Dynamical Textures Modeling via Joint Video Dictionary Learning

Xian Wei, *Student Member, IEEE,* Yuanxiang Li, *Member, IEEE,* Hao Shen, *Member, IEEE,* Fang Chen, Martin Kleinsteuber, *Member, IEEE,* and Zhongfeng Wang, *Fellow, IEEE*

***Abstract*—Video representation is an important and challenging task in the computer vision community. In this paper, we consider the problem of modeling and classifying video sequences of dynamic scenes which could be modeled in a dynamic textures (DT) framework. At first, we assume that image frames of a moving scene can be modeled as a Markov random process. We propose a sparse coding framework, named joint video dictionary learning (*JVDL*), to model a video adaptively. By treating the sparse coefficients of image frames over a learned dictionary as the underlying "states", we learn an efficient and robust linear transition matrix between two adjacent frames of sparse events in time series. Hence, a dynamic scene sequence is represented by an appropriate transition matrix associated with a dictionary. In order to ensure the stability of *JVDL*, we impose several constraints on such transition matrix and dictionary. The developed framework is able to capture the dynamics of a moving scene by exploring both sparse properties and the temporal correlations of consecutive video frames. Moreover, such learned *JVDL* parameters can be used for various DT applications, such as DT synthesis and recognition. Experimental results demonstrate the strong competitiveness of the proposed *JVDL* approach in comparison with state-of-the-art video representation methods. Especially, it performs significantly better in dealing with DT synthesis and recognition on heavily corrupted data.**

***Index Terms*—Dynamic textures modeling, sparse representation, dictionary learning, linear dynamical systems.**

I. Introduction

TEMPORAL or dynamic textures (DT) are video sequences that exhibit spatially repetitive and certain stationarity properties in time. This kind of sequences is typically videos of processes, such as moving water, smoke, swaying trees, moving clouds, or a flag blowing in the wind. Furthermore, consistent spatio-temporal motion, such as facial expressions, orderly pedestrian crowds, and vehicular traffic, can be seen as a generalization of DT. Study and analysis of DT attracts both theoretical and practical research efforts, such as video modeling [1], [2], DT segmentation [3], video recognition [4], object tracking [5], saliency (e.g., emergency) detection [6] and video synthesis [2]. However, the continuous change in the shape and appearance of a dynamic texture makes the application of traditional computer vision algorithms very challenging. Thus, finding an appropriate spatio-temporal generative representation model that can explore the evolution of the dynamic textured scenes, is the key to the success of many DT applications.

In the past several decades, various approaches have been proposed for modeling and synthesizing video sequences of dynamic textures [1], [7], [8], [9], [2], [3], [10]. Among them, one classical approach is to model dynamic scenes via the optical flow [1]. However, such methods require a certain degree of motion smoothness and parametric motion models. Non-smoothness, discontinuities, and noise inherence to rapidly varying, non-stationary DTs (e.g., fire) pose a challenge to develop optical flow based algorithms. Another technique, called particle filter [11], models the dynamical course of DTs as a Markov process. A reasonable assumption in DT modeling is that each observation is correlated to an underlying latent variable, or "state", and then derives the parameter transition operator between these states. Some approaches directly treat each observation as a state, and then focus on transitions between the observations in the time domain, cf. [9], [7], [8]. For instance, the method in [9] describes this transition process as a generative probabilistic model from one frame to another, and methods in [7], [8] construct a spatio-temporal autoregressive model (STAR) or a position affine operator for this transition. However, since natural images often have complex statistical structure with unknown distribution, they are difficult to be explicitly parameterized. Therefore, some machine learning techniques, such as linear smooth regression, may not be directly used to model the transition of consecutive raw images.

Alternatively, representation-based models capture the underlying dynamics of the observations by representing the observations into a novel state space, where the transition model is constructed. By projecting the observations onto a low-dimensional subspace via principle component analysis (PCA), G. Doretto et al. [12], [2] model the evolution of the dynamic textured scenes as a linear dynamical system (LDS) under a Gaussian noise assumption. As a popular method in dynamic textures, LDS and its derivative algorithms (e.g., kernel LDS [13] and tensor LDS [14]) have been successfully used for various dynamic texture applications [2], [12], [13], [10], [14]. However, constraints are imposed on the types of motion and noise that can be modeled in LDS. For instance, it is sensitive to input variations due to various noises. Especially, it is vulnerable to non-Gaussian noise, such as missing data or occlusion of the dynamic scenes. Moreover, stability is also

X. Wei, Y. Li and F. Chen are with School of Aeronautics & Astronautics, Shanghai Jiao Tong University, Dongchuan Road 800, 200240 Shanghai, China. X. Wei is also with Quanzhou Institute of Equipment Manufacturing, Haixi Institutes, Chinese Academy of Sciences, 362200 Quanzhou, China. H. Shen and M. Kleinsteuber are with Department of Electrical and Computer Engineering, Technische Universität München, Arcisstr. 21, 80333 Munich, Germany. M. Kleinsteuber is also with Mercateo AG, Fürstenfelder Str. 5, 80331 Munich. Z. Wang is with Integrated Circuits and Intelligent Systems Lab, Nanjing University, Hankou Road 22, 210093 Nanjing, China.

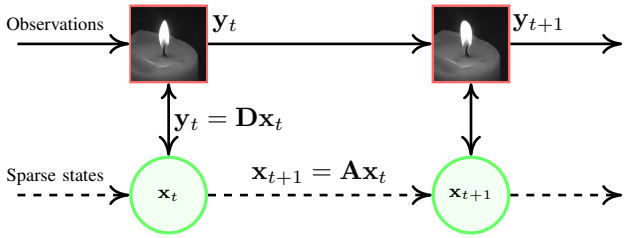


Fig. 1. Pipeline of the proposed *JVDL* model. Therein, $\mathbf{y}_t$, $\mathbf{x}_t$, $\mathbf{D}$ and $\mathbf{A}$ denote the t^{th} observation, its hidden "state" or feature, the dictionary, and the state transition matrix, respectively.

a challenging problem for LDS [15]. Additionally, another possible challenge is that such a LDS may suffer a weak data reconstruction when observations' first several largest singular values are not dominant.

To tackle these challenges, the approach in this paper is to explore an alternative method to model the DTs by appealing to the principle of sparsity. Instead of using the Principle Components (PCs) as the transition "states" in LDS, sparse coefficients over a learned dictionary are imposed as the underlying "states". In this way, the dynamical process of DTs exhibits a transition course of corresponding sparse events. These sparse events can be obtained via a recent technique on linear decomposition of data, called dictionary learning [16], [17], [18]. Formally, these sparse representations $\mathbf{x} \in \mathbb{R}^k$ to a signal $\mathbf{y} \in \mathbb{R}^m$, can be described as

$$\mathbf{y} = \mathbf{D}\mathbf{x}, \tag{1}$$

where $\mathbf{D} \in \mathbb{R}^{m \times k}$ is a dictionary, and $\mathbf{x}$ is sparse, i.e., most of its entries are zero or small in magnitude. That is, the signal $\mathbf{y}$ can be sparsely represented only using a few elements from the dictionary $\mathbf{D}$.

Based on the sparse factorization of Eq. (1), our goal is to find an efficient and robust linear transition matrix between two adjacent frames of sparse representations in time series. With the aim of making the state transition stable and adapt to the sparse structures of image pairs from such adjacent frames, we enforce this linear transition matrix with moderate determinant and bounded largest eigenvalue. In this work, we start with a brief review of the dynamic textures model from the viewpoint of convex ℓ_2 optimization, and then deduce a combined regression associated with several regularizations for a joint process—"state extraction" and "state transition". Then we treat the solution of the above combined regression as a joint dictionary learning problem, which can achieve two distinct yet tightly coupled tasks— efficiently reducing the dimensionality via sparse representation and robustly modeling the dynamical process. In the rest of the paper, we refer to such a proposed model as joint video dictionary learning method (*JVDL*), i.e., simultaneously learn a dictionary and a sparse transition matrix to represent a video sequence. The pipeline is summarized in Fig. 1.

With the DT model at hand, this paper also focuses on how to incorporate such a model into several video processing applications, such as synthesis, denoising and recognition on DT sequences. Note that, video synthesis and denoising could be achieved directly via basic *JVDL* model. Then, we are interested in the problem of categorization of DT sequences, i.e., identifying which class a query DT sequence belongs to. We propose a discriminative *JVDL* model that learns uniform *JVDL* parameters for each class, i.e., a dictionary associated with a transition matrix, which minimize the state transition error for intraclass DTs but maximize the state transition error for interclass DTs.

A preliminary work about modeling DTs in sparse domain was presented in [19]. It has been successfully used for DT sequence synthesis. But the full investigation of the *JVDL* model has not been systematically conducted. In this work, we fully investigate the potential of the *JVDL* model in DT modeling and its various DT applications, i.e., synthesis, denoising and classification. In particular, we first introduce several constraints to improve the stability of basic *JVDL* model. We then extend such a *JVDL* model to the application of DTs classification, called *JVDL* classifier. Our experiments have verified that the *JVDL* classifier can capture the discrimination of DTs.

The rest of this paper is organized as follows. Section II provides an overview of related work. In Section III, we start with a brief review of both linear dynamical systems and sparse representation. In Section IV, we construct a generic cost function for learning both the dictionary and the linear transition operator, and develop a geometric gradient descent algorithm on the underlying smooth manifold. Two classification algorithms are developed in Section V. Numerical experiments on several applications of the proposed model are discussed in Section VI. Finally, conclusions and outlooks are given in Section VII.

II. Related Work

Modeling dynamic scenes for synthesis and automatic classification on DT sequences has been widely studied due to its importance in various video processing applications. One basic learning scheme for DT applications is performed with separated two stages. Most of the existing DT synthesis or classification methods first use spatio-temporal generative models [2], [20], [21] or local descriptors [4], [22], to describe the dynamics of a DT sequence or its spatio-temporal patches, with the task of synthesis or classification as an after-thought.

A. Modeling Dynamic Textures for Synthesis

Methods in [23], [7], [2], [22] showed that once a generative DT model was at disposal, it could generate a longer or an infinite DT sequence. By regarding the video sequence as a collection of repetitive dynamic patterns in time, authors in [9], [8], [22] recognized individual frames of a video texture was repeated from time to time. The key of these works is to find two matching frames, with high similarity but not consecutive, for dynamic transition. Then, the video synthesis is achieved by repetitively inserting the acquired matching frames into the original image sequence in suitable locations. These methods are nonparametric, with the aim of finding transition relations between frames, but cannot generate new frames that are not appeared in existing sequence.

Differently, various methods have been proposed to conduct the transition process using an explicit parametric model, cf. [1], [7], [24], [2]. For example, by exploring the physical mechanism underlying the natural process of DTs, DT synthesis were studied in [23], [25]. Since physics-based parametric models often suffer from the high computational complexity and the weak extensibility, i.e., often highly customized for particular textures, methods in [7], [24], [2] do not focus on modeling the physical mechanism of DTs, but the texture dynamics of images, called *image-based* parametric methods. The latter are more flexible and easier to represent different dynamic textures by changing the parameters, such as the STAR in [7] and the stochastic motion model in [26]. Recently, many *image-based* parametric methods have been conducted by transforming the observations into representation space, where the dynamics are more convenient to be captured. Then, the synthesis is achieved by performing learned dynamics transition models. For example, LDS and its derivatives projected the observations into a lower-dimensional subspace via an undercomplete orthogonal dictionary, cf. [2], [14]. Furthermore, methods in [24], [27], [28] represent each observation as a superposition of bases selected from a fixed overcomplete dictionary, such as Fourier bases, LoG, wavelets, and Gabor bases. Our work extends the fixed dictionary to an adaptively learned dictionary, which shows more power on data expressiveness. Moreover, jointly learning dynamic transition in sparse domain has more advantages on global optimization, in comparison with aforementioned two separate-stage learning methods.

B. Modeling Dynamic Textures for Classification

Many classical approaches for dynamic texture recognition are developed based on the spatio-temporal generative models that describe the global DT sequence [7], [26]. By modeling a DT sequence as the output of an LDS, a discrimination measurement space between the model parameters of two LDSs is defined, such as a distance space or a kernel space between two LDSs, cf. [12], [6], [13]. Finally, building upon such a discrimination measurement space, classifiers such as Nearest Neighbors (NN) or Support Vector Machines (SVMs) can be learned, and hence used to classify a query video sequence, cf. [12], [13], [29], [5]. Similarly, LDS-based methods, by describing each dynamic motion as a spatio-temporal representation of the optical flow, the motion recognition was recast as classifying a set of spatio-temporal flow models of motion events, cf. [20], [12], [1]. These methods model each video as a whole, and perform poorly when the DT sequences are taken under environmental changes or viewpoint changes. To address this challenge, in contrast to describing a DT by generative systems, various methods have been proposed to describe DT sequences by calculating the invariant statistics of local DT features. The typical local feature descriptors include local binary pattern (LBP) [4], [30], dynamic fractal spectrum (DFS) [31], wavelet-based multi-fractal spectrum (WMFS) [32], Gaussian derivative filters [33], optical flow estimation [34], and spatio-temporal transforms [24], [35]. By regarding LDS as a local descriptor, researchers in [21], [36] proposed to model each video sequence with a collection of LDSs, i.e., each one described a small spatio-temporal patch extracted from the video, called bag-of-systems (BoS) representation. The similar works were also presented in [37], [38]. The local descriptors based methods show more advantages on translation-invariance and view-point variations, but they are not capable of capturing longer-term motion dynamics of the texture process.

