 5).

The contributions of this work are as follows:

(1) Pixel intensity series is treated as a basic feature to describe DTs. Prior studies [10, 26] mainly focused on the spatial information of DTs and ignored the temporal information. Optical flow method is used to characterize the temporal information of DTs. However, this method only computes motion information between two frames. Pixel intensity series contains the complete temporal information that can be used to represent the DTs. Chaotic feature is introduced to characterize pixel intensity series. Prior work used chaotic features such as largest Lyapunov exponent, correlation dimension, and correlation integral to represent the motion information of the time series. In the present work, box count dimension is used to characterize the self-similarity of each pixel intensity series.

(2) A group sparsity for DT recognition is formulated. We used superposition structure to capture the joint sparsity model. The two structures captured the common and special features.

(3) The formulation proposed in this paper is solved using ADMM algorithm to compute the problem robustly and quickly.

The rest of the paper is organized as follows: Section 2 provides an in-depth review of related work, and Section 3 presents the proposed algorithm. The experimental results are shown in Section 4, and Section 5 summarizes the paper.

2. Related Work

Extensive literature on DT recognition is widely available [27–31]. We only briefly reviewed nominal DT recognition methods, chaos theory, and sparse representation, which are most related to the present work.

LDS is proposed in [5], which is learned by system identification as a model for DT recognition. The UCLA dataset provided contains 200 videos and is widely used as a benchmark dataset in various DT recognition methods. Gaussian mixture models of LDSs is used in [11] to model DTs. Kernel principal component analysis (PCA) is combined with LDSs to model a wider range of video motions [12]. DT is modeled as an LDS and compared with a probabilistic kernel [13]. Bag-of-words (BoWs) representation is used in [14] to model each DT video with LDSs and recognize DTs. However, the LDSs reside in a non-Euclidean space, making traditional methods of forming a codebook based on clustering Euclidean feature descriptors no longer applicable.

Linear assumption undermines the performance of DT recognition because nonlinearity exists in DTs. Chaos theory is an ideal tool for analyzing nonlinear systems. Different chaotic features have been recently introduced in the computer vision community to represent chaotic time series. Trajectories of reference points are used as chaotic time series as well [2]. Chaotic features such as Lyapunov exponent, correlation integral, and correlation dimension are computed and combined with a feature vector. Experimental results validated the feasibility and merits of using chaotic features. People’s tracks are treated as chaotic time series in [4]. Chaotic features such as largest Lyapunov exponent and correlation dimension are calculated and concatenated to a feature vector to detect and locate anomalies. The 960-dimensional GIST feature of the dynamic scene video is computed in [3]. Each dimensional feature is treated as a time series. Chaotic features, including correlation integrals, Lyapunov exponent, and correlation dimension are combined to become a feature vector. Experiments showed that the feature vector can differentiate different dynamic scenes.

Other chaotic features are used in image processing. A modified box count approach is proposed to estimate fractal dimension, and image segmentation experiment is effective [15]. Most of these works chose NN and SVM as classifiers.

Sparse representation has achieved promising results compared with NN and SVM in [8]. This method assumes that the training data of a particular class approximately form a linear basis set for any testing data belonging to this class. However, sparse representation only selects data from a group of correlated training data and does not represent the testing data in terms of all training data from the correct group. Joint sparse model is proposed to overcome this problem. Image classification problem is casted into a multi-task joint covariate selection model that is optimized through accelerated proximal gradient method [16]. An effective projected gradient method is developed for optimization of $l_{1,\infty}$ regularization problem and achieved good result on image annotation [17].

In this work, the complex structure of features in different classes does not solely fit any model. The data structure model can be expressed as the superposition of a number of simpler models. The difficulty is how to characterize different structures without any ambiguity. Thus, we need not only to reduce the size of the problem by imposing the structure but also to further restrict each structure to be consistently incoherent from each other to obtain robustness.

Inspired by the aforementioned study, the present work aims to improve the DT modeling and capitalize on the interdependence among features. To achieve this goal, we propose a chaotic feature vector modeling and group sparsity representation method for DT recognition. Different from prior works, we used box count dimension to depict the fractal information of the pixel intensity series. Furthermore, to capture the relationship among features, we imposed group sparsity condition that uniquely partitions the space model so that each non-zero element of the space belongs to only one structure, and the ADMM algorithm efficiently solved the optimization problem.

3. Proposed Algorithm

The proposed approach has three interconnected components. First, given a video, chaotic and other features are calculated from each pixel intensity series and combined to a chaotic feature vector. A video can be represented by a feature vector matrix. Then, videos are represented by the well-known BoWs representation, which has been adopted by many computer vision researchers [14, 21, 22, 32]. Finally, group sparsity model is learned, and a testing video is represented by the BoWs model and classified by the model.

Fig. 2 illustrates the flowchart of the DT recognition system.

