Equiangular Kernel Dictionary Learning with Applications to Dynamic Texture Analysis

Yuhui Quan[1,2], Chenglong Bao[2], and Hui Ji[2]
[1]School of Computer Science & Engineering, South China Univ. of Tech., Guangzhou 510006, China
[2]Department of Mathematics, National University of Singapore, Singapore 117542
{csyhquan@scut.edu.cn, matbc@nus.edu.sg, matjh@nus.edu.sg}

Abstract

Most existing dictionary learning algorithms consider a linear sparse model, which often cannot effectively characterize the nonlinear properties present in many types of visual data, e.g. *dynamic texture (DT). Such nonlinear properties can be exploited by the so-called kernel sparse coding. This paper proposed an equiangular kernel dictionary learning method with optimal mutual coherence to exploit the nonlinear sparsity of high-dimensional visual data. Two main issues are addressed in the proposed method: (1) coding stability for redundant dictionary of infinite-dimensional space; and (2) computational efficiency for computing kernel matrix of training samples of high-dimensional data. The proposed kernel sparse coding method is applied to dynamic texture analysis with both local DT pattern extraction and global DT pattern characterization. The experimental results showed its performance gain over existing methods.*

1. Introduction

In recent years, sparse dictionary learning has become one important tool in computer vision. Most existing methods for sparse dictionary learning consider a sparse linear model, which assumes that most local or global patterns of data, can be represented by the linear combinations of a small number of atoms from a dictionary. In other words, the underlying assumption of these linear model based dictionary learning methods is that the data for processing is dominated by the stationary patterns generated by some linear process. Clearly, such approaches will be less efficient when processing the data whose main structures are driven by nonlinear stochastic systems.

Indeed, there are many types of visual data, especially those high-dimensional ones, showing strong nonlinear behaviors in terms of visual features. One such representative data is dynamic texture (DT). DTs are referred to as the sequences of moving textures with certain stationary temporal changes in pixel intensities. The spatio-temporal behaviors of DTs are nonlinear in general. For instance, various distinguishable shapes may be observed from flickering fires with changes in wind, which implies multiple modalities of spatio-temporal appearance. Likewise, turbulent water exhibits chaotic behaviors with non-smooth motion, where pixel intensities do not change smoothly. Moreover, camera motion is likely to further aggregate the nonlinearities of correlations among the frames of a DT sequence.

In order to exploit the nonlinear properties existing in high-dimensional visual data, the so-called *kernel sparse coding* has been proposed in the literature which considers a nonlinear sparse model; see *e.g*. [12, 12, 44, 37, 11, 18, 25].The basic idea of kernel sparse coding is linearizing the nonlinear patterns existing in data in some implicit space and then studying the resultant linear structures by sparse coding under an implicit kernel dictionary.

1.1. Motivations

By considering a linear sparse model in implicit infinite-dimensional feature space, this paper aims at developing a nonlinear sparse model for high-dimensional visual data and investigating its applications in DT analysis and recognition. The motivations of applying kernel sparse coding to processing high-dimensional visual data are three-fold. Firstly, sparse coding is able to automatically discover the multiple modalities of patterns and distinguish the mixture of linear subspaces [1]. Secondly, sparse coding with dictionary learning adapts to the data and thus can better encode the stationary behaviors of data than the handcrafted features [39]. Thirdly, the nonlinearity of local structures could be partially linearized in a proper feature space induced by certain kernels, which improves both the accuracy and discriminability of sparse coding; see *e.g*. Fig. 2.

A direct call of existing kernel sparse coding approaches is not suitable for the tasks of processing high-dimensional visual data, such as DT analysis and recognition. Most existing kernel sparse coding methods do not consider the is-

1063-6919/16 $31.00 © 2016 IEEE
DOI 10.1109/CVPR.2016.40

sue of coding stability, *i.e.*, the optimal sparse code is not unique and stable when using a general redundant dictionary. The performance hit caused by such coding stability has been observed in various sparse coding based recognition tasks, including both kernel sparse coding (*e.g.* [31, 13]) and regular sparse coding *e.g.* [29, 3]). Such ambiguities and instabilities become worse when the dimensionality of data increases. Considering the fact that the dimensionality is very high or even infinite in the implicit space induced by kernel, how to stabilize the code in the implicit space becomes an important problem.

There have been extensive studies on the design of dictionaries to ensure stable and optimal sparse coding in the context of *compressed sensing*. One important property often considered in designing dictionaries is the so-called *mutual coherence* of dictionary, which is defined by the maximal absolute value of correlations of dictionary atoms. It is shown that the sparse code is unique and can be stably computed as long as the mutual coherence of the dictionary is sufficiently small; see *e.g.* [36, 33]. Learning an incoherent dictionary has seen its applications in various vision tasks with good performance (*e.g.* [29, 24, 3, 38]). However, how to learn an incoherent dictionary in the implicit space induced by kernel is still a question, as the space can be infinite-dimensional, which prohibits explicit constraints on the mutual coherence of dictionary. This inspired us to study new computational methods for learning an optimal incoherent dictionary in the implicit space.

1.2. Main Contributions

The contribution of this paper is two-fold. Firstly, a new kernel sparse coding method is proposed, which aims at addressing two main issues, *i.e.* coding stability and computational efficiency, existing in current sparse coding methods when processing high-dimensional visual data. More specifically, we proposed to learn an equiangular kernel dictionary, *i.e.*, the inner products of all pairs of normalized dictionary atoms are the same. Based on the observation that equiangular unit-norm atoms in the original space remain equiangular in the implicit space induced by Gaussian kernels or polynomial kernels, we proposed a new optimization model for learning an equiangular dictionary in the implicit space, as well as a convergent numerical solver.

Secondly, the potential of the proposed kernel sparse coding method in practical applications is investigated in DT analysis and recognition. A new kernel sparse coding based DT descriptor is presented, in which the capability of kernel sparse coding to extract nonlinear stationary patterns is utilized in both the low-level feature extraction and the high-level feature representation. The proposed descriptor was applied to DT recognition and it shows noticeable improvement over several state-of-the-art methods on some benchmark datasets.