Recently, sparse representation over a redundant dictionary has been verified as an efficient technique to solve computer vision problems, such as image denoising (including the Gaussian or non-Gaussian noise) [16], [17], [39], and classification [40], [41], [42], [43]. By taking advantage of such a benefit, some methods were proposed to learn dictionary and sparse coefficients in the matrices space of LDSs, cf. [38], [44], [45]. The theoretical support of these methods is from the study in [46], which showed that the parameters of LDSs could be embedded as points on a finite-dimensional Grassmann manifold. Thus, a wide variety of LDS-based video processing problems, e.g., classification and segmentation, can be recast as statistical inference problems on the Grassmann manifolds. With such an embedding, the method in [38] treated the whole LDSs as a dictionary and represented each LDS as a sparse vector over such a dictionary. More recently, authors in [47], [44] proposed an extrinsic dictionary learning algorithm for data points on Grassmann manifolds by embedding the manifolds into the space of symmetric matrices. Inferred by methods in [47], [44], the work in [45] directly performed sparse coding and dictionary learning on the space of extended LDSs. The success of aforementioned methods depends on an assumption that the LDS could capture the underlying dynamics of DTs. With the LDSs at hand, the learning task of classification on the space of infinite LDSs is an after-thought. However, this assumption is not theoretically guaranteed. Moreover, compared with the proposed methods that jointly learn the dictionaries and the classification task, such a two separate-stage learning scheme may fail to achieve a global optimization on classification. The other related methods in [48], [49] proposed to directly learn dictionary on image patches, e.g., the method in [48] constructed a 3D orthogonal dictionary to model each DT sequence. Hence, the classifiers are constructed on learned sparse coefficients. But, they represent each DT as a short-length sparse vector, which cannot capture the temporal dynamics. By jointly learning discriminative dictionaries and transition operators between consecutive sparse "states", the proposed methods are able to explore the discrimination hidden in both spatial and temporal dynamics of video frames.

III. Linear Dynamical Systems and Sparse Coding

We start with an introduction to notations and definitions used in the paper. In this work, vectors are denoted by bold lower case letters and matrices by upper case ones. We denote by $(\cdot)^\top$ the matrix transpose, $\mathbf{I}_n$ the $n \times n$-identity matrix, and, $\mathbf{v}_i$ the i^{th} column of matrix $\mathbf{V}$. By $\mathbf{E}_{ij}$, we denote a matrix whose i^{th} entry in the j^{th} column is equal to one, and all the rests are zero.

In this section, we recall some facts about both linear dynamical systems model and sparse representation.

A. Linear Dynamical Systems

Let us denote a given sequence of $(T+1)$ frames by $\mathbf{Y} := [\mathbf{y}_0, \ldots, \mathbf{y}_T] \in \mathbb{R}^{m\times(T+1)}$, where the time is indexed by $t = 0, 1, \ldots, T$. The evolution of a LDS is often described by the following two equations

$$\begin{cases} \mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{w}_t \\ \mathbf{y}_t = \mathbf{D}\mathbf{x}_t + \mathbf{v}_t, \end{cases} \tag{2}$$

where $\mathbf{y}_t \in \mathbb{R}^m$, $\mathbf{x}_t \in \mathbb{R}^k$, $\mathbf{w}_t \in \mathbb{R}^k$ and $\mathbf{v}_t \in \mathbb{R}^m$ denote the observation, its hidden "state" or feature, "state" noise, and observation noise, respectively. The system is described by the dynamics matrix $\mathbf{A} \in \mathbb{R}^{k\times k}$, and the modeling matrix $\mathbf{D} \in \mathbb{R}^{m\times k}$. Here we are interested in estimating the system parameters $\mathbf{A}$ and $\mathbf{D}$, together with the hidden states, given the sequence of observations $\mathbf{Y}$.

The problem of learning the LDS in Eq. (2) can be considered as a coupled linear regression problem [15]. Let us denote $\mathbf{X} = [\mathbf{x}_0, \ldots, \mathbf{x}_T] \in \mathbb{R}^{k\times(T+1)}$, $\mathbf{X}_0 = [\mathbf{x}_0, \ldots, \mathbf{x}_{T-1}] \in \mathbb{R}^{k\times T}$, and $\mathbf{X}_1 = [\mathbf{x}_1, \ldots, \mathbf{x}_T] \in \mathbb{R}^{k\times T}$. The system dynamics and modeling matrix are expected to be obtained by solving the following minimization problem,

$$\min_{\mathbf{A},\mathbf{D},\mathbf{X}} \left\|\mathbf{X}_1 - \mathbf{A}\mathbf{X}_0\right\|_F^2, \quad s.t.\ \left\|\mathbf{Y} - \mathbf{D}\mathbf{X}\right\|_F^2 \leq \varepsilon, \varepsilon \in \mathbb{R}^+. \tag{3}$$

Therein, $\|\cdot\|_F$ denotes the Frobenius norm of matrices.

Conventional LDS methods [2], [5], [15] often encode the observations as an *undercomplete* representation over a dictionary with orthogonal columns, i.e., $\mathbf{Y} := \mathbf{D}\mathbf{X}$, with $\mathbf{D} \in St(k, m)$. Here, $St(k, m)$ denotes the Stiefel manifold defined by $St(k, m) := \left\{\mathbf{V} \in \mathbb{R}^{m\times k} | \mathbf{V}^\top\mathbf{V} = I_k\right\}$. Hence, the solutions to the problem in Eq. (3) rely on the so called singular value decomposition (SVD) of observations, i.e., $\mathbf{Y} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ with $\mathbf{U} \in St(k, m)$ and $\mathbf{V} \in St(k, T+1)$. Therein, $\mathbf{\Sigma} = \mathrm{diag}\{\delta_1, \cdots, \delta_k\}$ contains the first k largest non-negative singular values with $k < m$. Finally, one can obtain suboptimal estimates of $\mathbf{D}$ and $\mathbf{X}$ as follows [15]:

$$\tilde{\mathbf{D}} = \mathbf{U} \qquad \text{and} \qquad \tilde{\mathbf{X}} = \mathbf{\Sigma}\mathbf{V}^\top \tag{4}$$

with $\mathbf{Y} \approx \tilde{\mathbf{D}}\tilde{\mathbf{X}}$. The estimate of $\mathbf{A}$ is

$$\tilde{\mathbf{A}} = \tilde{\boldsymbol{X}}_1\tilde{\boldsymbol{X}}_0^\dagger, \tag{5}$$

where $\dagger$ denotes the Moore-Penrose inverse.

B. Sparse Coding

Given a set of data samples $\mathbf{y}_t \in \mathbb{R}^m$, the aim of sparse coding is to find a collection of atoms $\mathbf{d}_i \in \mathbb{R}^m$ such that each data sample can be approximated by a linear combination of only a few of the atoms $\{\mathbf{d}_i\}$. According to [50], the atoms can be interpreted as underlying factors that are responsible for explaining the discrepancy in the data set. In other words, sparse coding generates sparsely distributed representations of data with respect to the specific atoms.

The collection of atoms (often as columns in a matrix) is called a dictionary $\mathbf{D} \in \mathbb{R}^{m\times k}$, leading to the model in Eq. (1) for each $\mathbf{y}_t, t = 0, 1, \cdots, T$. In this work, we restrict each column $\mathbf{d}_i \in \mathbb{R}^m$ of $\mathbf{D}$ to have unit norm, i.e.,

$$\mathcal{S}(m, k) := \left\{\mathbf{D} \in \mathbb{R}^{m\times k} \middle| \operatorname{rank}(\mathbf{D}) = \omega, \|\mathbf{d}_i\|_2 = 1\right\}, \tag{6}$$

which is a product manifold of $(m-1)$-dimensional unit spheres with $\omega := \min(m, k)$. Constraint $\mathbf{D} \in \mathcal{S}(m, k)$ is commonly employed in various dictionary learning procedures to avoid the scale ambiguity problem, cf. [39], [17], [18].

Once a dictionary is given, there are several ways of finding the sparse representation. If sparsity is measured by employing the ℓ_1-norm, a solution to the Lasso problem [51], i.e.,

$$\mathbf{x}_\mathbf{y}^*(\mathbf{D}) := \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^k} \tfrac{1}{2}\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1 \tag{7}$$

yields a convenient way to obtain the sparse representation. The regularization parameter $\lambda \in \mathbb{R}^+$ weighs the sparsity measurement against the reconstruction error. Solutions to the Lasso problem in Eq. (7) share a convenient fact that, under certain assumptions, there exists a closed from expression.

Let us denote the set of indexes of the non-zero entries of the solution $\mathbf{x}^* = [\boldsymbol{\varphi}_1^*, \ldots, \boldsymbol{\varphi}_k^*]^\top \in \mathbb{R}^k$ by

$$\Lambda := \{i \in \{1, \ldots, k\} | \boldsymbol{\varphi}_i^* \neq 0\}. \tag{8}$$

and by $S := |\Lambda|$ the cardinality of Λ or the sparsity of $\mathbf{x}^*$. Then the solution $\mathbf{x}^*$ has a closed-form expression as

$$\mathbf{x}_\mathbf{y}^*(\mathbf{D}) := \left(\mathbf{D}_\Lambda^\top\mathbf{D}_\Lambda\right)^{-1}\left(\mathbf{D}_\Lambda^\top\mathbf{y} - \lambda\mathbf{s}_\Lambda\right), \tag{9}$$

where $\mathbf{s}_\Lambda \in \{\pm 1\}^{|\Lambda|}$ carries the signs of $\mathbf{x}_\Lambda^*$, $\mathbf{D}_\Lambda$ is the subset of $\mathbf{D}$ in which the indexes of atoms (rows) fall into support Λ. With a reasonable assumption that the dictionary $\mathbf{D}$ is suitably incoherent and $\left(\mathbf{D}_\Lambda^\top\mathbf{D}_\Lambda\right)^{-1}$ holds, the solution $\mathbf{x}_\mathbf{y}^*(\mathbf{D})$ associated with these regularized functions shares an algorithmically convenient property of being locally twice differentiable with respect to both $\mathbf{D}$ and $\mathbf{y}$, [52], [40], [18].

IV. Joint Video Dictionary Learning

In this section, we develop a joint dictionary learning framework for modeling dynamic textures sequence.

A. A Dictionary Learning Model for Dynamic Scene

In our approach, we assume that all observations $\mathbf{y}_t$ admit a sparse representation with respect to an unknown dictionary $\mathbf{D} \in S(m, k)$, i.e.,

$$\mathbf{y}_t = \mathbf{D}\mathbf{x}_t, \qquad \text{for all } t = 0, 1, \ldots, T, \tag{10}$$

where $\mathbf{x}_t \in \mathbb{R}^k$ is sparse.

Then, by adopting the common sparse coding framework to the problem in Eq. (3), we have the following minimization problem

$$\min_{\mathbf{A},\mathbf{D},\mathbf{X}} \left\|\mathbf{X}_1 - \mathbf{A}\mathbf{X}_0\right\|_F^2 + \mu_1\left\|\mathbf{Y} - \mathbf{D}\mathbf{X}\right\|_F^2 + \mu_2\|\mathbf{X}\|_1, \tag{11}$$

with $\mathbf{D} \in \mathcal{S}(m, k)$, $\mathbf{A} \in \mathbb{R}^{k\times k}$, $\mathbf{X} \in \mathbb{R}^{k\times(T+1)}$, $\mu_1 \in \mathbb{R}^+, \mu_2 \in \mathbb{R}^+$. The parameter $\mu_2 \in \mathbb{R}^+$ weighs the sparsity measurement against the two residual terms.

Solving the minimization problem as stated in Eq. (11) is a very challenging task. In this work, we employ an idea similar

to *subspace identification methods* [53], [15], which treat the state as a function of $\mathbf{D}$.

Here, we confine ourselves to the sparse solution of a Lasso problem, as Eq. (7). By a slight modification of notations, we denote by $\mathbf{x}^*_{\mathbf{y}_t}(\mathbf{D})\colon \mathbf{y}_t \mapsto \mathbf{x}_{\mathbf{y}_t}$ the sparse solutions to Eq. (7) with specific $\mathbf{D}$. We further denote by $\mathbf{Y}_0 = [\mathbf{y}_0, \ldots, \mathbf{y}_{T-1}]$ and $\mathbf{Y}_1 = [\mathbf{y}_1, \ldots, \mathbf{y}_T]$. In this way, we define

$$\mathbf{X}_0(\mathbf{D}) \colon = [\mathbf{x}^*_{\mathbf{y}_0}(\mathbf{D}), \ldots, \mathbf{x}^*_{\mathbf{y}_{T-1}}(\mathbf{D})]. \tag{12}$$

In a similar way, $\mathbf{X}_1(\mathbf{D})$ is defined by

$$\mathbf{X}_1(\mathbf{D}) \colon = [\mathbf{x}^*_{\mathbf{y}_1}(\mathbf{D}), \ldots, \mathbf{x}^*_{\mathbf{y}_T}(\mathbf{D})]. \tag{13}$$

By regarding such sparse events as the underlying "states" of observations, the dynamic course of a moving scene can be modeled as a linear square regression problem with respect to a time-invariant transition matrix $\mathbf{A}$ and a dictionary $\mathbf{D}$, i.e.,

$$\begin{aligned} f\colon \mathbb{R}^{k\times k} \times \mathcal{S}(m,k) &\to \mathbb{R} \\ (\mathbf{A}, \mathbf{D}) &\mapsto \tfrac{1}{2T} \left\| \mathbf{X}_1(\mathbf{D}) - \mathbf{A}\mathbf{X}_0(\mathbf{D}) \right\|_F^2 . \end{aligned} \tag{14}$$

An illustration of such a process is described in Fig. 1.