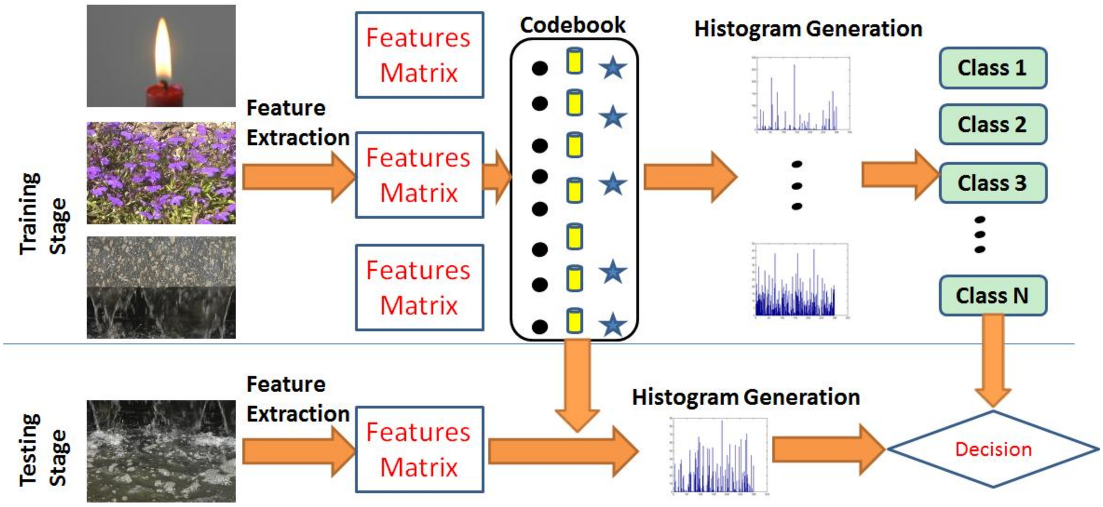


Fig. 2. Flowchart of the DT recognition system. In the training stage, chaotic feature vector is extracted from each pixel intensity series, and histograms are generated in the BoWs representation. All the training videos are represented by the histograms and fed into group sparsity model. In the testing phase, following a similar process, an unknown video is represented by a histogram of codewords learned from the training dataset, and a class label is outputted using the learned model.

Details on chaotic feature vector, BoWs representation, and group sparsity model are discussed in the following sections.

3.1 Chaotic feature vector

This section presents the background material related to the chaos theory. Supposing a collection of $V = v_1, \cdots, v_n$, video sequences, a one-dimensional pixel intensity series $\{x_{i,j}(t)\}_{t=1}^{T} = v_{n_l}(i,j,:)$is shown in **Fig. 1**, where i and j are horizontal row and vertical column coordinate of $x_{i,j}(t)$ in video v_{n_l}, respectively, T is the total number of the sequence, and $x_{i,j}(t)$ is a pixel intensity series. Takens' theorem [18] states that a map exists between the original state space and a reconstructed state space. Thus, the pixel intensity series can be written into a matrix as

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

where τ_{ij} is embedding delay, and m_{ij} is embedding dimension in position (i,j). τ_{ij} and m_{ij} can be computed by mutual information algorithm [19] and false nearest neighbor algorithm [20], respectively.

Chaotic features are measures that quantify the properties invariant under transformations of the state space. Next, the chaotic feature used in this work is introduced.

3.1.1 Box count dimension

The box count dimension [1] that measures the degree of a set held in space is one of the fractal dimensions. If a point set is covered by a regular grid of boxes with length δ and $N(\delta)$ is the number of boxes containing at least one point, then box count dimension $\mathrm{D_b}$ is

$$D_b = \lim_{\delta \to 0} \frac{\ln N(\delta)}{\ln \frac{1}{\delta}}. \tag{2}$$

3.1.2 Chaotic Feature Vector

Embedding delay and dimension are two important parameters to determine the structure of the phase space. The mean value of the pixel intensity series ($\mathrm{M_{ij}}$) is an important indicator of pixel intensity series. The chaotic feature vector in this work is $\mathrm{f_{ij}} = \left\{\mathrm{\tau_{ij}, m_{ij}, D_{b_{ij}}, M_{ij}}\right\}$. Each pixel intensity series $\mathrm{x_{i,j}(t)}$ is represented by a chaotic feature vector $\mathrm{f_{ij}}$, and a video $\mathrm{v_{n_\mathit{l}}}$ can be transformed to a feature vector matrix.

3.2 BoWs representation

The BoWs representation contains a codebook consisting of a set of representative chaotic feature vectors learned from the training samples. The codebook is learned by clustering through vector quantization. Each chaotic feature vector is assigned to the closest codeword in terms of Euclidean distance during clustering. These representative chaotic feature vectors are referred to as codewords in the context of BoWs representation. DT is represented as a histogram of the number of occurrences of each codeword count according to

$$h(v_{\mathrm{n}_l}) = \left(h_k(v_{\mathrm{n}_l})\right)_{k=1\ldots K},\ with\ h_k(v_{\mathrm{n}_l}) = NC(v_{\mathrm{n}_l}, f_k), \tag{3}$$

where $\mathrm{NC}(\mathrm{v_{n_\mathit{l}}}, \mathrm{f_k})$ denotes the number of occurrences of chaotic feature vector $\mathrm{f_k}$ in video $\mathrm{v_{n_\mathit{l}}}$, and K is the number of codewords.

3.3 Group sparsity model

Feature denotes the histogram obtained by BoWs representation in this section. We suppose that a training set $\mathrm{C} = [\mathrm{C_1}, \cdots, \mathrm{C_P}]$ denotes the training feature in which $\mathrm{C_p} \in \mathrm{R^{m_f * 1}}$, whereas P is the number of training samples. In this model, $\mathrm{m_f}$ is the dimension of the training samples. Q is the number of testing samples given the testing sample $\mathrm{Y} = \left[\mathrm{Y_1}, \cdots, \mathrm{Y_Q}\right]$, $\mathrm{Y_q} \in \mathrm{R^{m_f * 1}}$. Thus, we can consider the linear representation problem as follows:

$$\mathrm{Y} = \sum_{\mathrm{p}=1}^{\mathrm{P}} \mathrm{C_p W_p} + \varepsilon, \tag{4}$$

where $\mathrm{W_p} \in \mathrm{R^{P*1}}$ is a reconstruction coefficient vector associated with the pth class, and ε is the residual term. $\mathrm{W_p}$ denotes the representation coefficients from the pth class. Let $\mathrm{W} = [\mathrm{W_1}, \cdots, \mathrm{W_P}]$, the group sparsity representation is formulated as the solution to the following problem:

$$\min_{L,S}\frac{1}{2}\|Y-C(L+S)\|_F^2+\ \lambda_1\|L\|_{1,\infty}+\lambda_2\|S\|_{1,1},\ W=\ L+S\ , \tag{5}$$

where L is the row group sparsity component, and S is the elementwise sparse component. λ_1 and λ_2 are tradeoff parameters between reliable construction and joint sparsity regularization that control the group sparsity regularization on L and S, respectively.