2. Related Works and Preliminaries

2.1. Sparse Coding and Kernel Sparse Coding

By assuming signals can be represented by a sparse linear combination of atoms from some dictionary, regular sparse coding aims at finding the coefficients of the combination as well as the dictionary. Such a technique has been successful in many vision tasks; see *e.g.* [29, 1, 22]. However, regular sparse coding assumes that signals are in linear Euclidean spaces and thus it is not effective when dealing with signals in nonlinear low-dimensional manifolds.

To generalize the concept of sparse coding to handle the signals lying in nonlinear manifolds, kernel sparse coding (*e.g.* [12, 12, 20, 25, 37]) performs sparse coding in some higher-dimensional feature space which is the map of the original Euclidean space under some kernel function. With an appropriate nonlinear mapping, a more efficient linear representation is expected for signals in nonlinear manifolds (*e.g.* [12, 44, 11]), since the nonlinear structures of signals in the lower-dimensional Euclidean space can be transformed into linear structures in the higher-dimensional Euclidean space. Moreover, by some tricks in the formulation of kernel sparse coding, it can avoid the explicit computation in high-dimensional Euclidean space, which in general is very computationally expensive and sometimes is even impossible when the implicit space is infinite-dimensional.

The dictionary for kernel sparse coding is also learned to be adaptive to input data. In [12], the dictionary is directly learned with Gaussian kernel by gradient descent over the dictionary atoms in the original space. The kernel dictionary learning is conducted on the manifold of symmetric positive definite matrices by using Stein kernel in [20] and Log-Euclidean kernel in [23]. Instead of being directly learned, in [18, 25, 37], the dictionary atoms in feature space are expressed as the linear combination of the input signals in feature space.

All existing sparse coding problems require solving nonconvex optimization problems. Therefore, both the stability and the optimality of the computed sparse code are in question when the dictionary has no additional specific properties. In recent years, learning an incoherent dictionary for better sparse coding performance has drawn a lot of attention in compressed sensing and computer vision; see e.g. [29, 24, 3, 38]). The basic idea is from the theoretical work in compressed sensing which states that sparse coding problem can be well-defined and effectively computed when the mutual coherence of dictionary is sufficiently low. In existing incoherent dictionary learning methods, the constraints on mutual coherence is explicitly expressed in terms of the inner products of all pairs of atoms, which is not computationally feasible when the dimension of atoms is infinite. In other words, the existing incoherent dictionary learning methods are not applicable to kernel sparse coding.

2.2. Dynamic Texture Analysis

The analysis and recognition on dynamic textures provide useful cues for understanding dynamic data, which has seen their applications in video registration, surveillance, facial expression recognition, motion analysis, and many others; see *e.g.* [17, 46, 26].By assuming the underlying dynamics of DT sequences is linear, many generative methods characterize the local behaviors of DTs with linear models, which include the linear dynamical system (LDS) [32, 40, 16] and its hierarchical extension [19], the autoregressive model [35, 39] and its multi-scale extension [9], etc. These methods are vulnerable to the DT sequences with camera motions like zooming and panning or with chaotic motions driven by nonlinear stochastic dynamic systems (*e.g.* flapping flags and turbulent waters).

To characterize the nonlinearities of DTs, some nonlinear generative models [5, 14] have been proposed, which is done by imposing priors on the form of possible nonlinearities. In [14], Fourier phase is used for modeling the global motion patterns of DTs. In [5], LDS is extended to a nonlinear system using kernel tricks. Such approaches are vulnerable to the DTs that do not satisfy the imposed specific priors [42]. A promising alternative to the generative approach is viewing each of DT sequences as a bag of local features. By treating DTs as 3D volume data, a promising alternative to the generative approach is the discriminative one, which directly extracts local DT features and organizes them into global features by some statistical measurements, *e.g.* histograms [46, 6] and fractal spectra [42, 43]. Most existing DT features are the spatio-temporal extension from the traditional ones in images or videos, *e.g.* spatio-temporal filters [42, 21], volume local binary patterns [46, 19], and histograms of orientations [6, 7]. Despite these attempts, it remains an open question how to effectively extract important nonlinear patterns that exist in both local low-level DT features and global high-level organizations.

Considering the fact that the above hand-crafted features do not adapt to the structures of DT data, dictionary learning based methods are proposed for better performance. In [39], LDS is learned as a dictionary from each class of DT sequences. In [30], the parameters of LDS are used as local features from which a cookbook is learned for generating feature codes. In [28], an orthogonal tensor dictionary is proposed for computational efficiency when using dictionary learning for DT recognition. All these methods use linear models for characterizing local DT structures without considering the nonlinearities. In [19], dictionary learning is applied to encoding the nonlinearities among global DT features instead of learning local features, which does not fully exploit the essential local property of DTs for recognition. In comparison with these approaches, our method considers the nonlinear stationary properties of DTs as well as the nonlinearities existing among global DT features.

3. Equiangular Kernel Dictionary Learning

In this section, we propose an equiangular kernel dictionary learning method for kernel sparse coding, which learns a dictionary with optimal mutual coherence has computational feasibility.

3.1. Problem Formulation

Given N training samples $\{Y_i\}_{i=1}^N \subset \mathcal{M} \subset \mathbb{R}^m$, where $\mathcal{M}$ is a Riemannian manifold which also can be a subspace. Denote $\Phi : \mathcal{M} \to \mathcal{H}$ to be a nonlinear mapping from $\mathcal{M}$ into a high-dimensional or infinite-dimensional dot product space $\mathcal{H}$. This mapping is associated with some kernel $k(x, y) = \langle \Phi(x), \Phi(y) \rangle = \Phi(x)^\top \Phi(y)$, where $x, y \in \mathcal{M}$. Accordingly, denote $\Phi(Y) = [\Phi(Y_1), \dots, \Phi(Y_N)]$ and define $K(X, Y) = \Phi(X)^\top \Phi(Y)$.