The linear dynamic system referring to Eq. (14) may suffer from the three aspects as follows: i) The learned linear transition matrix may not adapt to the distribution of sparse "states"; ii) The instability of the learning system; iii) The high coherence of non-orthogonal atoms in dictionary may result in an ambiguity of sparse representation.

With the aim of building a solvable and stable learning procedure for the problem in Eq. (14), in the following, we further regularize the problem by imposing several constraints on $\mathbf{A}$ and $\mathbf{D}$.

1) The Choice of Dictionary: Recalling the fact that the Eq. (9) exists if $\left(\mathbf{D}_\Lambda^\top \mathbf{D}_\Lambda\right)^{-1}$ holds, i.e., $\mathbf{D}_\Lambda$ is full rank for all $S < m$. A critical choice for $\mathbf{D}$ is a set of orthonormal atoms, i.e., $\mathbf{D} \in St(k,m)$ with $k \ll m$, and the problem in Eq. (14) is simply solved by Eq. (4) and Eq. (5). Such a dictionary can efficiently project the observations into an low-dimensional orthogonal subspace, but it may yield a bad approximation for data reconstruction, i.e., $\left\|\mathbf{Y} - \mathbf{D}\mathbf{X}\right\|_F^2$ may exceed the allowable limit. On the other hand, practically, finding a sparse representation over an orthogonal dictionary is often a challenge for some natural images.

Now, we relax the orthogonal constraint on $\mathbf{D}$ to a general $\mathbf{D} \in \mathcal{S}(m,k)$ under appropriate incoherence conditions. Let us define the mutual coherence of $\mathbf{D}$ by

$$\mu(\mathbf{D}) = \max_{1\le i<j\le k} |\mathbf{d}_i^\top \mathbf{d}_j|.$$

In order to prevent solution dictionaries from being highly coherent, we employ a log-barrier function on the scalar product of all dictionary columns to control the mutual coherence of the learned dictionary $\mathbf{D}$, cf. [17], i.e.,

$$\kappa(\mathbf{D}) := -\sum_{1\le i<j\le k} \log(1 - (\mathbf{d}_i^\top \mathbf{d}_j)^2). \tag{15}$$

It is easy to see that a non-zero dictionary $\mathbf{D} \in \mathcal{S}(m,k)$ with $\kappa(\mathbf{D}) = 0$ indicates $\mathbf{D} \in St(k,m)$. Moreover, the constraint $\kappa(\mathbf{D})$ could reduce the linear dependence of atoms in $\mathbf{D}$, which plays a critical role on the stability and the smoothness of sparse solutions in Eq. (7), cf. [54], [17], [18].

2) Stability Analysis: The stability is a desirable characteristic for LDS problems in Eq. (2) and Eq. (3), especially when simulating long sequences from the system in order to generate representative data or infer stretches of missing values.

In general, when ℓ_1 norm is used to measure the sparity in Eq. (7), the prior distribution for the elements of each coefficient vector $\mathbf{x}$ is zero-mean i.i.d. with standard Symmetric Laplace in $\mathbb{R}$, which could be defined as

$$p(\mathbf{x}) = \prod_{j=1}^{k} p(\varphi_j),\ p(\varphi_j) = \frac{\lambda}{2}\exp\{-\lambda|\varphi_j - \mu|\}. \tag{16}$$

where $\mathbf{x} = [\varphi_1, \ldots, \varphi_k]^\top \in \mathbb{R}^k$, $\lambda \in \mathbb{R}^+$ is a scale parameter, and $\mu = 0$ is the location parameter. Let us denote by $\mathbf{x}_\mathbf{y} \sim \mathcal{L}(\mu, \lambda)$ the *Univariate Symmetric Laplace distribution* for $\mathbf{x}_\mathbf{y}$ with parameters $\mu = 0$ and $\lambda \in \mathbb{R}^+$.

Let us consider the sparse representations matrices $\mathbf{X}_0(\mathbf{D})$ and $\mathbf{X}_1(\mathbf{D})$ of the data $\mathbf{Y}_0$ and $\mathbf{Y}_1$. The multidimensional extension of the generative model of Eq. (16) for vectors set $\mathbf{X} := \mathbf{X}(\mathbf{D})$ is straightforward. Here, we adopt the setting for multivariate Laplace (ML) distribution as shown in [55], which defines the formulation of ML distribution as a scale mixture of a multivariate Gaussian given by $\mathbf{x} = \boldsymbol{\mu} + \sqrt{Z}\boldsymbol{\Sigma}^{1/2}\boldsymbol{\theta}$. Therein, $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_k)$, $\boldsymbol{\mu}$ is the mean vector of $\mathbf{X}$, $\boldsymbol{\Sigma}^{1/2} \in \mathbb{R}^{k\times k}$ is a positive definite, i.e., covariance matrix of $\mathbf{X}$, and Z is drawn from a univariate exponential distribution with probability density function (pdf) $p_Z(z) = \lambda\exp(-\lambda z)$. The integrated distribution of $\{\mathbf{x}_t\}$ over the prior distribution $p_Z(z)$ is given by

$$\begin{aligned} p_\mathbf{X}(\mathbf{x}) &= \int_0^\infty p_{\mathbf{X}|Z}(\mathbf{x}|Z=z)p_Z(z)dz \\ &= \int_0^\infty \frac{1}{(2\pi z)^{k/2}}\exp(-\frac{1}{2z}q(\mathbf{x}))p_Z(z)dz \\ &= \frac{2\lambda\mathbf{K}_{(k/2)-1}\left(\sqrt{2\lambda q(\mathbf{x})}\right)}{(2\pi)^{(k/2)}\left(\sqrt{\frac{1}{2\lambda}q(\mathbf{x})}\right)^{(k/2)-1}}, \end{aligned} \tag{17}$$

with $q(\mathbf{x}) = (\mathbf{x}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$, and $\mathbf{K}_d(x)$ denotes the modified Bessel function of the second kind and order k, evaluated at $\mathbf{x}$. In what follows, we will use the notation $\mathbf{X} \sim \mathcal{ML}(\boldsymbol{\mu}, \lambda, \boldsymbol{\Sigma})$ to denote that $\mathbf{X}$ is an ML distributed variable with parameters $\boldsymbol{\mu}$, λ, and $\boldsymbol{\Sigma}$. The model parameters of the Eq. (17) could be estimated using classical maximum-likelihood approach, e.g., iterative EM-type algorithm.

Let matrices $\boldsymbol{\Sigma}_1$, $\boldsymbol{\Sigma}_2$ define the internal covariance structure of the variables of $\mathbf{X}_0$ and $\mathbf{X}_1$, respectively. Now, let

$$\mathbf{X}_1 = \mathbf{A}^\top\mathbf{X}_0 + \boldsymbol{\zeta} \tag{18}$$

with $\boldsymbol{\zeta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}')$, be an arbitrary linear transformation of a $\mathcal{ML}(\boldsymbol{\mu}_1, \lambda_1, \boldsymbol{\Sigma}_1)$ random vector $\mathbf{X}_0$, where $\mathbf{A} \in \mathbb{R}^{k\times k}$. The transformed variable $\mathbf{X}_1$ admits $\mathbf{X}_1 \sim \mathcal{ML}(\boldsymbol{\mu}_2, \lambda_2, \boldsymbol{\Sigma}_2)$ with

$$\begin{cases} \boldsymbol{\Sigma}_2 = \mathbf{A}^\top\boldsymbol{\Sigma}_1\mathbf{A}|\det(\mathbf{A})|^{-(2/k)}, \\ \lambda_2 = \lambda_1|\det(\mathbf{A})|^{(1/k)}, \\ \boldsymbol{\mu}_2 = \mathbf{A}^\top\boldsymbol{\mu}_1 + \boldsymbol{\mu}(\zeta), \end{cases} \tag{19}$$

where $\boldsymbol{\mu}(\boldsymbol{\zeta})$ is the mean vector of $\boldsymbol{\zeta}$ and assumed to be $\mathbf{0}$ in this work. Finding $\widetilde{\mathbf{A}}$ given $\mathbf{X}_0$ and $\mathbf{X}_1$ is triple approximation problems of Eq. (19).

In this work, we assume the length of sequence is sufficiently large. Thus, $\mathbf{X}_0 := \mathbf{X}_0(\mathbf{D})$ and $\mathbf{X}_1 := \mathbf{X}_1(\mathbf{D})$ share the same distribution as sparse coefficients set $\mathbf{X}$. We assume $\mathbf{X} \sim \mathcal{ML}(\boldsymbol{\mu}, \lambda, \boldsymbol{\Sigma})$ with $\boldsymbol{\mu} = \mathbf{0}$. From the definition of $\mathbf{X}_0(\mathbf{D})$ and $\mathbf{X}_1(\mathbf{D})$, it easily infers that $\mathbf{X}_0(\mathbf{D}) \sim \mathcal{ML}(\boldsymbol{\mu}, \lambda, \boldsymbol{\Sigma})$, $\mathbf{X}_1(\mathbf{D}) \sim \mathcal{ML}(\boldsymbol{\mu}, \lambda, \boldsymbol{\Sigma})$. Therefore, the linear transformation satisfies

$$\begin{cases} \boldsymbol{\Sigma} = \mathbf{A}^\top \boldsymbol{\Sigma} \mathbf{A} |\det(\mathbf{A})|^{-(2/k)}, \\ \lambda = \lambda |\det(\mathbf{A})|^{(1/k)}. \\ \boldsymbol{\mu} = \mathbf{A}^\top \boldsymbol{\mu}, \end{cases} \tag{20}$$

Eq. (20) shows that a stable transition process implies that $\det(\mathbf{A}) = 1$ and $\|\mathbf{A}^\top \mathbf{A}\|_2 = 1$ with $\|\cdot\|_2$ denoting the ℓ_2 norm of matrices.

Given data sequence $\{\mathbf{x}_t \in \mathbb{R}^k\}_{t=0}^T$, we hope to learn a stable linearity of expectation for Eq. (18), i.e., the latent variable fitting Eq. (20). Eq. (20) shows that a stable linear transformation requires a moderate $\det(\mathbf{A})$ and a moderate $\|\mathbf{A}^\top \mathbf{A}\|_2$. Given a square matrix $\mathbf{A} \in \mathbb{R}^{k \times k}$, it is known that $\|\mathbf{A}\|_F \geq \|\mathbf{A}\|_2$. Thus we can enforce constraints on $\mathbf{A}$ with the penalty functions

$$h(\mathbf{A}) = \frac{1}{4 \log(k)} \left(\log\left(\eta + \det(\mathbf{A}^\top \mathbf{A})\right)\right)^2, \tag{21}$$

$$\rho(\mathbf{A}) = \frac{1}{2k^2} \|\mathbf{A}\|_F^2, \tag{22}$$

with $\eta \in (0, 1)$ being a small smoothing parameter. $h(\mathbf{A})$ is provided to void the worst case of $\det(\mathbf{A}^\top \mathbf{A})$ being exponentially big.

Let $\{\sigma_i\}_{i=1}^k$ denote the singular values of a transition matrix $\mathbf{A} \in \mathbb{R}^{k \times k}$ in decreasing order of magnitude, and $\sigma(\mathbf{A})$ denotes the largest one. It is known that $\|\mathbf{A}\|_F^2 = \sqrt{\sum_{i=1}^k \sigma_i^2} \geq \sigma(\mathbf{A})^2$. Thus, imposing a penalty as in Eq. (22) could result in a small $\sigma(\mathbf{A})$. On the other hand, $\mathbf{A}$ is expected to be full rank, and the Gram matrix $\mathbf{A}^\top \mathbf{A}$ is positive definite, which implies $\det(\mathbf{A}^\top \mathbf{A}) > 0$. Recalling that $\det(\mathbf{A}^\top \mathbf{A}) = \prod \sigma_i^2$ and $0 < \eta \ll 1$, thus the constraint term in Eq. (21) is imposed to restrict all singular values around 1. Such two constraints are similar, but more critical, to the conventional work in [53], [15], which states that an LDS with dynamics matrix $\mathbf{A}$ is stable if all of $\mathbf{A}$'s eigenvalues have magnitude at most 1.

3) The Objective Function: By combining the regularizers discussed above, we construct the following cost function to jointly learn both the dictionary and the linear transition matrix, i.e.,

$$\begin{aligned} &J \colon \mathbb{R}^{k \times k} \times \mathcal{S}(m, k) \to \mathbb{R} \\ &(\mathbf{A}, \mathbf{D}) \mapsto f(\mathbf{A}, \mathbf{D}) + \gamma_1 \rho(\mathbf{A}) + \gamma_2 h(\mathbf{A}) + \gamma_3 \kappa(\mathbf{D}), \end{aligned} \tag{23}$$

where the weighting factors $\gamma_1, \gamma_2, \gamma_3 \in \mathbb{R}^+$ control the influence of three constraints on the final solution. Our experiments have verified that the regularizers $\rho(\mathbf{A})$ and $h(\mathbf{A})$ ensure solutions of the global cost function J defined in Eq. (23) to be self explanatory to the data, and guarantees stable performance towards the task of learning. On the other hand, such a learned $\mathbf{A}$ could capture the dynamic course of a moving scene and it is the key parameter for dynamic scenes synthesizing and classification, cf. [2], [12].