Group sparsity representation formulates the unknown parameter as a superposition of a row group sparsity matrix L and a sparse matrix S that correspond to the features shared across many and few samples, respectively. Different norms are enforced on L and S, encouraging row group sparsity in L and elementwise sparsity in S. The corresponding models use row group sparsity and elementwise sparsity regularizations.

The ADMM algorithm [23] is a convex optimization algorithm that has recently attracted attention because of its applicability to various machine learning and computer vision problems. In particular, the ADMM algorithm can take advantage of the structure of the problems, which involve optimizing sums of fairly simple convex functions as follows:

$$\min_{d,z} g_1(d)+g_2(z)\,,\text{s.t.}\,Ad+Bz=e, \tag{6}$$

where $d\in R^b$, $z\in R^a$, $e\in R^p$, $A\in R^{p*b}$, $B\in R^{p*a}$. $g_1{:}\,R^b\to R$, $g_2{:}\,R^a\to R$, g_1 and g_2 are convexes.

The scaled form of the ADMM algorithm consists of the following iterations:

$$d^{k+1}\coloneqq \operatorname{argmin}_d\left(g_1(d)+(\rho/2)\left\|Ad+Bz^k-e+u^k\right\|_2^2\right) \tag{7}$$

$$z^{k+1}\coloneqq \operatorname{argmin}_z\left(g_2(z)+(\rho/2)\left\|Ad^{k+1}+Bz-e+u^k\right\|_2^2\right) \tag{8}$$

$$u^{k+1}\coloneqq u^k+Ad^{k+1}+Bz^{k+1}-e \tag{9}$$

where $\rho>0$ is the scaled dual variable.

The construction of the algorithms consists of two main steps: (1) reformulating the optimization problem into one that has partially separable objective functions by adding new variables and constraints, and (2) applying an alternating direction method to the resulting problem.

The proposed group sparsity representation problem can be decomposed into two sub-problems [Equations (10) and (11)], and then each of the two sub-problems is iterated until convergence as follows:

$$\text{L:}\ \text{minimize}_L\frac{1}{2}\|(Y-CS)-CL\|_F^2+\ \lambda_1\|L\|_{1,\infty}, \tag{10}$$

$$\text{S:}\ \text{minimize}_S\frac{1}{2}\|(Y-CL)-CS\|_F^2+\lambda_2\|S\|_{1,1}. \tag{11}$$

The proposed ADMM algorithm comprises two alternately updating matrix sequences L and S. The approach is summarized as Algorithm 1 (**Fig. 3**).

```
Algorithm 1: group sparsity algorithm implemented by ADMM.
Input: $Y, C$,
Initialize L, S,
k=1,
While stopping criterion is not met do
    Solve the convex optimization problem:
        $L^{k+1} = \text{minimize}_L \frac{1}{2} \|(Y^k - C^k S^k) - C^k L^k\|_F^2 + \lambda_1 \|Z_L^k\|_{1,\infty}$
        $s.t.\ L^k = Z_L^k,$
        The scaled form of ADMM consists of the following iteration.
        $L^{k+1} = ((C^k)^T C^k + \rho I)^{-1}[(C^k)^T (Y^k - C^k S^k) + \rho(Z_L^k - u_L^k)],$
        $Z_L^{k+1} = prox_{1inf}(Z_L^k),$
        $u_L^{k+1} = u_L^k + L^{k+1} - Z_L^{k+1},$
        where $prox_{1inf}$ denotes proximal method for $\|*\|_{1,\infty}$ [24].

    Solve the convex optimization problem with updated $L^{k+1}$:
        $S^{k+1} = \text{minimize}_S \frac{1}{2} \|(Y^k - C^k L^{k+1}) - C^k S^k\|_F^2 + \lambda_2 \|Z_S\|_{1,1}$
        $s.t.\ S^k = Z_S^k,$
        The scaled form of ADMM consists of the following iteration.
        $S^{k+1} = ((C^k)^T C^k + \rho I)^{-1}[(C^k)^T (Y^k - C^k L^{k+1}) + \rho(Z_S^k - u_S^k)],$
        $Z_S^{k+1} = prox_{11}(Z_S^k),$
        $u_S^{k+1} = u_S^k + S^{k+1} - Z_S^{k+1},$
        where $prox_{11}$ denotes proximal method for $\|*\|_{1,1}$.

    k = k+1;
end while
$W = L + S$,
Output: solution W, L, S to equation (5).
```

Fig. 3. Group sparsity algorithm implemented by ADMM algorithm.

4. Experiments

4.1 Dataset introduction

Most LDS-based DT recognition methods choose the UCLA dataset as the test bed containing 50 categories of different DTs, each with four gray-scale videos captured from different viewpoints. Each sequence consists of 75 frames with a size of 110×160. The UCLA-50 dataset is classified into 50 classes as implemented in [25].

The second dataset is called new DT-10 dataset. We collected 16 river videos with smooth shaking and combined them with the UCLA dataset. In each video, the dimension is reduced to 48×48 with 75 frames. The dataset is classified into 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 the proposed algorithm when DTs are taken under different viewpoints, scales, and other unconstrained environments.

The third dataset is DynTex++ dataset [26], which contains 36 categories of different DTs with 100 DTs in each category. This dataset contains a total of 3600 videos, thereby providing a richer benchmark.

Fig. 4 shows examples from the new DT-10 and DynTex++ datasets.

Codebook formation:

The proposed four-attribute chaotic feature vector consists of embedding delay, embedding dimension, box count dimension, and mean value of pixel intensity series. The chaotic feature vector is normalized to obtain values between 0 and 1. To generate the codebook, K-means clustering algorithm is directly used on the Euclidian distance of the four-attribute vector across the entire training feature vector matrix. The obtained cluster centers form the histogram bins. The number of cluster K is the codebook size, which varies from 100 to 1000. After formation of the codebook, each four-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 the feature vector matrix are mapped to the cluster centers, the feature vector matrix can be represented by a histogram of the codebook.