Kernel sparse coding aims at finding a dictionary $A \in \mathcal{H}$ such that $\Phi(Y) \approx AC$ and columns of C are sparse. Regular sparse coding cannot be directly called to solve this problem, as Φ is implicit and it maps finite vectors into an infinite-dimensional space in most cases. In this paper, we propose to learn a dictionary $A = \Phi(D)$ in feature space with $D \in \mathbb{R}^{m \times n}$. The mutual coherence of such a dictionary in feature space is then given by

$$\mu(\Phi(D)) = \max_{i \neq j} \frac{|\langle \Phi(D_i), \Phi(D_j) \rangle|}{\|\Phi(D_i)\| \|\Phi(D_j)\|}.$$

A dictionary optimized in terms of mutual coherence is closely related to the so-called *equiangular tight frame* (Grassmannian frame) [34]. It is shown in [34] that for a complete system D with for $\mathbb{R}^m$ with n unit-norm atoms, the lowest bound of mutual coherence is $\sqrt{\frac{n-m}{m(n-1)}}$ and such a bound is achieved if and only if the dictionary is an equiangular tight frame such that

$$|\langle D_i, D_j \rangle| = c_0, \quad \forall i \neq j \tag{1}$$

for some constant c_0 which equals to the mutual coherence of unit-norm atoms. Thus, in this paper, we consider learning an equiangular dictionary for obtaining a dictionary with optimal mutual coherence.

However, using the constraints of the form (1) for designing the dictionary $\Phi(D)$ is challenging. The following proposition solves such a problem by showing that the equiangular constraints on the atoms of $\Phi(D)$ can be transfered to that of D in a finite-dimensional space for certain types of kernel functions.

Proposition 3.1. *Let $\Phi : \mathcal{M} \to \mathcal{H}$ to be a mapping from $\mathcal{M} \subset \mathbb{R}^m$ into a dot product space $\mathcal{H}$ with its associated kernel k of the form $k(x, y) = \psi(\|x - y\|_2^2)$. Let $D = \{D_1, D_2, \dots, D_n\} \subset \mathcal{M}$ be an equiangular dictionary such that $\|D_i\|_2 = \|D_j\|_2$ and $\langle D_i, D_j \rangle = \mu_0$, for* *all i, j and some constant μ_0. Then, $\Phi(D)$ also forms an equiangular dictionary in $\mathcal{H}$ such that*

$$\|\Phi(D_i)\|_2 = c_0, \forall i \text{ and } \langle \Phi(D_i), \Phi(D_j)\rangle = \eta, \forall i \neq j,$$

for some constants c_0 and η.

Proof. See the proof in supplementary materials. □

It can be seen that the kernel matrix $K = \Phi(D)^\top \Phi(D)$ used in computation has the structure

$$K_{i,j} = k(D_i, D_j) = \langle \Phi(D_i), \Phi(D_j)\rangle = \eta, \quad \forall i \neq j.$$

Based on Prop. 3.1, we formulate the problem of equiangular kernel sparse coding as follows,

$$\begin{aligned} &\min_{D,C} \tfrac{1}{2}\|\Phi(Y) - \Phi(D)C\|_F^2 \\ &\text{s.t. } \|C_z\|_0 \leq T, \ \forall z, \\ &\qquad \|D_i\|_2 = 1, \ \langle D_i, D_j\rangle = \mu, \ \forall i \neq j, \end{aligned} \tag{2}$$

where T is the predefined sparsity level, μ is the predefined incoherence level, and Φ is the nonlinear mapping with its associated kernel function k satisfying

$$k(x, y) = \psi(\|x - y\|_2^2). \tag{3}$$

Remark 3.2. *Both Gaussian function and polynomial function can be used as ψ in (3). When using Gaussian function, it becomes the Gaussian kernel which is often seen in applications.*

To further simplify the computation when dealing with volume data like DTs, we set $\mu = 0$ in (2) in the application of DT recognition. In other words, we consider an orthogonal dictionary D. It is noted that $\Psi(D)$ is not an orthogonal dictionary in feature space. More concretely, the model (2) for DT analysis is in the following form:

$$\begin{aligned} &\min_{D,C} \tfrac{1}{2}\|\Phi(Y) - \Phi(D)C\|_F^2 \\ &\text{s.t. } \|C_z\|_0 \leq T, \ \forall z, D^\top D = I, \ \|C\|_\infty \leq M. \end{aligned} \tag{4}$$

The last constraint in (4) is mainly for the stability in numerical solvers, where M can be set sufficiently large so as to keep the accuracy of sparse approximation.

3.2. Numerical Algorithm

To avoid direct computation in the implicit space induced by kernel, the model (4) is reformulated as follows,

$$\begin{aligned} \|\Phi(Y) - \Phi(D)C\|_F^2 &= \mathrm{Tr}(C^\top K(D,D)C) \\ &- 2\mathrm{Tr}(K(D,Y)^\top C) + \mathrm{Tr}(K(Y,Y)), \end{aligned} \tag{5}$$

which is based on the fact that $\|X\|_F^2 = \mathrm{Tr}(X^\top X)$ and the kernel trick $K(X,Y) = \Phi(X)^\top \Phi(Y)$. Substituting (5) into (4), we rewrite (4) as

$$\min_{D\in\mathcal{D}, C\in\mathcal{C}} \tfrac{1}{2}\mathrm{Tr}(C^\top QC - 2K(D,Y)^\top C), \tag{6}$$

where $\mathcal{C} = \{C : \|C\|_\infty \leq M, \ \|C_z\|_0 \leq T, \ \forall z\}$, $\mathcal{D} = \{D : D^\top D = I\}$, and $Q = K(D,D)$. It is easy to verify that Q is fixed for $D \in \mathcal{D}$ given the kernel k satisfying (3). When k is the Gaussian kernel, Q is positive definite.