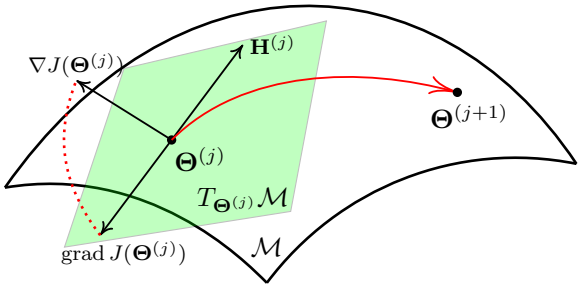


Fig. 2. This figure shows two points $\boldsymbol{\Theta}^{(j)}$ and $\boldsymbol{\Theta}^{(i+1)}$ on a manifold $\mathcal{M}$ together with some required concepts on $\mathcal{M}$. Tangent space (green area): $T_{\boldsymbol{\Theta}}\mathcal{M}$, a real vector space containing all possible directions that tangentially pass through $\boldsymbol{\Theta}$; The search direction (tangent vector) at $\boldsymbol{\Theta}$: $\mathbf{H} \in T_{\boldsymbol{\Theta}}\mathcal{M}$; The Euclidean gradient $\nabla J(\boldsymbol{\Theta})$ and its orthogonal projection onto the tangent space at $\boldsymbol{\Theta}$, called Riemannian gradient: $\operatorname{grad} J(\boldsymbol{\Theta}) \in T_{\boldsymbol{\Theta}}\mathcal{M}$;

B. Optimization Algorithm for JVDL

In this subsection, we employ a gradient descent (GD) algorithm to minimize Eq. (23). Let $\boldsymbol{\mathcal{M}} := \mathbb{R}^{k \times k} \times \mathcal{S}(m, k)$ be a product manifold of a Riemannian submanifold of $\mathbb{R}^{k \times k} \times \mathbb{R}^{m \times k}$, and let $J \colon \boldsymbol{\mathcal{M}} \to \mathbb{R}$ be the differentiable cost function of Eq.(23). The general solution to optimization problem in Eq. (23) on matrix manifold, an element of $\boldsymbol{\mathcal{M}}$, is denoted by $\boldsymbol{\Theta} \in \boldsymbol{\mathcal{M}}, \boldsymbol{\Theta} := (\mathbf{A}, \mathbf{D})$. Some required concepts on $\boldsymbol{\mathcal{M}}$ are depicted in Fig. 2 to alleviate the understanding. For a detailed overview on optimization on matrix manifolds, we refer the interested reader to [56], [17]. Before introducing the GD algorithm on $\boldsymbol{\mathcal{M}}$, we first compute the Riemannian gradients of J with respect to $\mathbf{A}$ and $\mathbf{D}$.

As depicted in Fig. 2, the Riemannian gradient of J is a tangent vector that points in the direction of steepest ascent of J on $\mathcal{M}$. Then, by recalling the geometry of product manifold, we denote the Riemannian gradient of J at $\boldsymbol{\Theta}$ by $\operatorname{grad} J(\boldsymbol{\Theta}) := (\operatorname{grad} J(\mathbf{A}), \operatorname{grad} J(\mathbf{D}))$. In this work, we assume J is globally defined on the whole $\mathbb{R}^{k \times k} \times \mathcal{S}(m, k)$, then, $T_{\boldsymbol{\Theta}}\mathcal{M} := \mathbb{R}^{k \times k} \times T_{\mathbf{D}}\mathcal{S}(m, k)$ with $T_{\mathbf{D}}\mathcal{S}(m, k)$ denoting the tangent space at $\mathbf{D} \in \mathcal{S}(m, k)$. Therefore, $\operatorname{grad} J(\mathbf{A}) := \nabla J(\mathbf{A})$ with $\nabla J(\mathbf{A})$ being the Euclidean gradient on $\mathbf{A}$, and the Riemannian gradient $\operatorname{grad} J(\mathbf{D})$ is simply described as

$$\operatorname{grad} J(\mathbf{D}) := \Pi_{\mathbf{D}}\left(\nabla J(\mathbf{D})\right), \tag{24}$$

with $\nabla J(\mathbf{D})$ being the Euclidean gradient of J with respect to $\mathbf{D}$. Therein, $\Pi_{\mathbf{D}}(\mathbf{Z})$ denotes an orthogonal projection that projects an arbitrary point $\mathbf{Z} \in \mathbb{R}^{m \times k}$ onto $T_{\mathbf{D}}\mathcal{S}(m, k)$, i.e.,

$$\Pi_{\mathbf{D}}(\mathbf{Z}) := \mathbf{Z} - \mathbf{D} \operatorname{ddiag}(\mathbf{D}^\top \mathbf{Z}). \tag{25}$$

Thus, the Riemannian gradient of J at $\boldsymbol{\Theta}$ is computed by

$$\operatorname{grad} J(\boldsymbol{\Theta}) = \left(\nabla J(\mathbf{A}), \Pi_{\mathbf{D}}\left(\nabla J(\mathbf{D})\right)\right), \tag{26}$$

which is degenerated to compute $\nabla J(\mathbf{A})$ and $\nabla J(\mathbf{D})$.

Since all measures of J in Eq. (23) on $\mathbf{A}$ and $\mathbf{D}$ are differentiable, thus, the current key challenge for Eq. (23) is the differentiability of $\mathbf{x}_{\mathbf{y}}(\mathbf{D})$ with respect to $\mathbf{D}$. Let us denote $\mathbf{K} := \mathbf{D}_\Lambda^\top \mathbf{D}_\Lambda$ and $\mathbf{u} := \mathbf{D}_\Lambda^\top \mathbf{y} - \lambda \mathbf{s}_\Lambda$. The first derivative of $\mathbf{x}_{\mathbf{y}}^*(\mathbf{D})$ of Eq. (9) with respect to $\mathbf{D}$ in the direction $\mathbf{H}_{\mathbf{D}} \in T_{\mathbf{D}}\mathcal{S}(m, k)$ is

$$\mathcal{D}\mathbf{x}_{\mathbf{y}}^*(\mathbf{D})\mathbf{H}_{\mathbf{D}} = \mathbf{K}^{-1}\mathbf{H}_{\mathbf{D}}^\top \mathbf{y} - \mathbf{K}^{-1}(\mathbf{D}_\Lambda^\top \mathbf{H}_{\mathbf{D}} + \mathbf{H}_{\mathbf{D}}^\top \mathbf{D}_\Lambda)\mathbf{K}^{-1}\mathbf{u}. \tag{27}$$

Algorithm 1: A GD-JVDL Algorithm.

Input : Given training set $\{\mathbf{y}_t \in \mathbb{R}^m\}_{t=0}^T$, parameters $\gamma_1, \gamma_2, \gamma_3$ and λ;
Output: $(\mathbf{A}^*, \mathbf{D}^*) \in \mathbb{R}^{k\times k} \times \mathcal{S}(m,k)$;
Step 1: Generate initialization for $(\mathbf{A}^{(0)}, \mathbf{D}^{(0)})$, and set $j = -1$;
Step 2: Set $j = j + 1$;
Step 3: Update sparse codes $\mathbf{X}_\mathbf{Y}(\mathbf{D}^{(j)})$ for each $\mathbf{x}_t(\mathbf{D}^{(j)})$ using Lasso in Eq. (7);
Step 4: Update $\mathbf{H}^{(j+1)} \leftarrow -\operatorname{grad} J(\mathbf{A}^{(j)}, \mathbf{D}^{(j)})$;
Step 5: Update $(\mathbf{A}^{(j+1)}, \mathbf{D}^{(j+1)}) \leftarrow (\mathbf{A}^{(j+1)}, \mathbf{D}^{(j+1)}) + \lambda\mathbf{H}^{(j)}$, where λ is computed by employing a backtracking line search, cf. [17]. Project $\mathbf{D}^{(j+1)}$ onto $\mathcal{S}(m,k)$;
Step 6: If $\|\mathbf{H}^{(j+1)}\|$ is small enough, stop. Otherwise, go to Step 2;

Using the shorthand notation, for all $t = 0, 1, \ldots, T$, let Λ_{t+1} be the support of nonzero entries of $\mathbf{x}_{t+1}(\mathbf{D})$, and denote $\mathbf{u}_{t+1} := \mathbf{D}_{\Lambda_{t+1}}^\top \mathbf{y}_{t+1} - \lambda_1 \mathbf{s}_{\Lambda_{t+1}}$, $\Delta\mathbf{x}_{t+1} := \mathbf{x}_{t+1}(\mathbf{D}) - \mathbf{A}_{\Lambda_t}\mathbf{x}_t(\mathbf{D})$, and $\mathbf{q}_t := \mathbf{u}_t \Delta\mathbf{x}_{t+1}^\top$, the Euclidean gradient $\nabla J(\mathbf{D})$ of J with respect to $\mathbf{D}$ is

$$\begin{aligned}\nabla J(\mathbf{D}) = \sum_{t=0}^{T-1} \frac{1}{T}\mathcal{V}\{\big(\mathbf{y}_{t+1}\Delta\mathbf{x}_{t+1}^\top - \mathbf{D}_{\Lambda_{t+1}}\mathbf{K}_{t+1}^{-1}(\mathbf{q}_t + \mathbf{q}_t^\top)\big) \\ \cdot \mathbf{K}_{t+1}^{-1}\} + \frac{1}{T}\mathcal{V}\{\big(\mathbf{D}_{\Lambda_t}(\mathbf{K}_t)^{-1}(\mathbf{A}_{\Lambda_t}\mathbf{q}_t + \mathbf{q}_t^\top\mathbf{A}_{\Lambda_t}^\top) \\ - \mathbf{y}_t(\Delta\mathbf{x}_{t+1})^\top\mathbf{A}_{\Lambda_t}\big)(\mathbf{K}_t)^{-1}\} + \gamma_3\nabla_\kappa(\mathbf{D})\end{aligned}$$

with

$$\nabla_\kappa(\mathbf{D}) = \mathbf{D}\sum_{1\le i<j\le k} \frac{2\mathbf{d}_i^\top\mathbf{d}_j}{1-(\mathbf{d}_i^\top\mathbf{d}_j)^2}(\mathbf{E}_{ij} + \mathbf{E}_{ji}) \tag{28}$$

being the gradient of the logarithmic barrier function Eq. (15). Therein, $\mathcal{V}\{\cdot\}$ denotes the full length vector of sparse coefficients $\{\cdot\}$.

The Euclidean gradient $\nabla J(\mathbf{A})$ is computed as

$$\nabla J(\mathbf{A}) = \sum_{t=0}^{T} \frac{1}{T}\mathbf{x}_{t+1}\Delta\mathbf{x}_{t+1}^\top + \gamma_1\nabla_\rho(\mathbf{A}) + \gamma_2\nabla_h(\mathbf{A}) \tag{29}$$

with

$$\begin{aligned}\nabla_h(\mathbf{A}) &= \frac{\eta}{\log(k)}\mathbf{A}(\eta\mathbf{A}^\top\mathbf{A})^{-1}, \\ \nabla_\rho(\mathbf{A}) &= \frac{1}{k^2}\mathbf{A}.\end{aligned}$$

Then, with the Euclidean gradients $\nabla J(\mathbf{A})$ and $\nabla J(\mathbf{D})$ at hand, the Riemannian gradient of J at $\mathbf{\Theta} := (\mathbf{A}, \mathbf{D})$, i.e., $\operatorname{grad} J(\mathbf{\Theta}) \in T_{(\mathbf{\Theta})}\mathcal{M}$, is computed via Eq. (25) and Eq. (26). Finally, we summarize a gradient descent algorithm for minimizing the function J as defined in Eq. (23), cf. Algorithm 1. In Algorithm 1, $\operatorname{grad} J(\mathbf{\Theta})$ is employed as the gradient direction for updating $\mathbf{A}$ and $\mathbf{D}$.

V. DTs Classification using *JVDL* Model

In the previous section, we proposed a generic regularized cost function to model the evolution of a temporal DT sequence, namely, *JVDL*. In this section, we present one application of the proposed *JVDL* model, to demonstrate its validity for DTs classification.

It is observed that the DTs from the same class exhibit the similar spatial and temporal dynamics, which show strong dissimilarity for DTs from different classes, cf. [48], [57], [14], [34], [58]. In order to capture the similarity of dynamics of intraclass DTs, we propose to learn one unified *JVDL* model for all samples in such a class. At the same time, such learned class-wise *JVDL* parameters are expected to be against the dynamics of DTs outside the class. The key idea behind our development is to minimize the dissimilarity of dynamics of intraclass DTs, and simultaneously maximize the dissimilarity of dynamics of interclass DTs.

In this section, we consider to independently learn one *JVDL* classifier, i.e., one dictionary and one transition matrix, for each class. In what follows, at first, we introduce the *JVDL* classifier that suits for modeling each whole video sequence using a single *JVDL* model, which is called global *JVDL* classifier in the rest of the paper. However, in the practical applications of visual recognition, one issue often challenges most sparse coding based algorithms, i.e., the linear system of Eq. (1) might become prohibitively expensive when the dimensionality of the raw image of input DT is huge. For addressing such a challenge, we then consider to learn one *JVDL* classifier for the small spatio-temporal patches extracted from the DT videos, which is simply called patch-based *JVDL* classifier.