The results reported in this work have been averaged for over 10 times. The regularization constants λ_1 and λ_2 are set to 0.01.

We chose the smallest reconstruction error as the classification method, similar to the approach in [8].

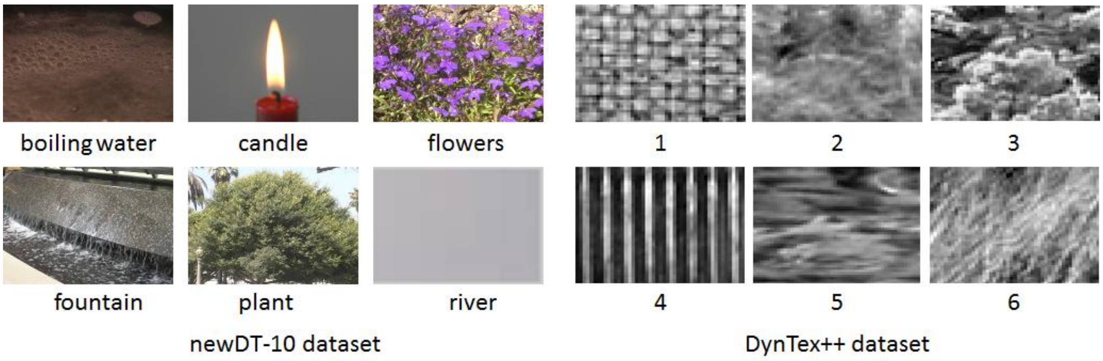


Fig. 4. Examples from new DT-10 and DynTex++ datasets

4.2 UCLA-50 dataset

Single LDS Approach [29]: In the first baseline method, we modeled the entire DT video using a single LDS. Given a testing DT video, we computed the Martin distance between the testing LDS and each of the LDS models of the training videos, and then we used an NN classifier based on this distance. We tested all system orders in the range [2, 4, 6] and considered the best results out of these as the single LDS baseline. This approach is identical to the one originally proposed in [29].

Spatial temporal feature: The second baseline method is the BoWs [32] approach. We extracted spatial temporal features from the DT videos and reduced 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.

GIST [33] feature is adopted in the present work. We first extracted the 960-dimensional GIST feature per videoframe and used the BoWs approach.

Dense SIFT [21] that represents the nature of images is also used in the present work along with the BoWs approach.

Furthermore, a one-versus-all scheme is used. We listed the state-of-the-art results from the UCLA-50 dataset using BoWs representation related to **Table 1**. The confusion matrix is shown in **Fig. 5**. The best recognition rate achieved on the UCLA-50 dataset is 81% [25]. The proposed approach achieved the best recognition accuracy of 92% among all cases. The size of the codebook is 700.

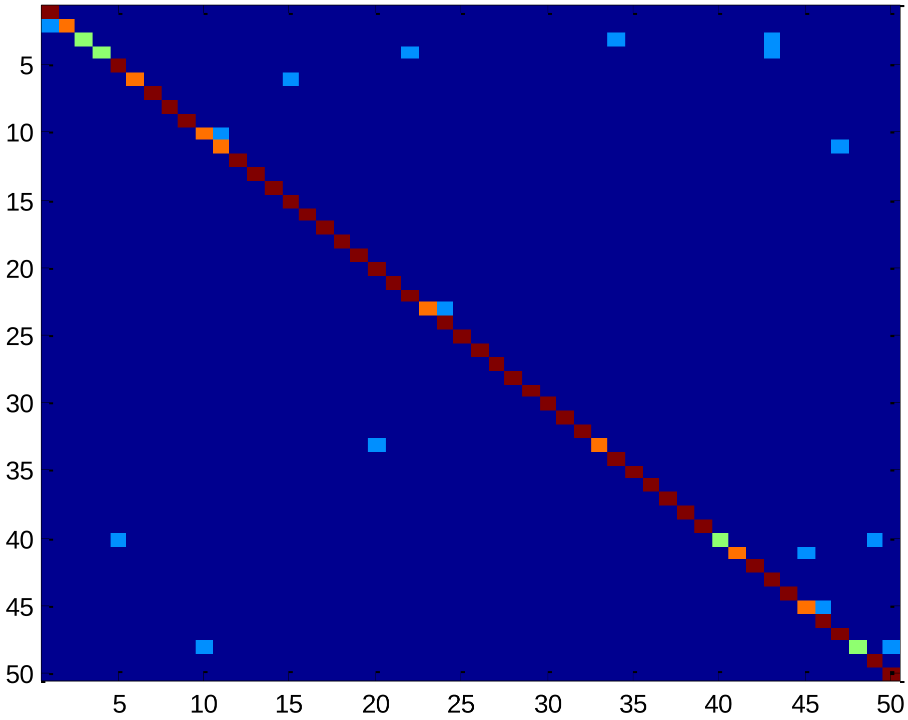


Fig. 5. Confusion matrix on the UCLA-50 dataset in the proposed method. Each row and column in the confusion matrix corresponds to the ground truth class and assigned label, respectively.