The problem (6) is a challenging non-smooth and non-convex problem, owing to the terms with l_0 norm, the orthogonality constraints, and the ambiguity between D and C. Based on the proximal alternating scheme [2], we present an efficient numerical solver for (6) with rigorous convergence analysis. The proposed algorithm is summarized in Alg. 1. In the next, we will detail each step of the algorithm. To streamline the presentation of our algorithm, we define

$$\begin{aligned} &H(C,D) = \tfrac{1}{2}\mathrm{Tr}(C^\top QC - 2K(Y,D)^\top C), \\ &F(C) = \delta_{\mathcal{C}}(C), \ G(D) = \delta_{\mathcal{D}}(C), \end{aligned} \tag{7}$$

where $\delta_{\mathcal{C}}(C)$ is an indicator function of $\mathcal{C}$, *i.e.* $\delta_{\mathcal{C}}(C) = 0$ if $C \in \mathcal{C}$ and $\delta_{\mathcal{C}}(C) = +\infty$ if $C \notin \mathcal{C}$, The problem (6) is equivalent to the unconstrained minimization problem:

$$\min_{D,C} \ F(C) + G(D) + H(C,D). \tag{8}$$

We alternately update C and D with the following scheme.

Algorithm 1 Equiangular kernel dictionary learning

```
INPUT: Training signals $Y$ and kernel function $k$.
OUTPUT: Learned dictionary $D$ and sparse code $C$.
Initialization: set $\{\gamma_j\}_{j=1}^{\infty} > 1$, $\lambda_{max}(Q) < a < b$
    and $0 < c < d$ such that $d > L_{max}$
for $j = 1, 2, \ldots,$ do
    Update the sparse code $C$ via (10).
    Set the step size $t^j$ via (15b).
    Update the dictionary $D$ via (13).
    Set the step size $s^j$ via (15a).
end for
```

1. Kernel sparse coding. When the dictionary D is fixed, we update the sparse code C via solving:

$$C^{j+1} \in \operatorname*{argmin}_{C} F(C) + \tfrac{s^j}{2}\|C - U^j\|_F^2, \tag{9}$$

where $U^j = C^j - \nabla_C H(C^j, D^j)/s^j$ and s^j is some positive step size. This subproblem has a closed-form solution given by the following proposition.

Proposition 3.3. *The solution of (9) is given by*

$$C^{j+1} = \mathrm{sign}(U^j) \odot \min(H_T(|U^j|), M), \tag{10}$$

where $H_T(C)$ keeps only the largest T entries in each column of C.

Proof. The problem (9) is equivalent to

$$\min_{C} \ \|C - U^j\|_F^2, \text{ s.t. } \|C_z\|_0 \leq T, \forall z \ \|C\|_\infty \leq M. \tag{11}$$

It is easy to verify that (10) is the solution of (11) . □

2. Dictionary update. When the sparse code C is fixed, we update the dictionary D by solving

$$D^{j+1} \in \underset{D}{\operatorname{argmin}}\, G(D) + \tfrac{t^j}{2}\|D - V^j\|_F^2, \qquad (12)$$

where $V^j = D^j - \nabla_D H(C^{j+1}, D^j)/t^j$ and t^j is some positive step size. This problem has a closed-form solution given by the following proposition.

Proposition 3.4. *The problem* (12) *has a closed-form solution which is given by*

$$D^{j+1} = UW^\top \qquad (13)$$

where $U\Sigma W = V^j$ is the SVD of V^j.

Proof. The problem (12) is equivalent to

$$\min_D \|D - V^j\|_F^2, \text{ s.t. } D^\top D = I. \qquad (14)$$

From [47], we conclude that (13) is a solution of (14). □

3. Setting step sizes. There are two step sizes s^j and t^j that need to be set in Alg 1. Here we give a strategy for setting these two parameters for the Gaussian kernel. For this purpose, we first show that $H(C, D)$ has Lipschitz gradient in $\mathcal{D} \times \mathcal{C}$. Given a function f, we say f is Lipschitz in Ω with modulus L, if $\|f(x) - f(y)\|_2 \leq L\|x - y\|_2,\ \forall x, y \in \Omega$.

Proposition 3.5. *Let $H(C, D)$ to be the function defined in* (7) *where $k(x, y) = \exp\left(\frac{-\|x-y\|_2^2}{2\sigma^2}\right)$ is the Gaussian kernel. Then, we have $\nabla_C H$ is Lipschitz with modulus $\lambda_{max}(Q)$, where $\lambda_{max}(Q)$ is the maximal eigenvalue of Q and $\nabla_{D_\ell} H$ is Lipschitz in $\Omega := \{d : \|d\|_2 = 1\}$ with modulus $L(C_\ell)$, where $L(C_\ell)$ is defined as*

$$\frac{1}{\sigma^2}\sum_{i=1}^{n} |C_{\ell i}| \exp\left(-\frac{(1 - \|Y_i\|_2^2)^2}{2\sigma^2}\right)\left(1 + \frac{1}{\sigma^2}(1 + \|Y_i\|_2^2)^2\right).$$

Proof. See the proof in supplementary materials. □

Given $\gamma_j > 1$, $0 < a < b$ and $0 < c < d$ such that $b > \lambda_{max}(Q)$ and $d > L_{max}$, where $\lambda_{max}(Q)$ is the maximal eigenvalue of Q and $L_{max} = \max(\{L(C_\ell) : \ell = 1, 2, \ldots, m, C_\ell \in \mathcal{C})\}$. According to Prop. 3.5, we set s^j and t^j as follows:

$$s^j = \max(\min(\gamma_j \lambda_{max}(Q), b), a), \qquad (15a)$$
$$t^j = \max(\min(\gamma_j L(C^{j+1}), d), c), \qquad (15b)$$

where $L(C^{j+1}) = \max(\{L(C_\ell^{j+1}), \ell = 1, 2, \ldots, m\})$.

Remark 3.6. *Compared with the kernel sparse coding methods [18, 25, 37], our algorithm avoids the computation of $K(Y, Y)$ which is expensive when the Y is large.*

3.3. Convergence Analysis

Based on the convergence results of proximal alternating algorithm for general non-convex and non-smooth minimization problems [4], we establish the global convergence of Alg. 1 in the following theorem.

Theorem 3.7. *The sequence, $\{(C^j, D^j)\}$, generated by Alg. 1 converges to a critical point of* (6).