A. Global JVDL classifier

Let us denote by images set $\boldsymbol{\mathcal{Y}} = [\mathbf{Y}_1, \ldots, \mathbf{Y}_n] \in \mathbb{R}^{m\times n\times(T+1)}$ with each $\mathbf{Y}_i$ denoting one DT sequence. The corresponding sparse coefficients are denote by $\boldsymbol{\mathcal{X}} = [\mathbf{X}_1, \ldots, \mathbf{X}_n] \in \mathbb{R}^{k\times n\times(T+1)}$ with each $\mathbf{X}_i$ being one sequence of $(T+1)$ sparse events. We now assume that each training sequence $\{\mathbf{Y}_i \in \mathbb{R}^{m\times(T+1)}\}$ is associated with a indicator vector $z_i \in \mathbb{R}$, which indicates the corresponding class label. Let $c > 1$ denote the number of classes, n_j denotes the number of data samples in the j-th class with $n = \sum_{j=1}^{c} n_j$. Let $\mathcal{S}_{c_j}$ refer to the subset of $\{\mathbf{Y}_i\}_{i=1}^n$ in the j^{th} class.

Let us denote $\mathbf{A}_j$ and $\mathbf{D}_j$ as parameters for modeling samples from the j^{th} class. Minimizing the dissimilarity of intraclass DTs can be read as an optimization problem to minimize

$$E_w^j = \frac{1}{2Tn_j}\sum_{i\in\mathcal{S}_{c_j}}\sum_{t=1}^{T}\left\|\mathbf{x}_{\mathbf{y}_{i,t}}(\mathbf{D}_j) - \mathbf{A}_j\mathbf{x}_{\mathbf{y}_{i,t-1}}(\mathbf{D}_j)\right\|_2^2 \tag{30}$$

with $\mathbf{y}_{i,t}$ being the $(t+1)^{\text{th}}$ frame of the i^{th} DT sequence $\mathbf{Y}_i$. On the contrary, maximizing the dissimilarity of interclass

DTs can be cast as an optimization problem to maximize

$$E_b^j = \frac{1}{2T(n-n_j)} \sum_{i \notin \mathcal{S}_{c_j}} \sum_{t=1}^{T} \left\| \mathbf{x}_{\mathbf{y}_{i,t}}(\mathbf{D}_j) - \mathbf{A}_j \mathbf{x}_{\mathbf{y}_{i,t-1}}(\mathbf{D}_j) \right\|_2^2. \tag{31}$$

By combining Eq. (30) and Eq. (31), learning the j^{th}-class predictive model parameters $\{\mathbf{D}_j, \mathbf{A}_j\}$ could be formulated as the following minimization problem

$$(\mathbf{A}_j, \mathbf{D}_j) := \arg\min \{E_w^j - \gamma_4 E_b^j\} \tag{32}$$

with $\gamma_4 \in \mathbb{R}^+$ being a tuning parameter. We simply set $\gamma_4 = 1$ in our following experiments.

Minimizing (32) could endow the model parameters ($\mathbf{A}_j$ and $\mathbf{D}_j$) with discrimination, but it does not take advantage of the sparse structure of "states" set $\boldsymbol{\mathcal{X}}$. Various works have verified that the sparse coefficients carry rich discriminative information, cf. [40], [18]. In order to explore the useful information from the sparse structure of $\boldsymbol{\mathcal{X}}$, in the following, we improve the classification model in Eq. (32) by imposing a constraint on $\mathbf{A}$.

1) Sparse Transition Matrix: Let us focus on the problem of DTs classification, the stability for sparse state transition in (14) is not necessary. Our goal is to build an efficient mapping between the sparse coefficients of the current and previous images in time, and this mapping could capture the discriminative information hidden in sparse coefficients.

Works in [48], [14], [58] find that there exist strong spatial homogeneity and temporal periodicity in a single moving scene or motion, which implies that the DT patterns from one sequence are repetitive and often show a similar sparse structure over a suitable dictionary. On the other hand, the sparse events $\{\mathbf{X}_i\}_{i=1}^{n_i}$ from the same class are often ideally assumed to share the similar essential sparse structure. Therefore, capturing such a similarity of sparse events of intraclass DTs provides a good way to help DTs classification, and hence the suitable choice of transition matrix $\mathbf{A}$ is sparse. The nonzero support of $\mathbf{A}$ is dominated by the support of nonzero entries in the sparse representations of consecutive images from intraclass DT sequences. In other words, the sparse structure of $\mathbf{A}^j$ is shared by sparse "states" of all DT sequences in the j^{th} class.

Here, we admit this assumption, and enforce the sparsity of each row of $\mathbf{A}$ as minimizing a ℓ_p norm with $0 \leq p \leq 1$. In this work, we use the following term to measure the overall sparsity of $\mathbf{A} := \{\mathbf{a}_{ij}\}$, i.e.,

$$r(\mathbf{A}) = \frac{1}{2k} \sum_{i=1}^{k} \sum_{j=1}^{k} \log(1 + \nu \mathbf{a}_{ij}^2) \tag{33}$$

with $0 < \nu < 1$ being a weighting parameter.

Essentially, for each DT sequence, the sparse transition matrix is expected to capture the dynamics of sparse events in time. Correspondingly, a learned highly row sparse matrix $\mathbf{A}_j$ could push the consecutive sparse vectors $\{\mathbf{x}_t, \mathbf{x}_{t+1}\}$ in j^{th} class tending to the similar permutation of nonzero entries. Therefore, it logically infers that such a sparse transition matrix could capture the similarity hidden in sparse representation sequences from the same class. Simultaneously, such a transition matrix may also promote the discrepancy of structures of interclass sparse events.

2) The Objective Function: By taking advantage of the sparsity constraint on $\mathbf{A}$, i.e., Eq. (33), we modify the optimization problem in Eq. (32) by minimizing

$$\begin{aligned} &\mathcal{L}\colon \mathcal{S}(m,k) \times \mathbb{R}^{k \times k} \to \mathbb{R}, \\ &\quad \mathcal{L}(\mathbf{A}_j, \mathbf{D}_j) := E_w^j - \gamma_4 E_b^j + \gamma_5 r(\mathbf{A}_j), \end{aligned} \tag{34}$$

where $\gamma_5 > 0$ is introduced to promote the sparse structure of $\mathbf{A}_j$. Our experiments have verified that an appropriately sparse $\mathbf{A}$ can significantly improve the results of DTs classification.

3) Classification: For the multi-class classification problem, i.e., $c > 2$, we use the one-against-all or one-against-one strategy to learn $\{\mathbf{A}_j, \mathbf{D}_j\}$. Let us consider one example which adopts the one-against-all strategy. When the training parameters $\{\mathbf{A}_j, \mathbf{D}_j\}_{j=1}^{c}$ are learned, classifying a test DT sequence $\mathbf{Y} := \{\mathbf{y}_t\}_{t=0}^{T}$ can be formulated as finding

$$\text{identity}(\mathbf{Y}) = \underset{j}{\text{argmin}} \sum_{t=1}^{T} \left\| \mathbf{x}_{\mathbf{y}_t}(\mathbf{D}_j) - \mathbf{A}_j \mathbf{x}_{\mathbf{y}_{t-1}}(\mathbf{D}_j) \right\|_2^2,$$

for all $j = 1, 2, \cdots, c$.

B. Patch-based JVDL classifier

Given a set DT sequences of j^{th} class with each sequence $\mathbf{Y} \in \mathbb{R}^{a \times b \times (T+1)}$, $m = a \times b$, we divide it into non-overlapping spatio-temporal volumes of size $p \times p \times \tau$, where p represents the spatial size and τ represents the temporal size. The patch size is set according to the resolution of training sequences to ensure that we utilized the entire video sequence while extracting non-overlapping patches and not disregarding any region. We randomly select n_j patches $\{\widetilde{\mathbf{Y}}_i\}_{i=1}^{n_j}$ from each category for training its sub-dictionary $\mathbf{D}_j$ with the size of $p^2 \times k$. Therefore, for j^{th} class, we learn one dictionary $\mathbf{D}_j$ and n_j sparse transition matrices $\{\mathbf{A}_i\}_{i=1}^{n_j}$ by minimizing

$$\begin{aligned} \frac{1}{2(\tau-1)n_j} \sum_{i=1}^{n_j} \sum_{t=2}^{\tau} &\left\| \mathbf{x}_{\widetilde{\mathbf{y}}_{i,t}}(\mathbf{D}_j) - \mathbf{A}_i \mathbf{x}_{\widetilde{\mathbf{y}}_{i,t-1}}(\mathbf{D}_j) \right\|_2^2 \\ &+ \gamma_6 r(\mathbf{A}_j), \end{aligned} \tag{35}$$

with $\gamma_6 \in \mathbb{R}^+$ and $\widetilde{\mathbf{y}}_{i,t}$ being the $(t+1)^{\text{th}}$ frame of the i^{th} patch sequence $\widetilde{\mathbf{Y}}_i$. We shortly denote the j^{th} class *JVDL* parameters by $\mathcal{M}_j = (\mathbf{D}_j, \{\mathbf{A}\}_{i=1}^{n_j})$.

With the learned *JVDL* parameters $\{\mathcal{M}_j\}_{j=1}^{c}$ at hand, some standard classification methods can be employed. Here, $c \in \mathbb{Z}^+$ denotes the number of classes. In this section, the classification is performed by *JVDL* associated with the sparse representation-based classifier (SRC) [59], [38], called *JVDL-SRC*, which is discussed in detail as follows.

Before performing *JVDL-SRC*, we combine all the sub-dictionaries $\{\mathbf{D}_j\}_{j=1}^{c}$ as a shared dictionary $\mathbf{D}$ with the size of $p^2 \times (ck)$.

Given a query DT sequence $\mathbf{Y}$, we first divide it into N spatio-temporal patches $\widetilde{\mathbf{Y}}_i := \{\widetilde{\mathbf{y}}_{it}\} \in \mathbb{R}^{p^2 \times \tau}$ and for each patch we obtain its sparse coefficients set $\widetilde{\mathbf{X}}_i := \{\widetilde{\mathbf{x}}_{it}\} \in \mathbb{R}^{(ck) \times \tau}$ via performing Eq. (7) with respect to $\mathbf{D}$. Let us denote an operator $\delta^j : \mathbb{R}^{ck} \to \mathbb{R}^{ck}$ be the characteristic

function which selects the coefficients associated with the j^{th} class, cf. [59]. For our learned $\widetilde{\mathbf{x}}$, $\delta^j(\widetilde{\mathbf{x}}) \in \mathbb{R}^{ck}$ denotes the sparse codes of class j, i.e., all entries are set to zero if they do not belong to class j. By using the sparse codes from j^{th} class, we calculate the reconstruction error via

$$R_j(\widetilde{\mathbf{Y}}, \mathbf{D}) = \frac{1}{N\tau} \sum_{i=1}^{N} \sum_{t=1}^{\tau} \|\widetilde{\mathbf{y}}_{it} - \mathbf{D}\delta^j(\widetilde{\mathbf{x}}_{it})\|_2. \tag{36}$$

Similarly, by using $\{\mathbf{A}\}_{i=1}^{n_j}$ from j^{th} class, we approximate the temporal dynamic process by solving the following optimization problem

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \sum_{t=1}^{\tau-1} \|\widetilde{\mathbf{x}}_{i(t+1)} - \sum_{\iota=1}^{L} \boldsymbol{\alpha}_{i\iota} \mathbf{A}_i \widetilde{\mathbf{x}}_{it}\| + \lambda \sum_{i=1}^{N} \|\boldsymbol{\alpha}_i\|_1$$

with $L = cn_j$. In our experiments, in order to reduce the computation cost, we set $L = c(n'_j)$ with $n'_j \in \mathbb{Z}^+$, $n'_j < n_j$.

With the sparse vectors $\{\boldsymbol{\alpha}_{i\iota}\}_{i=1,\iota=1}^{i=N,\iota=L}$ at hand, we calculate the approximate error by

$$R_j(\widetilde{\mathbf{X}}, \mathbf{A}) = \frac{1}{N(\tau-1)} \sum_{i=1}^{N} \sum_{t=1}^{\tau-1} \|\widetilde{\mathbf{x}}_{i(t+1)} - \sum_{i=1}^{L} \delta_j(\boldsymbol{\alpha}_{i\iota}) \mathbf{A}_i \widetilde{\mathbf{x}}_{it}\|_2.$$

Therein, $\delta_j(\boldsymbol{\alpha}_{i\iota})$ keeps the value of $\boldsymbol{\alpha}_{i\iota}$ if $\mathbf{A}_i$ belongs to j^{th} class, $\delta_j(\boldsymbol{\alpha}_{i\iota}) = 0$ otherwise.

Hence, we classify a query DT sequence $\mathbf{Y}$ by

$$\text{identity}(\mathbf{Y}) = \underset{j}{\text{argmin}}\{R_j(\widetilde{\mathbf{Y}}, \mathbf{D}) + R_j(\widetilde{\mathbf{X}}, \mathbf{A})\}.$$

Therein, $\widetilde{\mathbf{Y}}$ and $\widetilde{\mathbf{X}}$ denote the patches set of $\mathbf{Y}$ and the corresponding sparse coefficients set, respectively.