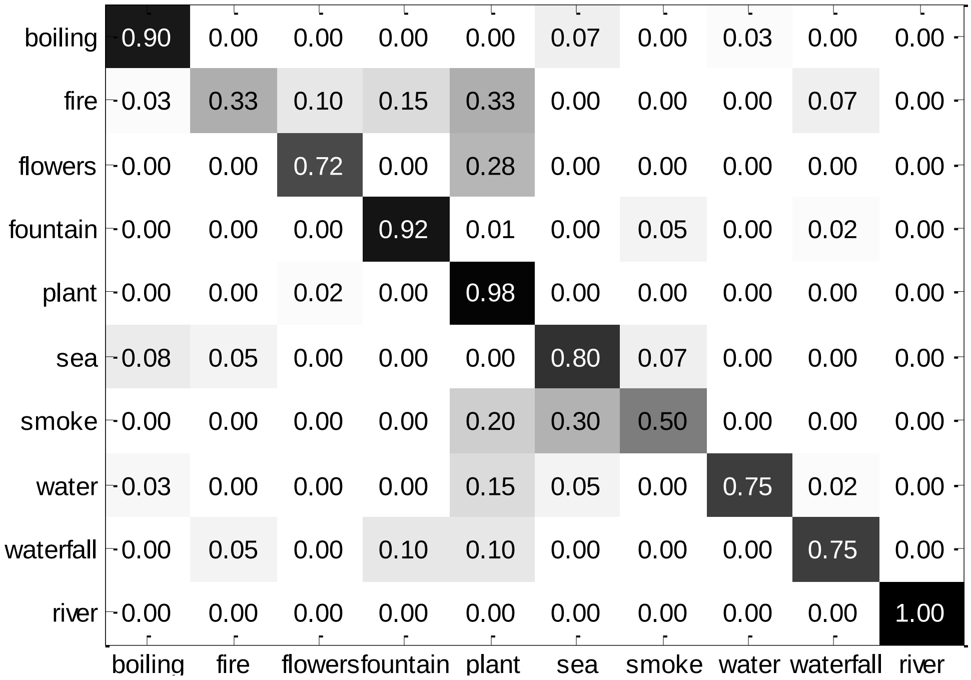
Table 1. Recognition results on the UCLA-50 dataset.

Method	1-NN	Sparse coding	SVM	Group sparsity representation
[25]	81%	—	—	—
Dense SIFT	61%	62%	62%	64%
GIST	41%	40%	38%	41%
Spatial temporal feature	70%	70%	69%	72%
Pixel intensity series	54.5%	60%	61%	62%
Chaotic feature vector	79%	91%	82%	92%

4.3 New DT-10 dataset

We chose 50% of the dataset for training and the rest for testing and compared the performance of the proposed approach with two baselines: single LDS approach and BoWs approach.

The state-of-the-art results on the new DT-10 dataset using BoWs representation are listed in **Table 2**. **Fig. 6** shows the confusion matrix for the proposed approach on the new DT-10 dataset corresponding to the recognition rate of 89.48%. The codebook size is 200. The recognition rate using single LDS, dense SIFT, GIST, and spatial temporal feature is 63%, 63%, 55%, and 78.33%, respectively.



	boiling	fire	flowers	fountain	plant	sea	smoke	water	waterfall	river
boiling	0.90	0.00	0.00	0.00	0.00	0.07	0.00	0.03	0.00	0.00
fire	0.03	0.33	0.10	0.15	0.33	0.00	0.00	0.00	0.07	0.00
flowers	0.00	0.00	0.72	0.00	0.28	0.00	0.00	0.00	0.00	0.00
fountain	0.00	0.00	0.00	0.92	0.01	0.00	0.05	0.00	0.02	0.00
plant	0.00	0.00	0.02	0.00	0.98	0.00	0.00	0.00	0.00	0.00
sea	0.08	0.05	0.00	0.00	0.00	0.80	0.07	0.00	0.00	0.00
smoke	0.00	0.00	0.00	0.00	0.20	0.30	0.50	0.00	0.00	0.00
water	0.03	0.00	0.00	0.00	0.15	0.05	0.00	0.75	0.02	0.00
waterfall	0.00	0.05	0.00	0.10	0.10	0.00	0.00	0.00	0.75	0.00
river	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Fig. 6. Confusion matrix on the new DT-10 dataset for the proposed method.

Table 2. Recognition results of the new DT-10 dataset.

Method	1-NN	Sparse Coding	SVM	Group sparse representation
Single LDS	63%	—	—	—
Dense SIFT	60%	61%	58%	63%
GIST	42%	46%	45%	55%
Spatial temporal feature	78.33%	—	—	—
Pixel intensity series	76%	72%	73%	79%
Chaotic feature vector	80%	82%	82%	89.48%

Overall, the results are reasonable although a few classes performed poorly. The confusion matrix shows the confusion between “boiling” and “sea,” “sea” and “smoke,” “water” and “boiling,” and “waterfall” and “fountain.” This result is consistent with our intuition that similar DTs are more easily confused with one another. From the confusion matrix, one can observe that the “fire” and “smoke” are confused with “plant” and “sea,” respectively, but no confusion is observed between “waterfall” and “plant.” The reason may be that the chaotic feature vectors between these classes are similar because of the analogous structure of pixel intensity series in these classes.

4.4 DynTex++ dataset

The proposed approach is applied to the DynTex++ dataset using an experimental setup similar to the one in the new DT-10 dataset experiment. We chose 50% of the dataset for training and the rest for testing and compared the performance of the proposed approach with the single LDS approach to categorize DTs. The best recognition rate of single LDS on the

DynTex++ dataset is 47.2%.

An average recognition rate of 53.4% is obtained for pixel intensity series and 65.11% for chaotic feature vector. The size of the codebook is 900. The best performance in [26] is 63.7% on the DynTex++ dataset. The state-of-the-art results on the DynTex++ dataset are listed in **Table 3**.