Proof. The proof can be done by checking the conditions of the Thm. 1 in [4], and we sketch the proof as follows. Firstly, it is easy to verify that all functions $H(C, D)$,$F(C)$ and $G(D)$ are bounded below and lower semi continuous and $H(C, D)$ is a C^1 function. Secondly, from Prop. 3.5, $\nabla_C H(C, D)$ and $\nabla_D H(C, D)$ are Lipschitz continuous with modulus $L_1(D)$ and $L_2(C)$, respectively. Moreover, $\nabla H(C, D)$ is Lipschitz continuous on any bounded set. From (15a) and (15b), the two step sizes s^j and t^j satisfy $s^j \in [a, b], t^j \in [c, d]$, for all j where $a, b, c, d > 0$. Thirdly, the function $H(C, D)$ are analytic functions, as polynomial functions and exponential functions are analytic, and $F(C)$ and $G(D)$ are semi-algebraic functions, as ℓ_0 norm and indicator function over Stiefel manifold are semi-algebraic [2]. Hence, $H(C, D) + F(C) + G(D)$ satisfies the so-called K-L property. From the Thm. 1 in [4], we conclude the global convergence of (C^j, D^j) of Alg. 1. □

3.4. Extension to Supervised Case

The proposed kernel dictionary learning method can be easily applied to supervised sparse coding which utilizes labels of signals for further discrimination. For instance, the supervised term used in D-KSVD [45] which is defined by linear prediction error can be incorporated to (4) as follows:

$$\underset{D\in\mathcal{D}, C\in\mathcal{C}, W}{\operatorname{argmin}}\ H(C, D) + \tfrac{\beta}{2}\|L - WC\|_F^2 + \tfrac{\alpha}{2}\|W\|_F^2, \quad (16)$$

where L_i is the binary label vector of the i-th sample, W is a classifier to be learned, and the scalars α and β are two weights for controlling the contribution of each term in the model. The algorithm and convergence results for (16) are detailed in the supplementary materials. Figure 2 compares the coding results of (16) and D-KSVD on 2D spirals, which demonstrates the capability of our method to capture nonlinear structures. We also extended (4) by incorporating the label consistency term developed in LC-KSVD [22]. The recognition test on the AR face dataset shows 3.2% accuracy improvement of our method over LC-KSVD.

4. Constructing DT Descriptor via Two-Layer Kernel Sparse Coding

To address the DT analysis problem in the presence of nonlinearities, we apply our kernel sparse coding approach to extracting DT features in two layers, *i.e.* the low-level feature description layer and the high-level feature representation layer. The first layer is to learn a kernel dictionary from DT patches, which is for characterizing the nonlinear local behaviors of DTs and generating useful local DT features. Using the dictionary learned in the first layer, the sparse code of each DT sequence is calculated, and the histograms on sparse code over space and time are used to construct a global feature vector for each DT sequence. Then the second layer is to obtain better representations from the global feature vectors, which is done by using kernel sparse coding to analyze the nonlinear relationships among different DT samples. The pipeline of our two-layer scheme is illustrated in Fig. 1.

In details, given a set of DT sequences $\{g_i\}_i$ with labels $\{l_i\}_i$ for training, we sample Z DT patches of size $m \times m \times m$ from each class of DT sequences and stack all of them as a matrix $Y \in R^{m^3 \times Z}$ by vectorization. Then we apply (4) to learning a dictionary D_{L} from Y, and each training DT sequence g_i is represented by its sparse code X_i under D_{L} via calculating $X_i = \mathcal{F}(\mathcal{P} \circ g_i, D_{\mathrm{L}})$, where

$$\mathcal{F}(Y, D) := \operatorname*{argmin}_X \|\Phi(Y) - \Phi(D)X\|_F^2, \qquad (17)$$

subject to $\|X_z\|_0 \leq T$ for all z, and $\mathcal{P}$ denotes the operator that extracts patches from a DT sequences with a sliding window and stacks them as a matrix. This problem can be efficiently solved by proximal methods. Due to space limit, we list the algorithm for solving (17) in the supplementary materials. Working on X_i, we calculate the histogram over the whole sequence and three mean histograms along different axes in each coding channel (*i.e.* the sparse code corresponding to the same dictionary atom). See our supplementary materials for the illustration of the above process. These histograms are concatenated as a feature vector f_i. Collecting all f_is from the training sequences as a matrix F, we further apply (4) to learning a dictionary D_{H} from F. Using the high-level dictionary D_{H}, we obtain the new representation of each f_i by calculating $\mathcal{F}(f_i, D_{\mathrm{H}})$. Such representations can be used to train a classifier or for other DT analysis problems.

When a unlabeled DT sequence arrives, we calculate its descriptor using the same process as above, *i.e.* calculate its sparse code via (17). More concretely, we compute the histogram-based feature, and obtain the high-level representation by using (17) one more time. The feature vector is input to the trained classifier for classification or for other DT analysis problems.

5. Experiments

In this section, we present the experimental evaluation on the proposed methods. In particular, we present the DT classification results on three widely-used benchmark datasets for demonstrating the effectiveness the proposed DT descriptor. Before that, we first use some synthetic data to examine the performance and computational efficiency of the proposed equiangular kernel dictionary learning approach. Throughout the experiments, the Gaussian kernel is used with careful parameter tuning.

5.1. Experiments on Synthetic Data

To demonstrate the effectiveness of Alg. 1, we generate the synthetic data as follows. First, a dictionary $D \in \mathcal{R}^{m \times n}$ is sampled from Gaussian random matrices. Then three sparse matrices C_1,C_2,$C_3 \in \mathcal{R}^{n \times N}$ are generated such that each of them has T rows of nonzeros with random values drawn from the normal distribution, and t rows out of the T rows share the same row indices among these three matrices while the indices of the remaining $T - t$ rows are totally different across different matrices. Finally we generate the signal matrix $Y = DC$ where $C = [C_1, C_2, C_3]$. As a result, we obtain three classes of signals. Each class of signals lies at a T-dimensional subspace, and these three subspaces have t-dimensional overlap.