VI. Numerical Experiments for Evaluating the *JVDL* Model

In this section, we carry out several experiments on natural image sequences data to demonstrate the practicality of the proposed algorithm. Our test dataset comprises of videos from several benchmark datasets, and data from internet sources (for instance, YouTube).

A. Datasets

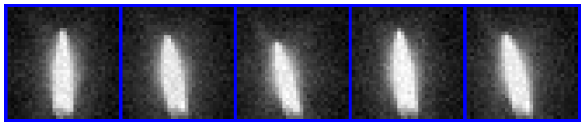
So far, two basic databases have been widely used for DT analysis: the UCLA-DT database [2] and the DynTex database [60].

The UCLA-DT database originally consists of 200 DT sequences with 50 categories, and each category contains 4 video sequences captured from different viewpoints. Its DT sequences have already been pre-processed from their raw form, whereby each sequence is cropped to show its representative dynamics in absence of any static or dynamic background. For each DT sequence, there is only a single DT is present. Each sequence has $T = 75$ frames with $m = 48 \times 48$ pixels.

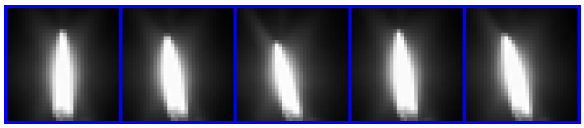
The DynTex database is a large pool of DT sequences and consists of a total of 656 AVI video sequences with the size of 720×576. It aims to serve as a standard database for dynamic texture research and to accommodate the needs for assessing the different research issues, such as texture synthesis, detection, segmentation and recognition, cf. [60].

DynTex++ [37] is a well-designed dataset from original DynTex database and is often used for evaluating DT classification algorithms. It eliminated sequences that contained more than one DT, contained dynamic background, included panning/zooming, or did not depict much motion. The remaining sequences were then labeled as 36 classes. Each class has 100 subsequences of length 50 frames with 50×50 pixels cropped from the original sequences.

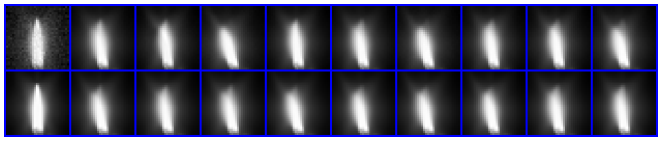
As the color information is not our focus, all images will be normalized to the grayscale between 0 and 1.



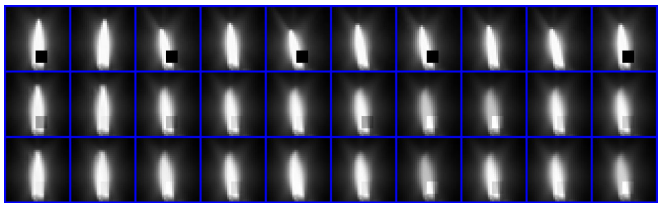
(a) Corrupted original sequence



(b) Reconstructed sequence



(c) Synthesized video using LDS and *JVDL* on DTs with Gaussian noise



(d) Synthesized video using LDS and *JVDL* on DTs with missing data

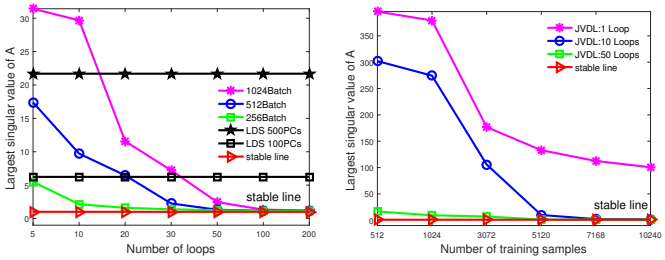
Fig. 3. Reconstruction and synthesizing on the candle scene. (a), (b) are $(t = 1, 64, 128, 512, 1024)th$ frame of the corrupted data by Gaussian noisy and the reconstructed data using *JVDL*, respectively. (c) The top row is the synthesized sequence using LDS (128PCs), and the bottom row is the synthesized sequence using *JVDL* ($(t = 2, 1024, 3072, 5120, \ldots, 20480)th$ frame). (d) The top row is the sequence with missing data. The middle row the synthesized sequence using LDS, and the bottom row is the synthesized sequence using *JVDL*.

B. Dynamical Textures Synthesis

DT synthesis is the process of creating a longer or infinite DT sequence using a video exemplar as input. This can be achieved starting either from a model of a physical phenomenon or from existing video sequences, cf. [23]. This work focuses on the image-based methods, i.e., finding a mathematical model of the video, e.g., LDS or *JVDL*, that can explain the dynamic process of generation of a DT sequence. Once this model is at disposal, it can generate longer video sequences, by just producing new video frames using this

TABLE I
SYNTHESIZING RESULTS ON SEQUENCE OF BURNING CANDLE.

Instance	LDS, (PCs)			*JVDL*, (loops)				
	64	128	256	1	50	100	200	400
Compression rate (%)	6.25	12.50	25.00	1.02	3.29	3.41	3.50	3.55
σ	0.9802	0.9833	0.9849	1.78	1.06	0.9992	0.9994	0.9994
$e_{\mathbf{y}}$	135265	135138	135060	1360	60.2	58.8	56.0	71.3
$e_{\mathbf{x}}$	101.58	135.88	168.95	37500	171.99	75.52	61.96	46.18



(a) The largest singular value of $\mathbf{A}$ for *JVDL* and LDS with increasing loops.

(b) The largest singular value of $\mathbf{A}$ for *JVDL* with increasing number of training samples

Fig. 4. The largest singular value of $\mathbf{A}$ for *JVDL* and LDS. The "stable line" denotes the boundary for stable $\mathbf{A}$, in which the largest singular value is equal to 1. (a). Comparing the largest singular value of $\mathbf{A}$ with increasing loops, on candle video. (b). Largest singular value of $\mathbf{A}$ with increasing training samples, on candle video, $n = 512, 1024, 3072, 5120, 7168, 10240$. Both select the 1024×512 Dictionary

TABLE II
SYNTHESIZING RESULTS ON BURNING CANDLE WITH DIFFERENT DICTIONARY SIZES (50 LOOPS FOR TRAINING, M: MINUTE).

k	64	128	256	512	1024	2048
σ	1.21	1.06	1.02	0.99	0.99	0.99
$\frac{e_{\mathbf{y}}}{T+1}$	13.9	4.2	1.8	1.6	1.5	1.5
$\frac{e_{\mathbf{x}}}{T}$	0.57	0.66	0.68	0.68	0.72	1.16
Training time (m)	2.4	6.6	16.1	22.2	43.0	212.6

model. In what follows, we test our *JVDL* on DT synthesis in comparison with classical LDS method.

Firstly, we show the performance on reconstruction and synthesis with a grayscale video of burning candle from YouTube, which is corrupted by Gaussian noise or occlusion. This video has 10240 frames with size of 32×32, as seen in Fig. 3(a). But in the first experiment, we select its first 1024 frames as training sequence. By applying the classical LDS on image sequence, the dictionary is initialized as the orthogonal projection on LDS, i.e. $\mathbf{D}^{(0)} = \mathbf{U}$ in Eq. (4). Then, $\mathbf{A}^{(0)}$ is initialized by performing Eq. (5). The initial dictionary is with the size of 1024×512. We set $\eta = 10^{-3}$, $\lambda = 0.2$, $\gamma_1 = 0.5$, $\gamma_2 = 0.02$, and $\gamma_3 = 0.0005$. After obtaining $\mathbf{D}$ and $\mathbf{A}$ by minimizing Eq. (23), the synthetic data can be generated easily by $\mathbf{x}_{t+1} = \Gamma_\beta(\mathbf{A}\mathbf{x}_t)$, where Γ_β is the element-wise hard thresholding operator which keeps the elements whose magnitudes are larger than β while setting the rest zeros. We also use a following convex formulation to estimate $\mathbf{x}_{t+1}$, i.e.,

$$\min_{\mathbf{x}_{t+1}} \frac{1}{2}\|\mathbf{x}_{t+1} - \mathbf{A}\mathbf{x}_t\|_2^2 + \lambda\|\mathbf{x}_{t+1}\|_1 + \lambda_2\|\mathbf{x}_{t+1}\|_2$$

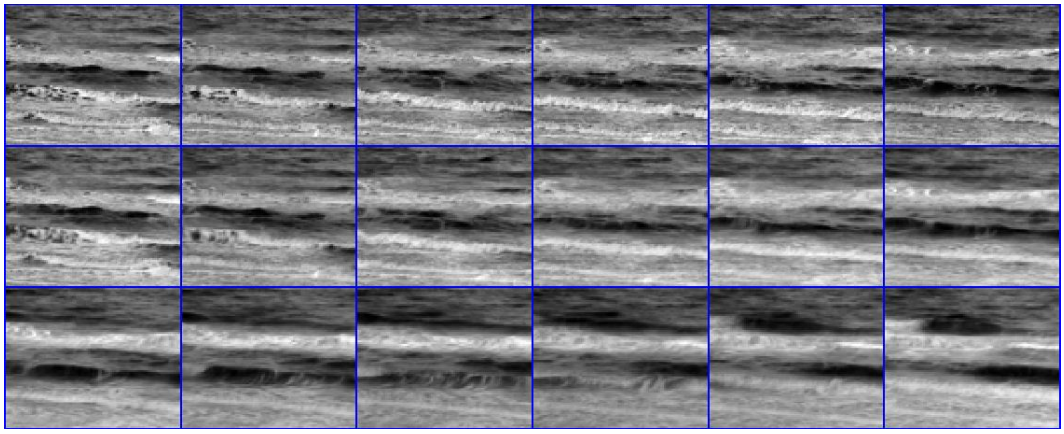
with $\lambda_2 = 0.05$.

Table I shows the DT synthesis performance on burning candle with Gaussian noise. The error pairs $(e_{\mathbf{y}}, e_{\mathbf{x}})$ are defined as $e_{\mathbf{y}} = \sum_t \|\mathbf{y}_t - \mathbf{D}\mathbf{x}_t\|$, $e_{\mathbf{x}} = \sum_t \|\mathbf{x}_{t+1} - \mathbf{A}\mathbf{x}_t\|$, and the largest eigenvalue of $\mathbf{A}$ is denoted by σ. The compression rate for *JVDL* is the sparsity of $\mathbf{x}$ to m, and for LDS is the number of PCs to m. Table I shows that *JVDL* can obtain the stable dynamic matrix $\mathbf{A}$ ($\sigma \leq 1$), smaller compression rate and smaller error $(e_{\mathbf{y}}, e_{\mathbf{x}})$ of cost function (23), by increasing the number of main loops in Algorithm 1. Stability for (2) and (3) will be achieved while the largest singular value is bounded by $\mathbf{1}$, cf. [15]. The main formulation (23) with constraints on $\mathbf{A}$ has enforced stability on $\mathbf{A}$, but doesn't guarantee all the maximum of singular values are less than $\mathbf{1}$. However, this goal can be reached while the training samples are huge or increasing the number of main loops in Algorithm 1, as seen in Fig. 4. When running on a 64-bit computer with double 3.5G HZ processors, Table II compares the results on $T = 5119$ frames of burning candle with different dictionary sizes. The parameters are fixed as above. From Table II, we can see that the larger dictionary sizes can improve the stability and reduce the reconstruction error $e_{\mathbf{y}}$, but increase the transition error $e_{\mathbf{x}}$ and the computing time. $k = 512$ is an efficient choice for the learning process.

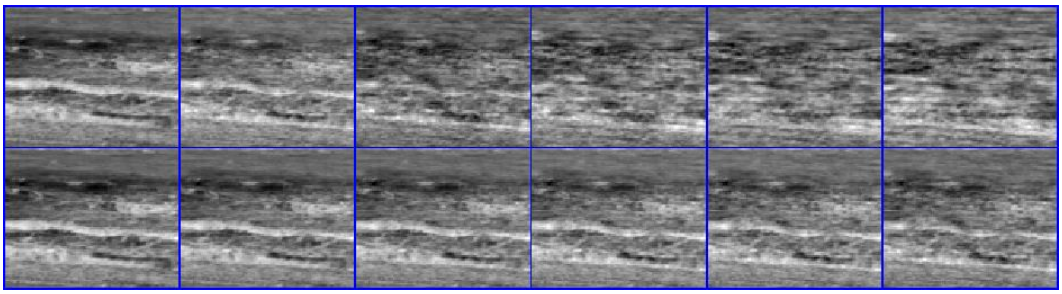
Fig. 3 ($a \sim c$) is the visual comparison between LDS and *JVDL*. *JVDL* performs well on denoising against corruption by Gaussian noise. In the case of occlusion in Fig. 3 (d), random 50 frames of the 1024 burning candle video are corrupted by a (6×7) rectangle. The length of both synthesizing data is 1024, based on the first frame of the burning candle. The experimental results show that 87.01% of the synthesizing data from LDS are corrupted by this rectangle, but only 9.47% are slightly corrupted by this rectangle for *JVDL*. The synthesizing images are shown in the bottom two lines of Fig. 3 (d).

Similar to Fig. 3, we then perform *JVDL* on another DT sequence, namely Tidewater, from DynTex database. The synthesizing experiments are depicted in Fig. 5. Fig 5(a) shows that *JVDL* can model and synthesize such DT sequence. For synthesizing a longer videos in Fig 5(b), compared with LDS, *JVDL* also performs better.