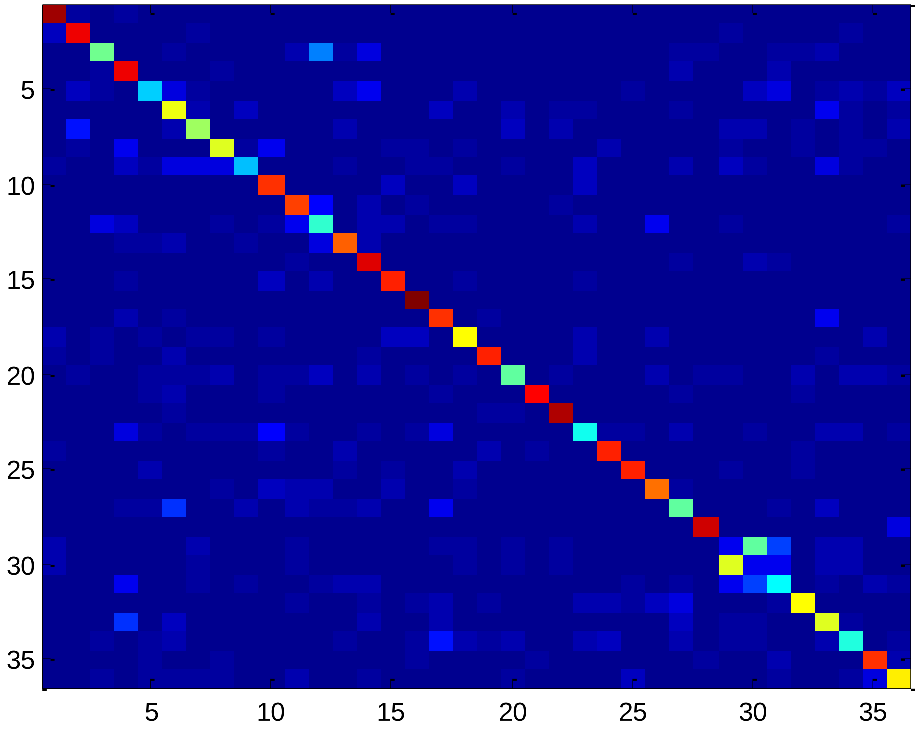


Fig. 7. Confusion matrix on the DynTex++ dataset of the proposed method.

Table 3. Recognition results on the DynTex++ dataset.

Method	1-NN	Sparse Coding	SVM	Group sparse representation
Single LDS	47.2%	—	—	—
Dense SIFT	44%	49%	48%	55%
GIST	23%	22%	24%	28%
Spatial temporal feature	50%	51%	50%	57%
Pixel intensity series	49.67%	35.67%	40%	53.4%
Chaotic feature vector	63.89%	64%	63%	65.11%

4.5 Codebook size

This experiment aims to validate the effect of different codebook sizes on DT recognition performances. As shown in **Fig. 8**, some dependencies of the recognition accuracy are observed on the codebook size, and recognition accuracy is not increased as the codebook size increased. “1,” “2,” and “3” stand for the recognition results of the proposed chaotic feature vector method for the UCLA-50 dataset, new DT-10 dataset, and DynTex++ dataset, respectively. “4,” “5,” and “6” denote the recognition results of the pixel intensity series as features of the UCLA-50 dataset, new DT-10 dataset, and DynTex++ dataset, respectively. The recognition rate of the chaotic feature vector is higher than that of the pixel intensity series most of the time.

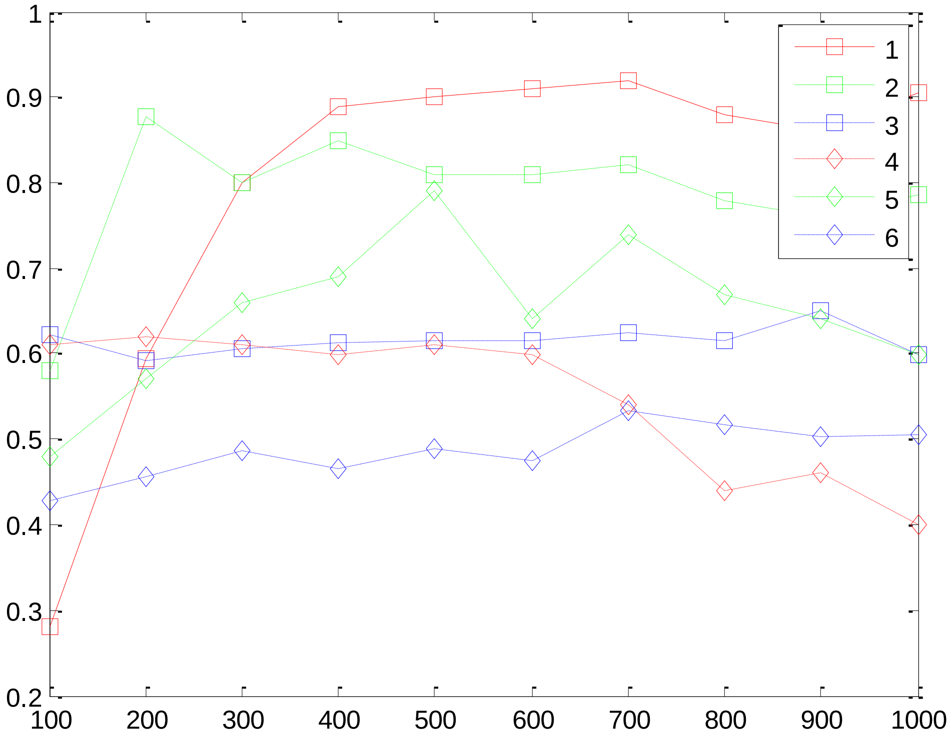


Fig. 8. Recognition performance on UCLA-50 dataset, new DT-10 dataset, and DynTex++ dataset using different codebook sizes. The horizontal row indicates the codebook size, and the vertical column indicates the recognition rate.

4.6 Discussions

The following are observed from the experimental results:

The chaotic feature vector can outperform LDSs in most cases. Therefore, we propose the use of the chaotic feature vector because, along with more accurate modeling of DTs, this vector also offers ways to combine with the group sparsity algorithm. The proposed group sparsity modeling approach significantly outperforms the traditional LDS-based method. **Table 2**, **3** demonstrate that the proposed approach exhibits more than 20% (new DT-10 dataset) and 15% (DynTex++ dataset) improvement over LDS-based counterpart [29]. However, we did not compare these results with [14] because of different experiment dataset and experiment settings.