The computational efficiency of Alg. 1 is tested by setting $T = 10$, $t = 5$, $n = 100$ and varying $N = 3 \times 500$: $3{\times}500$: $3{\times}4000$. The test was conducted in MATLAB on a PC with an Intel i5 CPU and 32G memory. For comparison, we implemented and tested the kernel KSVD method [37] in the same setting, *i.e.* with the same iteration number set to 20 and the same dictionary size set to n. The results are shown in Fig. 3. Obviously, our method is much faster and more scalable w.r.t. the amount of data. The reason is kernel KSVD requires the computation of $K(Y, Y)$, whose cost is much more expensive than that in computing $K(D, Y)$ in iterations when the amount of data is much larger than the dimension of data. To demonstrate the effectiveness of Alg. 1 in revealing the subspaces of data, we show the coding results with $N = 1500$ in Fig. 3. Furthermore, we show two interesting demos in Fig. 2 using synthetic data on nonlinear manifolds, which demonstrates the capability of Alg. 1 to linearize the nonlinear structures of data, .

5.2. Experiments on Real Datasets

There are mainly three benchmark datasets for evaluating DT analysis methods: the UCLA-DT dataset [8], the DynTex dataset [27] and the DynTex++ dataset [15]. Due to the expensive cost in gathering DT data, the first two datasets have been rearranged in previous studies to obtain multiple datasets with different setting for evaluation.[1] Throughout all these three datasets, we converted all frames to gray-scale images for removing the benefits from color. A support vector machine (SVM) with the RBF kernel is trained for classification, and the parameters of SVM are determined by cross-validation.

[1] Indeed, DynTex++ originates from DynTex. However, regarding its wide use, we consider it a separate dataset in this paper.

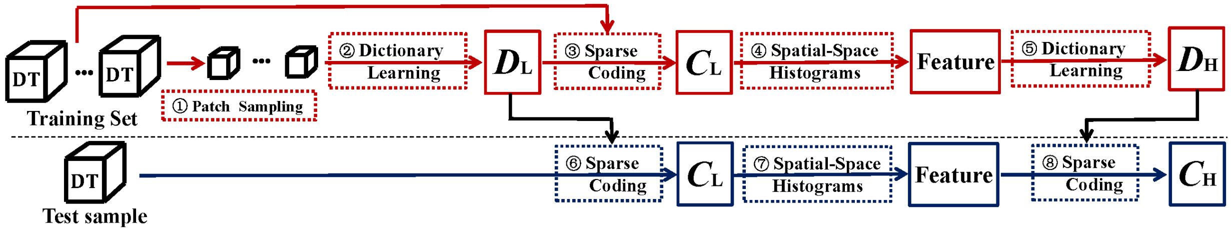
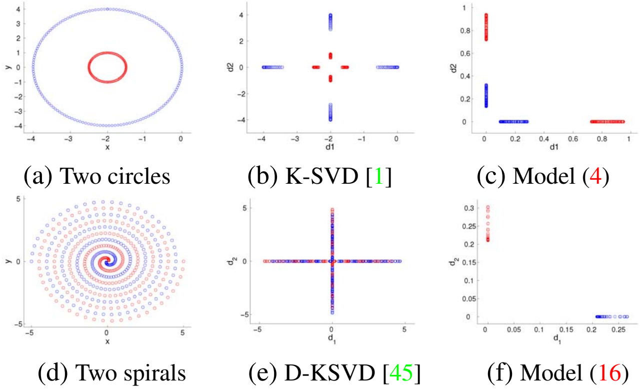


Figure 1. Pipeline of the proposed DT descriptor.



(a) Two circles (b) K-SVD [1] (c) Model (4)

(d) Two spirals (e) D-KSVD [45] (f) Model (16)

Figure 2. Proposed kernel sparse coding methods versus general sparse coding methods on two synthetic data. Note that general sparse coding methods cannot learn informative dictionary atoms in (a) and (b), as the distributions of red points and blue points along different directions are the same to each other.

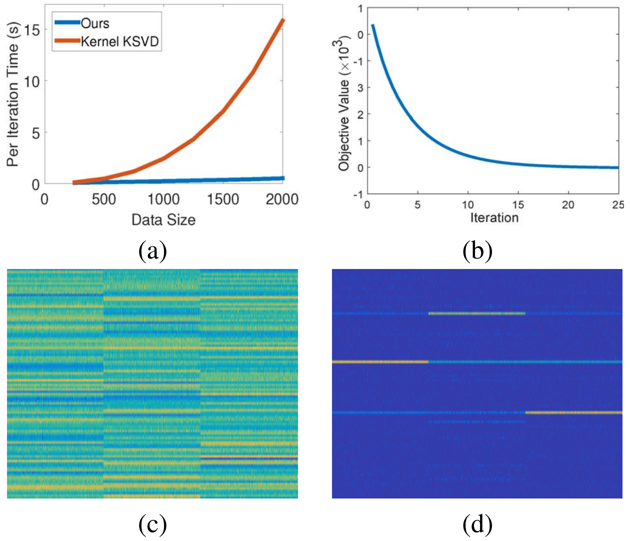


Figure 3. Some results. (a) Time costs of Alg. 1 and kernel KSVD; (b) Objective function value decay of Alg. 1 when applying Alg. 1 to (c); (c) Synthetic data; (d) Coding results from (c) by Alg. 1.

5.2.1 The UCLA-DT Dataset

The UCLA-DT dataset originally contains 200 DT sequences from 50 categories, and each category contains four video sequences captured from different viewpoints. All the videos sequences are of the size 160 × 110 × 75. Figure 4 shows some sample sequences from the dataset. There are several rearrangements for this dataset in the literature, but the performance of recent methods on most of the rearrangements has been saturated.[2] Therefore, we only selected two most challenging rearrangements for evaluation, including the 'Cat-9' protocol [15] and the 'SIR' protocol [6]:

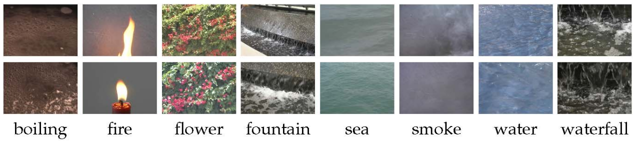


Figure 4. Snapshots of DT sequences from the UCLA-DT dataset.