Finally, we investigate the sensitivity of the performance of *JVDL* while varying parameters, i.e., γ_1 and γ_2 in Eq. (23), which reveals the effects of corresponding regularization terms in Eq. (23). The experiments are performed with $T = 5119$, $k = 512$ and 20 loops. Fig. 6 depicts the sensitivity of weighing parameters γ_1 and γ_2 on the largest singular value of $\mathbf{A}$ and the transition error $e_{\mathbf{x}}$. Roughly, the increasing values on γ_1 and γ_2 could improve the stability, but also enlarge the transition error $e_{\mathbf{x}}$. Thus, it is easy to see that a suitable

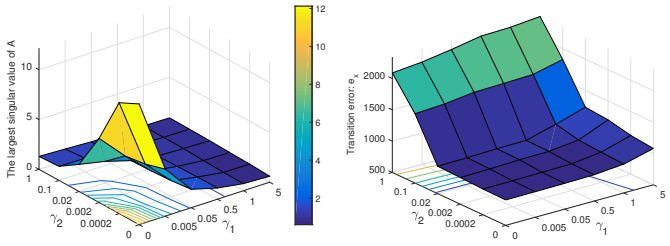


(a) Tidewater and synthesized data (bottom two rows)



(b) synthesized data using LDS and *JVDL*

Fig. 5. Tidewater from DynTex database. (a) (Original) Tidewater sequence ($m = 40 \times 56, T = 3297 - 1$) and reconstructed data via *JVDL* (bottom 2 rows $(t = 1, 21, 41, \ldots, 101)st$ frame). (b)The top row is synthesized sequence using LDS (200PCs), and the bottom row is synthesized sequence using *JVDL*, $((t = 4001, 5351, 6401, \ldots, 8551)st$ frame).



(a) Stability analysis (b) Transition error $e_{\mathbf{x}}$

Fig. 6. Sensitivity in stability and transition error $e_{\mathbf{x}}$ with respect to weighing parameters γ_1 and γ_2.

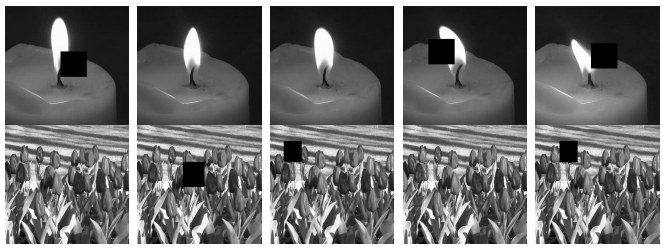


Fig. 7. Examples of some training samples. The top line images set is from the class of "candles" and the bottom one is from the class of "flowers".

choice of γ_1 and γ_2 could improve the performance of the *JVDL* system. For example, $\gamma_1 = 0.5$ and $\gamma_2 = 0.002$ could achieve that i) the largest singular value of $\mathbf{A}$ is bounded by 1, ii) the transition error on $e_{\mathbf{x}}$ is moderate small.

TABLE III
DT RECOGNITION RATES ON THE DYNTEX++ DATABASE WITH OCCLUSION.

Occlusion rate (%)	0	5	15	30
LDS-NN (20PCs)	69.72	45.00	25.14	14.17
LDS-SRC (20PCs)[38]	73.14	56.66	29.04	15.26
MMDL [37]	63.7	50.10	26.45	10.06
KGDL [44] -SVM	92.8	84.25	75.00	55.26
LBP-TOP [4]	89.2	81.44	61.60	31.06
WMFS [32]	88.8	—	—	—
DFS[31]	89.9	—	—	—
EKDL [49]	93.4	—	—	—
JVDL-NN	71.04	65.55	45.12	22.50
global *JVDL*	89.60	88.42	84.85	73.26
JVDL-SRC	91.80	89.64	84.00	69.12

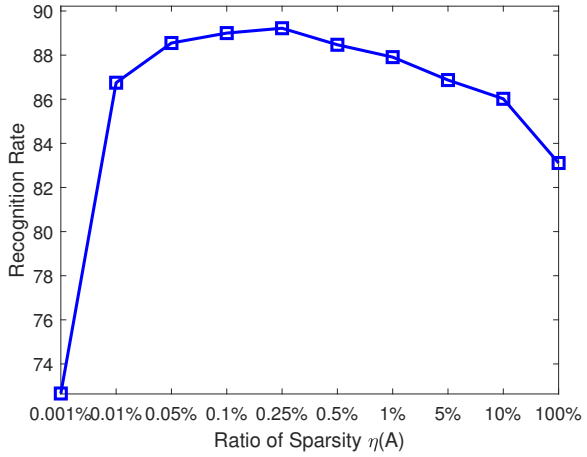
C. Dynamical Textures Classification

In this section, the performance of the proposed *JVDL* is evaluated for DT classification on DynTex++ and UCLA-DT 50 databases. We address the multi-class classification problem with a one-against-all strategy. All recognition experiments are repeated ten times with different randomly selected training and test subsets, and the average of per-class recognition rates is recorded for each run.

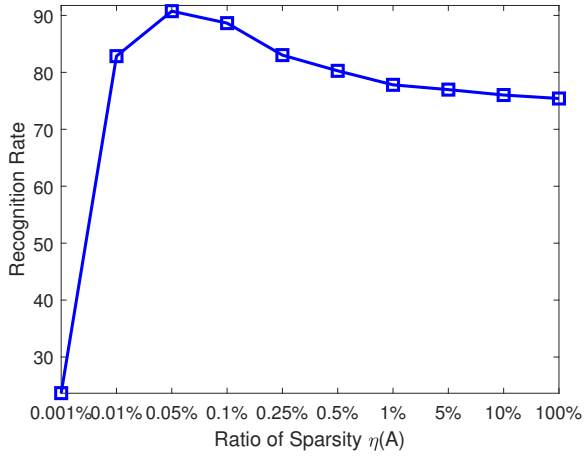
The first classification experiment is applied on DynTex++. We use total 3600 videos with $c = 36$ and randomly choose 50 videos per class for training, and the rest 50 videos for test. For *JVDL-SRC* classifier in Section V-B, we set $\lambda = 0.1$, $\gamma_6 = 0.2$, $p = 10$, $\tau = 25$, and $c \times k = 36 \times 20$. For DT sequences from j^{th} class, we train its sub-dictionary $\mathbf{D}_j$ with

TABLE IV
DT RECOGNITION RATES ON THE UCLA-DT 50 DATABASE WITH MISSING PIXELS.

The rates of missing pixels (%)	LDS-NN (25PCs)	LDS-SRC (25PCs)[38]	MMDL [37]	KGDL [44] -NN	KGDL [44] -SVM	LBP-TOP [4]	KDT [13] -SVM	JVDL -NN	global *JVDL*	JVDL *-SRC*
0% missing pixels	88.08	94.32	99.00	89.24	97.40	96.20	97.5	90.22	96.68	97.83
5% missing pixels	83.72	85.10	89.40	87.52	93.45	90.44	—	88.13	96.12	97.04
15% missing pixels	55.06	64.26	70.64	76.90	86.30	85.65	—	84.34	94.80	94.88
30% missing pixels	17.90	26.28	18.18	70.22	78.00	76.80	—	78.42	91.44	85.68
50% missing pixels	12.88	20.18	14.42	36.54	50.25	58.22	—	58.65	84.20	75.22



(a) global *JVDL*



(b) *JVDL-SRC*

Fig. 8. Applying *JVDL* classifier on DynTex++ with different choices of $\eta_{\mathbf{A}}$.

the size of 100×20. We then combine all the sub-dictionaries as dictionary $\widehat{\mathbf{D}}$ with the size of 100×720. For global *JVDL* classifier in Section V-A, we set $\gamma_5 = 0.5$. We choose the dictionary size as $m = 2500, k = 36 \times 20$.

We compared our method with LDS-NN (Nearest Neighbors) [12], LDS-SRC [38], MMDL (Maximum Margin Distance Learning) [37], KGDL (Kernelized Grassman Dictionary Learning) [44], LBP-TOP (LBP on three orthogonal planes) [4], EKDL (Equiangular Kernel Dictionary Learning) [49], WMFS [32] and DFS[31]. Note that, LDS-NN, *JVDL-NN*, LDS-SRC, *JVDL-SRC* are classification methods that employ NN classifier or SRC classifier to classify the model parameters $(\mathbf{A}, \mathbf{D})$ learned by LDS or JVDL. Following [48], SVM is also used for classification. For KGDL [44] and LBP-TOP [4], we use the softwares that were released by the authors. The software for MMDL [37] is also online available. For WMFS [32], EKDL [49] and DFS[31], we show the results as they were reported in references. In order to evaluate the robustness of *JVDL* to non-Gaussian noise, Table III depicts the recognition results with increasing occlusion rates for test data. Compared to LDS-NN, LDS-SRC, MMDL, KGDL and LBP-TOP, Table III shows the proposed global *JVDL* and *JVDL-SRC* classifiers perform better while the test videos are corrupted by increasing occlusion. Some DT examples corrupted by occlusion are shown in Fig. 7. Table III also demonstrates that the proposed *JVDL-SRC* achieves a competitive performance, which slightly behind the recent records in KGDL-SVM [44] and EKDL [49]. In addition, compared with *JVDL-SRC* classifier, the global *JVDL* classifier achieves a higher classification rate while having a heavy occlusion (e.g., 30% occlusion).

Let us denote by $\eta(\mathbf{A}) = s/(k^2)$ the sparsity ratio of the transition matrix $\mathbf{A}^{k\times k}$ with s denoting the sparsity of $\mathbf{A}$. For global *JVDL* and *JVDL-SRC*, the sparsity of $\mathbf{A}$ is controlled by γ_5 and γ_6, respectively. Fig. 8(a) and Fig. 8(b) depict the classification rates against $\eta(\mathbf{A})$ using global *JVDL* and *JVDL-SRC* classifiers. It is easy to see that the setting of a suitable sparsity of $\mathbf{A}$ can improve the classification rates.

The second classification experiment is performed on UCLA-DT 50 database. Since these 50 classes contain the same DTs at different viewpoints, they can be grouped together to form 9 classes, as in [21]. For *JVDL-SRC* classifier, we set $\lambda = 0.1$, $\gamma_6 = 0.25$, $p = 10$, $\tau = 25$, and $c \times k = 50 \times 20$. The size of each sub-dictionary is set with 100×20. We also combine all the sub-dictionaries as dictionary $\widehat{\mathbf{D}}$ with the size of 100×1000. For global *JVDL* classifier, we set $\gamma_5 = 0.2$. We choose the dictionary size as $m = 2304, k = 50 \times 20$. The missing pixels for an image are another kind of non-Gaussian noise. Note that, the image with $a\%$ missing pixels means that we set the values of random $a\%$ pixels in such an image to zero. Table IV shows the recognition results for DTs corrupted by missing pixels, in comparison with classical methods, i.e., LDS-NN, LDS-SRC, MMDL, KGDL, LBP-TOP and KDT [13]. For KDT, we report the results in literature [13]. For LDS-SRC, we choose 25 PCs for "states" which achieved the best performance recorded in [38]. As shown in Table IV, for the DT classification without missing pixels, *JVDL* classifiers perform better than classical LDS-NN and LDS-SRC, and behind the current record achieved by MMDL. But for classification with increasing missing pixels, *JVDL* based methods (i.e., *JVDL-NN*, global *JVDL* and *JVDL-SRC*) decrease slowly, compared with the dramatically decreasing

performance of LDS-NN, LDS-SRC, MMDL, KGDL and LBP-TOP.

Overall, the results in this subsection suggest that the proposed *JVDL* classifiers can achieve a good performance on DTs classification, especially when the DTs are corrupted by heavy non-Gaussian noise.

VII. Conclusions

This paper has presented a new method, called *JVDL*, to model the dynamic process of DTs. In *JVDL*, the sparse events over a dictionary are imposed as transition "states". A constrained transition matrix is learned to represent each DT sequence. It has been demonstrated that the proposed method is much more robust in synthesis and reconstruction on DTs corrupted by Gaussian noise. To enable the *JVDL* in DT's classification, we proposed two discriminative *JVDL* algorithms associated with a sparse transition matrix. Our experiments have shown that an appropriately sparse transition matrix could well capture the discrimination of DT sequences. Especially, *JVDL* and *JVDL* classifiers become more powerful in the case of test data corrupted by non-Gaussian noise, such as occlusion or missing pixels. For instance, in one test case, the recognition rate of *JVDL-SRC* decreased from 97.83% to 85.68% while conventional LDS-SRC approach decreased from 94.32% to 26.28% when 30% missing pixels occur. One possible future extension is to learn a dictionary for large scale DT sequences based on *JVDL*, for example, using tensor dictionary to model the DT sequence with high resolution. Another potential research direction would be to construct a hierarchical learning scheme, e.g., BoS, on a collection of *JVDL* parameters (i.e., the dictionary matrix and the sparse transition matrix) to promote the task of interest, such as motion classification, detection and segmentation.

Acknowledgments

This work was partially supported by National Natural Science Foundation of China under Grant No. 61501428, No. U1406404, No.61331015 and No. 11672183.

References

[1] Berthold KP Horn and Brian G Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1, pp. 185–203, 1981.

[2] Gianfranco Doretto, Alessandro Chiuso, Ying Nian Wu, and Stefano Soatto, "Dynamic textures," *International Journal of Computer Vision*, vol. 51, no. 2, pp. 91–109, 2003.