As shown in Tables 1 to 3, the group sparsity method improved the recognition results for either the pixel intensity series or chaotic feature vector compared with the traditional methods. This result can be attributed to the fact that the group sparsity model captures the relationship among features. The model enforces the two norms regulating the common and special features across the samples, respectively.

Errors appeared when the structures of pixel intensity series of the two classes are similar. Given that the chaotic feature vector mainly captures the texture information of the pixel intensity series, this vector ignores the other DT information, such as the motion information of the two frames.

Traditional DT recognition methods, such as the LDS-based one, have been studied and perfected for at least a decade, whereas the proposed method is built on chaotic feature vector and group sparsity learning, which have not been previously applied to DT analysis. The proposed method may have a much greater potential for improvement in the future. In addition, this model can be used in video segmentations, video localization, and other applications because the chaotic feature vector models each pixel intensity series.

The present work and the one in reference [34] exhibit two differences. First, a different chaotic feature vector is used with [a]. Box count dimension and correlation dimension are used in [a] as well, which accounts for the redundancy in the feature vector. In the present work, we improved the chaotic feature vector to include the mean value of the pixel intensity series,which can capture the motion information of the series. The second difference is that multi-task learning is used in the present work, and the ADMM algorithm is used to solve the formulation. The group sparsity formulation decomposes the representation matrix to the inliers and outliers, which is more convenient to characterize the common parts of the same class and the different parts of different classes. The ADMM algorithm guarantees the convergence of the equation. The experimental results showed that the present work achieves better results than [34].

The ADMM algorithm convergence curve on three test videos is shown in **Fig. 9**. The value of the cost function in Equation (5) is gradually minimized with the increase in iteration number. **Fig. 9** reflects the minimization speed of the cost function with respect to iterations.

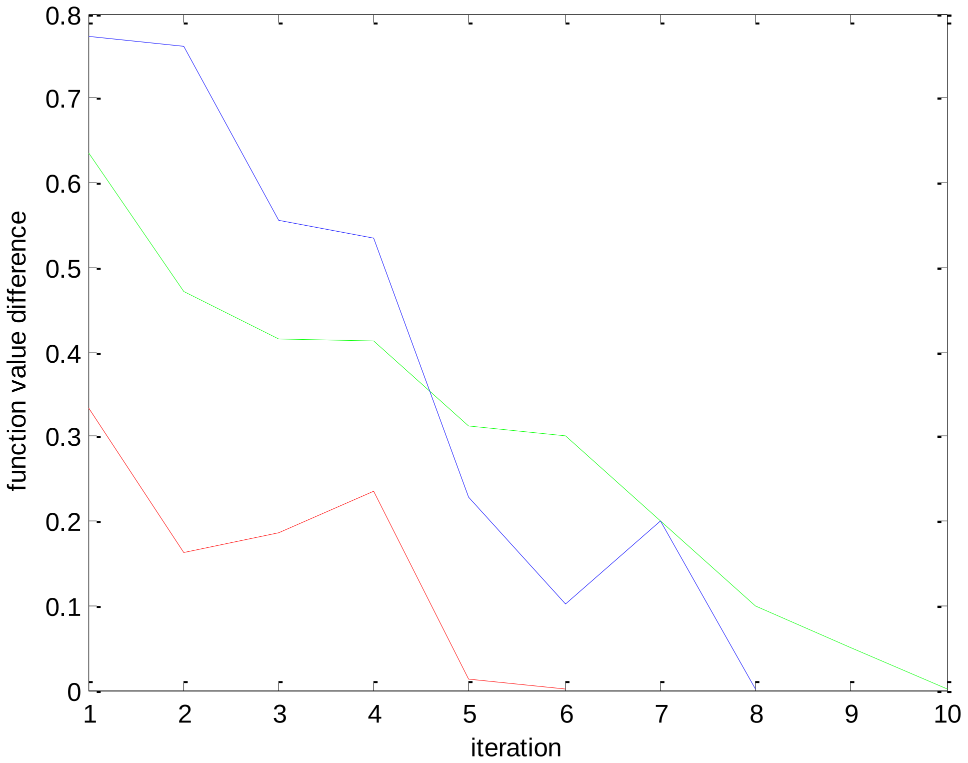


Fig. 9. Convergence curve of the cost function in Equation (5) on an example frame of three test videos for the ADMM algorithm.

5. Conclusion

A powerful DT recognition framework is developed in this work based on the chaotic feature vector and group sparsity model. The proposed ideas are fairly general and applicable to other recognition problems, such as action recognition. The experimental results demonstrate that the proposed approach is highly accurate and is robust against scale variations and, to a certain extent, to viewpoint changes. This robustness is achieved by exploiting the discriminative nature of the chaotic feature vector modeling combined with joint group sparsity regularization. The chaotic feature vector extracted from each pixel intensity series induces effective characterization of the fractal property in the DTs, and the group sparsity model captures the relationship among features. Furthermore, the ADMM algorithm-based numerical solvers ensure the fast and accurate convergence of group sparsity model. Future work must include the deep learning method to directly learn the pixel intensity series and multiple features fusion to recognize the DTs.

Acknowledgements

The authors would like to thank Dr. G. Doretto for sharing the datasets that are used in this paper. This paper is jointly supported by the National Natural Science Foundation of China No. 61374161, China Aviation Science Foundation 20142057006. The authors would like to thank Dr. G. Doretto for sharing the datasets that were used in this paper.