- **Cat-9 (9 Categories)** [15]: The original sequences are combined from different viewpoints to form 9 categories, with the number of samples per category varying from 4 to 108. Half of the samples per category are used for training and the rest are used for test. This protocol can evaluate the robustness to viewpoint changes.
- **SIR (Shift-Invariant Recognition)** [6]: Each original video sequence is cut into non-overlapping left and right halves with careful panning, where one half is used for training and the other half for test. This protocol is mainly to evaluate the shift-invariance of descriptors.

5.2.2 The DynTex Dataset

The DynTex dataset contains a large number of DT sequences of size 720 × 576 × 250. See Figure 5 for some samples from the dataset. There are three breakdowns of the dataset which are challenging and used in previous study, including 'Alpha' [27], 'Beta' [27], and 'Gama' [27]. These breakdowns share the same protocol where five samples per category are used for training and the rest are used for test, and the differences between them are as follows:

- **Alpha [27]:** 60 sequences divided into three categories (sea, grass and trees), with 20 samples in each category.
- **Beta [27]:** 162 sequences from 10 categories, with the number of samples per class varying from 7 to 20.

[2] The saturated performance is over 90% and even has reached 100%.

Figure 5. Snapshots of DT sequences from the DynTex dataset.

- **Gamma [27]:** 275 sequences from 10 categories, with the number of samples per class varying from 7 to 38.

5.2.3 The DynTex++ Dataset

The DynTex++ dataset is a large DT dataset with 36 categories. Each category contains 100 DT sequences. See Fig. 6 for some samples of the dataset. One half samples per class are used for training and the rest are for test.

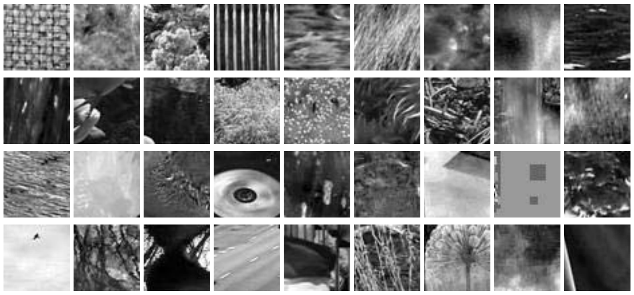


Figure 6. Snapshots of DT sequences from the DynTex++ dataset.

5.3. Implementation Details and Results

Throughout the experiments, we sampled 10000 patches from each class to stack the training set for the first-layer dictionary learning. The patch size is set according to the size as well as the resolution of training sequences, ranging from 4 to 7, and the sparsity degree is set to 5. The dictionary is initialized by a set of wavelet tight frame filters. Using the learned dictionary, we selected the sparse code which corresponds to the 25 most discriminative dictionary atoms for computing the histograms. Each histogram is 20-dimensional. In the second-layer dictionary learning, the sparsity degree is set to 7 and the dictionary is randomly initialized. In classifier training, we set the penalty coefficient of SVM to a multiple of the number of categories when the size of training set is insufficiently large for reliable cross-validation.

Results on UCLA-DT: Our method is compared to MMDL [15], DFS [42] and its variant DFS+ [43], HEM [26], OTF [41], and WMFS [21]. The results are summarized in Tab. 1, in which our method performs the best among the compared methods and shows noticeable improvement over the latest DT descriptors. It is worth mentioning that the overall performance of our method dropped by 2.8%-3.6 when we discarded the higher-level component and trained the classifiers directly by the first-layer output.

Table 1. Classification accuracies (%) on the UCLA-DT dataset.

Protocol	MMDL	DFS	DFS+	OTF	WMFS	HEM	Ours
Cat-9	95.6	97.5	97.5	97.2	97.1	97.3	98.6
SIR	-	73.8	74.2	67.4	61.2	58.0	75.8

Results on DynTex and DynTex++: Besides the aforementioned DFS, OTF, and WMFS methods, we compare our method with the LBP-TOP [46], KGDL [19], 2D+T [10]. Note that KGDL is a Grassman kernel dictionary learning method which is closely-related to our work. The classification results are summarized in Tab. 2. Again, our method outperforms the latest ones. The effect of not using kernel was simply evaluated in our framework by setting the mapping function Φ as $\Phi(x) = x$. Overall, around 2.2%-4.6% performance drop was observed on the tested datasets. We also tested the supervised extension (16) by using it for the high-level representation, and there is essentially no performance improvement. The possible reason could be that our original framework is sufficient for the complexity of the tested data.

Table 2. Classification accuracies (%) on the DynTex and and DynTex++ datasets.

Protocol	DFS	OTF	LBP-TOP	DFS+	2D+T	KGDL	Ours
Alpha	84.9	82.8	83.3	85.2	85.0	86.2	88.8
Beta	76.5	75.4	73.4	76.9	67.0	77.0	77.4
Gamma	74.5	73.5	72.0	74.8	63.0	75.1	75.6
DynTex++	89.9	89.8	89.2	91.7	-	92.8	93.4

6. Conclusion

Kernel sparse coding has been an effective tool for exploiting the nonlinear structures and patterns of data. In this paper, we proposed a new mathematical framework for learning an equiangular dictionary in the implicit space associated with some kernel, and then developed an efficient numerical solver with guaranteed convergence property. The learned equiangular dictionary has low mutual coherence to ensure the accuracy and stability of sparse coding. We also investigated the application of kernel sparse coding in DT analysis. A new DT descriptor is constructed via a two-layer kernel sparse coding based framework, where the proposed kernel sparse coding method is used for both local DT pattern extraction and global DT feature representation. The experiments on several DT datasets showed that the proposed equiangular dictionary learning method can effectively characterize DT sequences with better performance over the state-of-the-art methods. In future, we would like to investigate the potential applications of equiangular kernel dictionary learning in other visual recognition tasks.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.*, 54(11):4311–4322, 2006. 1, 2, 7