[3] Gianfranco Doretto, Daniel Cremers, Paolo Favaro, and Stefano Soatto, "Dynamic texture segmentation," in *Ninth IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2003, pp. 1236–1242.

[4] Guoying Zhao and Matti Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 915–928, 2007.

[5] Rizwan Chaudhry, Gregory Hager, and René Vidal, "Dynamic template tracking and recognition," *International journal of computer vision*, vol. 105, no. 1, pp. 19–48, 2013.

[6] Vijay Mahadevan and Nuno Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 171–177, 2010.

[7] Martin Szummer and Rosalind W Picard, "Temporal texture modeling," in *International Conference on Image Processing (ICIP)*. IEEE, 1996, vol. 3, pp. 823–826.

[8] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick, "Graphcut textures: image and video synthesis using graph cuts," *ACM Transactions on Graphics (ToG)*, vol. 22, no. 3, pp. 277–286, 2003.

[9] Arno Schödl, Richard Szeliski, David H Salesin, and Irfan Essa, "Video textures," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 489–498.

[10] Johannes Ballé, Aleksandar Stojanovic, and Jens-Rainer Ohm, "Models for static and dynamic texture synthesis in image and video compression," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1353–1365, 2011.

[11] Petar M Djuric, Jayesh H Kotecha, Jianqui Zhang, Yufei Huang, Tadesse Ghirmai, Mónica F Bugallo, and Joaquin Miguez, "Particle filtering," *Signal Processing Magazine, IEEE*, vol. 20, no. 5, pp. 19–38, 2003.

[12] Payam Saisan, Gianfranco Doretto, Ying Nian Wu, and Stefano Soatto, "Dynamic texture recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2001, vol. 2, pp. II–58.

[13] Antoni B Chan and Nuno Vasconcelos, "Classifying video with kernel dynamic textures," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2007, pp. 1–6.

[14] Roberto Costantini, Luciano Sbaiz, and Sabine Süsstrunk, "Higher order svd analysis for dynamic texture synthesis," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 42–52, 2008.

[15] Byron Boots, Geoffrey J Gordon, and Sajid M Siddiqi, "A constraint generation approach to learning stable linear dynamical systems," in *Advances in Neural Information Processing Systems (NIPS)*, 2007, pp. 1329–1336.

[16] Michael Elad and Michal Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[17] Simon Hawe, Matthias Seibert, and Martin Kleinsteuber, "Separable dictionary learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2013, pp. 438–445.

[18] Xian Wei, Hao Shen, and Martin Kleinsteuber, "Trace quotient meets sparsity: A method for learning low dimensional image representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5268–5277.

[19] Xian Wei, Hao Shen, and Martin Kleinsteuber, "An adaptive dictionary learning approach for modeling dynamical textures," in *Proceedings of the 39^{th} IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2014, pp. 3567–3571.

[20] Michael J Black, "Explaining optical flow events with parameterized spatio-temporal models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 1999, vol. 1.

[21] Arunkumar Ravichandran, Rizwan Chaudhry, and Rene 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.

[22] Yimo Guo, Guoying Zhao, Ziheng Zhou, and Matti Pietikäinen, "Video texture synthesis with multi-frame LBP-TOP and diffeomorphic growth model," *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 3879–3891, 2013.

[23] Jos Stam and Eugene Fiume, "Depicting fire and other gaseous phenomena using diffusion processes," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 1995, pp. 129–136.

[24] Yizhou Wang and Song-Chun Zhu, "A generative method for textured motion: Analysis and synthesis," in *European Conference on Computer Vision*. Springer, 2002, pp. 583–598.

[25] Vincent Pegoraro and Steven G Parker, "Physically-based realistic fire rendering," in *Proceedings of the Second Eurographics conference on Natural Phenomena*. Eurographics Association, 2006, pp. 51–59.

[26] Andrew W Fitzgibbon, "Stochastic rigidity: Image registration for nowhere-static scenes," in *Proceedings of Eighth IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2001, vol. 1, pp. 662–669.

[27] Ziv Bar-Joseph, Ran El-Yaniv, Dani Lischinski, and Michael Werman, "Texture mixing and texture movie synthesis using statistical learning," *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, no. 2, pp. 120–135, 2001.

[28] Sloven Dubois, Renaud Péteri, and Michel Ménard, "Decomposition of dynamic textures using morphological component analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 2, pp. 188–201, 2012.

[29] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, and Cordelia Schmid, "Evaluation of local spatio-temporal features for action recognition," in *British Machine Vision Conference (BMVC)*. BMVA Press, 2009, pp. 124–1.

[30] Jin Xie and Yi Fang, "Dynamic texture recognition with video set based collaborative representation," *Image and Vision Computing*, 2016.

[31] Yong Xu, Yuhui Quan, Haibin Ling, and Hui Ji, "Dynamic texture classification using dynamic fractal analysis," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 1219–1226.

[32] Hui Ji, Xiong Yang, Haibin Ling, and Yong Xu, "Wavelet domain multifractal analysis for static and dynamic texture classification," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 286–299, 2013.

[33] Konstantinos G Derpanis and Richard P Wildes, "Dynamic texture recognition based on distributions of spacetime oriented structure," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 191–198.

[34] Renaud Péteri and Dmitry Chetverikov, "Dynamic texture recognition using normal flow and texture regularity," in *Pattern Recognition and Image Analysis*, pp. 223–230. Springer, 2005.

[35] Sloven Dubois, Renaud Péteri, and Michel Ménard, "A comparison of wavelet based spatio-temporal decomposition methods for dynamic texture recognition," in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2009, pp. 314–321.

[36] Adeel Mumtaz, Emanuele Coviello, Gert RG Lanckriet, and Antoni B Chan, "A scalable and accurate descriptor for dynamic textures using bag of system trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 4, pp. 697–712, 2015.

[37] Bernard Ghanem and Narendra Ahuja, "Maximum margin distance learning for dynamic texture recognition," in *European Conference on Computer Vision (ECCV)*, pp. 223–236. Springer, 2010.

[38] Bernard Ghanem and Narendra Ahuja, "Sparse coding of linear dynamical systems with an application to dynamic texture recognition," in *20th International Conference on Pattern Recognition (ICPR)*. IEEE, 2010, pp. 987–990.

[39] Michal Aharon, Michael Elad, and Alfred Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[40] Julien Mairal, Francis Bach, and Jean Ponce, "Task-driven dictionary learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 791–804, 2012.

[41] Meng Yang, Lei Zhang, Xiangchu Feng, and David Zhang, "Sparse representation based fisher discrimination dictionary learning for image classification," *International Journal of Computer Vision*, vol. 109, no. 3, pp. 209–232, 2014.

[42] Ignacio Ramirez, Pablo Sprechmann, and Guillermo Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 3501–3508.

[43] Zhuolin Jiang, Zhe Lin, and Larry S Davis, "Label consistent K-SVD: learning a discriminative dictionary for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2651–2664, 2013.

[44] Mehrtash Harandi, Richard Hartley, Chunhua Shen, Brian Lovell, and Conrad Sanderson, "Extrinsic methods for coding and dictionary learning on grassmann manifolds," *International Journal of Computer Vision*, vol. 114, no. 2-3, pp. 113–136, 2015.

[45] Wenbing Huang, Fuchun Sun, Lele Cao, Deli Zhao, Huaping Liu, and Mehrtash Harandi, "Sparse coding and dictionary learning with linear dynamical systems," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3938–3947.

[46] Pavan Turaga, Ashok Veeraraghavan, Anuj Srivastava, and Rama Chellappa, "Statistical computations on grassmann and stiefel manifolds for image and video-based recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2273–2286, 2011.

[47] Mehrtash Harandi, Conrad Sanderson, Chunhua Shen, and Brian C Lovell, "Dictionary learning and sparse coding on grassmann manifolds: An extrinsic solution," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 3120–3127.

[48] Yuhui Quan, Yan Huang, and Hui Ji, "Dynamic texture recognition via orthogonal tensor dictionary learning," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 73–81.

[49] Yuhui Quan, Chenglong Bao, and Hui Ji, "Equiangular kernel dictionary learning with applications to dynamic texture analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 308–316.

[50] Yoshua Bengio, Aaron Courville, and Pascal Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[51] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al., "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[52] Jianchao Yang, Zhaowen Wang, Zhe Lin, Scott Cohen, and Thomas Huang, "Coupled dictionary training for image super-resolution," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3467–3478, 2012.

[53] Tony Van Gestel, Johan AK Suykens, Paul Van Dooren, and Bart De Moor, "Identification of stable models in subspace identification by using regularization," *IEEE Transactions on Automatic Control*, vol. 46, no. 9, pp. 1416–1420, 2001.

[54] Michael Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, New York, NY, 2010.

[55] Torbjørn Eltoft, Taesu Kim, and Te-Won Lee, "On the multivariate laplace distribution," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 300–303, 2006.

[56] P-A Absil, Robert Mahony, and Rodolphe Sepulchre, *Optimization algorithms on matrix manifolds*, Princeton University Press, 2008.

[57] Dmitry Chetverikov and Sándor Fazekas, "On motion periodicity of dynamic textures.," in *British Machine Vision Conference (BMVC)*. Citeseer, 2006, pp. 167–176.

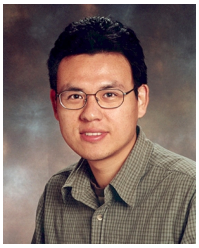
[58] Zhixiang Ren, Shenghua Gao, Deepu Rajan, Liang-Tien Chia, and Yun Huang, "Spatiotemporal saliency detection via sparse representation," in *2012 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2012, pp. 158–163.

[59] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma, "Robust face recognition via sparse representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

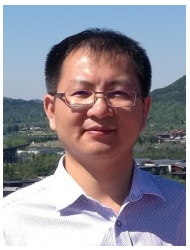
[60] Renaud Péteri, Sándor Fazekas, and Mark J Huiskes, "Dyntex: A comprehensive database of dynamic textures," *Pattern Recognition Letters*, vol. 31, no. 12, pp. 1627–1632, 2010.

Xian Wei is a Ph.D. student (since 2011) from Institute for Data Processing, Technical University of Munich, Germany. His current research is funded by Quanzhou Institute of Equipment Manufacturing, Haixi Institutes, Chinese Academy of Sciences, and Shanghai Jiao Tong University, Shanghai, China. His research interests focus on sparse coding, deep learning and geometric optimization. The applications include videos or images modeling, synthesis, visualization and recognition.

Yuanxiang Li is an associate professor at School of Aeronautics & Astronautics, Shanghai Jiao Tong University. He was a visiting professor from 2015 to 2016 at University of Michigan and a research fellow from 2002 to 2004 at National University of Singapore. He received his Ph.D. degree from Tsinghua University in 2001, China. His research interests focus on deep learning, transfer learning, sparse representation, image classification, image fusion, super-resolution reconstruction, image compression and object detection.



Hao Shen received his Bachelor degree in Mechanical Engineering and Applied Mathematics from Xi'an Jiaotong University, China, in 2000, and his Masters degrees in Computer Studies and Computer Science from the University of Wollongong, Australia, in 2002 and 2004, respectively; and received his PhD from the Australian National University, Australia, in 2008. Currently, he is a post-doctoral researcher at the Institute for Data Processing, Technische Universität München, Germany. His research interests focus on machine learning for signal processing, in particular, geometric optimization, blind signal separation, sparse representation, reinforcement learning, and deep representation learning.



Fang Chen received the Ph.D. degree in Fluid Mechanics from the Institute of Mechanics, Chinese Academy of Sciences, China in 2006. He is currently an Associate Professor at School of Aeronautics and Astronautics, Shanghai Jiao Tong University, China. His research interests include aerothermodynamics, combustion, and optical diagnostics.

Martin Kleinsteuber received his Ph.D. in Mathematics from the University of Würzburg, Germany, in 2006. After post-doc positions at National ICT Australia Ltd., the Australian National University, Canberra, Australia, and the University of Würzburg, he has been appointed assistant professor for geometric optimization and machine learning at the Department of Electrical and Computer Engineering, TU München, Germany, in 2009. He won the SIAM student paper prize in 2004 and the Robert-Sauer-Award of the Bavarian Academy of Science in 2008 for his works on Jacobi-type methods on Lie algebras. Since 2016, he is leading the Data Science Group at Mercateo AG, Munich.

Zhongfeng Wang received both B.E. and M.S. degrees from Tsinghua University, Beijing, China. He obtained the Ph.D. degree from University of Minnesota, Minneapolis in 2000. He joined Nanjing University in 2016 as a Distinguished Professor through the state's 1000-talent plan after serving Broadcom Corporation as a leading VLSI DSP architect for almost nine years. Prior to that, he was an Assistant Professor in the School of EECS at Oregon State University. Dr. Wang is a world-recognized expert on VLSI for Signal Processing Systems. He was the recipient of the IEEE Circuits and Systems Society VLSI Transactions Best Paper Award in 2007. In the current record (2007-present), he has had five papers ranked among top twenty most downloaded manuscripts in IEEE Trans. on VLSI Systems. Since 2004 Dr. Wang has served as Associate Editor for the IEEE Trans. on Circuits and Systems-I (TCAS-I), TCAS-II, and IEEE Trans. on VLSI Systems for numerous terms. In 2013, he served in the Best Paper Award selection committee for the IEEE Circuits and System Society. His current research interests are in the area of Digital Communications, Machine Learning, and efficient VLSI Implementation. He is a Fellow of IEEE.