References

[1] Kantz H and Schreiber T 1997 Nonlinear Time Series Analysis (Cambridge: Cambridge University Press)

[2] S. Ali, A. Basharat, and M. Shah, "Chaotic invariants for human action recognition," *IEEE International Conference on Computer Vision*, 2007. Article (CrossRef Link)

[3] 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 (CVPR)*, June 2010. Article (CrossRef Link)

[4] S. Wu, B. Moore, and M. Shah, "Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. Article (CrossRef Link)

[5] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic texture," *International Journal of Computer Vision*, 51(2), pp. 91-109, 2003. Article (CrossRef Link)

[6] A.P.Pentland, "Fractal Based Description of Natural Scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 6(6), pp. 661-674, 1984. Article (CrossRef Link)

[7] D. J. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *Journal Optical Society America*, vol. A4, pp. 2379-2394, 1987. Article (CrossRef Link)

[8] Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y., "Robust face recognition via sparse representation," *PAMI 31* pp. 210–227, 2009. Article (CrossRef Link)

[9] Jalali A, Ravikumar P D, Sanghavi S, et al., "A Dirty Model for Multi-task Learning[C]," *NIPS*, 3: 7, 2010. Article (CrossRef Link)

[10] A. Ravichandran, R. Chaudhry, and R. Vidal, "View-Invariant Dynamic Texture Recognition using a Bag of Dynamical Systems," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2009. Article (CrossRef Link)

[11] 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)

[12] A. B. Chan and N. Vasconcelos, “Classifying video with kernel dynamic textures,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007, Minneapolis. Article (CrossRef Link)

[13] 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 (CVPR)*, 2005. Article (CrossRef Link)

[14] A. Ravichandran, R. Chaudhry, and R. Vidal, “Categorizing Dynamic Textures using a Bag of Dynamical Systems,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2012. Article (CrossRef Link)

[15] Chaudhuri BB, “Sakar N. Texture segmentation using fractal dimension,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17, pp. 72-77, 1995. Article (CrossRef Link)

[16] Yuan, X., & Yan, S., “Visual classification with multi-task joint sparse representation” in *Proc. of IEEE conference on computer vision and pattern recognition*, pp. 3493–3500, 2010. Article (CrossRef Link)

[17] Quattoni, A., Carreras, X., Collins, M.,& Darrell, T, “An efficient projection for l 1, infinity regularization,” in *Proc. of International conference on machine learning*, pp. 857–864, 2009. Article (CrossRef Link)

[18] F. Taken, “Detecting Strange Attractors in Turbulence,” *Lecture Notes in Mathematics*, ed D. A.Rand& L. S. Young, 1981. Article (CrossRef Link)

[19] A. M. Fraser et. al., “Independent Coordinates for Strange Attractors from Mutual Information,” *Phys. Rev.*, 1986. Article (CrossRef Link)

[20] M. B. Kennel et. al, “Determining Embedding Dimension for Phase Space Reconstruction using A Geometrical Construction,” *Phys. Rev.A*, 45, 1992. Article (CrossRef Link)

[21] 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 (CVPR)*, 2006. Article (CrossRef Link)

[22] 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 (CVPR)*, pp. 524 – 531, vol. 2, June 2005. Article (CrossRef Link)

[23] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein. J., “Distributed optimization and statistical learning via the alternating direction method of multipliers” *Found. Trends Mach. Learn.*, 3(1):1–122, 2010. Article (CrossRef Link)

[24] Chen, X., Pan, W., Kwok, J., & Carbonell, J., “Accelerated gradient method for multi-task sparse learning problem,” in *Proc. of IEEE international conference on data mining*, pp. 746–751, 2009. Article (CrossRef Link)

[25] K. G. Derpanis and R. P.Wildes. “Dynamic texture recognition based on distributions of spacetime oriented.” *CVPR*, 2010. Article (CrossRef Link)

[26] B. Ghanem and N. Ahuja. “Maximum margin distance learning for dynamic texture recognition,” *ECCV*, pp. 223-236, 2010. Article (CrossRef Link)

[27] S. Fazekas T. Amiaz, D. Chetverikov, and N. Kiryati, “Dynamic texture detection based on motion analysis,” *Int. J. Comput. Vis.*, vol. 82, no. 1, pp. 48–63, 2009. Article (CrossRef Link)

[28] A. Fournier and W. Reeves, “A simple model of ocean waves,” in *Proc. of ACM SIGGRAPH*, pp. 75–84, 1986. Article (CrossRef Link)

[29] Saisan, P., Doretto, G., Wu, Y. N., and Soatto, S., “Dynamic texture recognition,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 58-63, 2001. Article (CrossRef Link)

[30] R.Peteri, and D.Chetverikov, “Dynamic Texture Recognition Using Normal Flow and Texture Regularity,” in *Proc. of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA 2005)*, Estoril, Portugal, pp.223-230, 2005. Article (CrossRef Link)

[31] D.Chetverikov, and R.Péteri, “A Brief Survey of Dynamic Texture Description and Recognition,” in *Proc. of 4th Int. Conf. on Computer Recognition Systems*, Poland, pp.17-26, 2005. Article (CrossRef Link)

[32]

[33] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” *VS-PETS*, 2005. Article (CrossRef Link)

[34] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *International Journal of Computer Vision*, 2001. Article (CrossRef Link)

[35] Wang Y, Hu S, “Chaotic features for dynamic textures recognition[J],” *Soft Computing*, pp. 1-13, 2015. Article (CrossRef Link)

Xinbin Luo received his MS degree in control theory and control engineering from University of Electronic Science and Technology, China, in 2007. He is currently pursuing his Ph.D. degree in the School of Aeronautics and Astronautics, Shanghai Jiaotong University, China. His research interests are in the area of image processing and analysis algorithms for computer vision inspection system, R&D for industrial automation instrument based on optical image technology.

Shan Fu is a professor in the School of Aeronautics and Astronautics at Shanghai Jiao Tong University. He obtained his first degree in electronic engineering from the Northwestern Polytechnic University in 1985, and PhD from Heriot-Watt University in 1995. His long-time research interest is in the area of computer vision/image processing and related system development, which has been closely linked to engineering/industry applications, such as computerized visual inspection and metrology, experimental mechanics, and structural material engineering.

Yong Wang received his Ph.D. degree 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.