[2] C. Bao, H. Ji, Y. Quan, and Z. Shen. L0 norm based dictionary learning by proximal methods with global convergence. In *CVPR*, pages 3858–3865. IEEE, 2014. 4, 5

[3] C. Bao, Y. Quan, and H. Ji. A convergent incoherent dictionary learning algorithm for sparse coding. In *ECCV*, pages 302–316. Springer, 2014. 2

[4] J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Math. Program.*, 146(1-2):459–494, 2014. 5

[5] A. B. Chan and N. Vasconcelos. Classifying video with kernel dynamic textures. In *ICPR*, pages 1–6, 2007. 3

[6] K. G. Derpanis and R. P. Wildes. Dynamic texture recognition based on distributions of spacetime oriented structure. In *CVPR*, pages 191–198. IEEE, 2010. 3, 7

[7] K. G. Derpanis and R. P. Wildes. Spacetime texture representation and recognition based on a spatiotemporal orientation analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(6):1193–1205, 2012. 3

[8] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *Int. J. Comput. Vision*, 51(2):91–109, 2003. 6

[9] G. Doretto, E. Jones, and S. Soatto. Spatially homogeneous dynamic textures. In *ECCV*, pages 591–602. Springer, 2004. 3

[10] S. Dubois, R. Péteri, and M. Ménard. Characterization and recognition of dynamic textures based on the 2d+ t curvelet transform. *Signal, Image and Video Processing*, pages 1–12, 2013. 8

[11] S. Gao, I. W. Tsang, and L.-T. Chia. Sparse representation with kernels. *IEEE Trans. Image Process.*, 22(2):423–434, 2013. 1, 2

[12] S. Gao, I. W.-H. Tsang, and L.-T. Chia. Kernel sparse representation for image classification and face recognition. In *ECCV*, pages 1–14. Springer, 2010. 1, 2

[13] W. Gao, J. Chen, C. Richard, and J. Huang. Online dictionary learning for kernel LMS. *IEEE Trans. Signal Process.*, 62(11):2765–2777, 2014. 2

[14] B. Ghanem and N. Ahuja. Phase based modelling of dynamic textures. In *ICCV*, pages 1–8. IEEE, 2007. 3

[15] B. Ghanem and N. Ahuja. Maximum margin distance learning for dynamic texture recognition. In *ECCV*, pages 223–236. Springer, 2010. 6, 7, 8

[16] B. Ghanem and N. Ahuja. Sparse coding of linear dynamical systems with an application to dynamic texture recognition. In *ICPR*, pages 987–990. IEEE, 2010. 3

[17] B. S. Ghanem. *Dynamic textures: Models and applications*. PhD thesis, University of Illinois at Urbana-Champaign, 2010. 3

[18] M. Harandi and M. Salzmann. Riemannian coding and dictionary learning: Kernels to the rescue. In *CVPR*, volume 31, pages 210–227, 2015. 1, 2, 5

[19] M. Harandi, C. Sanderson, C. Shen, and B. C. Lovell. Dictionary learning and sparse coding on grassmann manifolds: An extrinsic solution. In *ICCV*, pages 3120–3127. IEEE, 2013. 3, 8

[20] M. T. Harandi, C. Sanderson, R. Hartley, and B. C. Lovell. Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach. In *ECCV*, pages 216–229. Springer, 2012. 2

[21] H. Ji, X. Yang, H. Ling, and Y. Xu. Wavelet domain multifractal analysis for static and dynamic texture classification. *IEEE Trans. Image Process.*, 22(1):286–299, 2013. 3, 8

[22] Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In *CVPR*, pages 1697–1704. IEEE, 2011. 2, 5

[23] P. Li, Q. Wang, W. Zuo, and L. Zhang. Log-euclidean kernels for sparse representation and dictionary learning. In *ICCV*, pages 1601–1608. IEEE, 2013. 2

[24] T. Lin, S. Liu, and H. Zha. Incoherent dictionary learning for sparse representation. In *ICPR*, pages 1237–1240. IEEE, 2012. 2

[25] H. Liu, J. Qin, H. Cheng, and S. Fuchun. Robust kernel dictionary learning using a whole sequence convergent algorithm. In *IJCAI*, 2015. 1, 2, 5

[26] A. Mumtaz, E. Coviello, G. R. Lanckriet, and A. B. Chan. Clustering dynamic textures with the hierarchical em algorithm for modeling video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(7):1606–1621, 2013. 3, 8

[27] R. Péteri, S. Fazekas, and M. J. Huiskes. DynTex : A Comprehensive Database of Dynamic Textures. *Pattern Recogn. Lett.*, 31:1627–1632, 2010. 6, 7, 8

[28] Y. Quan, Y. Huang, and H. Ji. Dynamic texture recognition via orthogonal tensor dictionary learning. In *ICCV*, pages 73–81. IEEE, 2015. 3

[29] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *CVPR*. IEEE, 2010. 2

[30] A. Ravichandran, R. Chaudhry, and R. Vidal. View-invariant dynamic texture recognition using a bag of dynamical systems. In *CVPR*, pages 1651–1657. IEEE, 2009. 3

[31] C. Richard, J. C. M. Bermudez, and P. Honeine. Online prediction of time series data with kernels. *IEEE Trans. Signal Process.*, 57(3):1058–1067, 2009. 2

[32] P. Saisan, G. Doretto, Y. N. Wu, and S. Soatto. Dynamic texture recognition. In *CVPR*, volume 2, pages II–58. IEEE, 2001. 3

[33] K. Schnass and P. Vandergheynst. Dictionary preconditioning for greedy algorithms. *IEEE Trans. Signal Process.*, 56(5):1994–2002, 2008. 2

[34] T. Strohmer and R. W. Heath. Grassmannian frames with applications to coding and communication. *Appl. Comput. Harmonic Anal.*, 14(3):257–275, 2003. 3

[35] M. Szummer and R. W. Picard. Temporal texture modeling. In *ICIP*, volume 3, pages 823–826. IEEE, 1996. 3

[36] J. Tropp et al. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. Inf. Theory*, 50(10):2231–2242, 2004. 2

[37] H. Van Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa. Design of non-linear kernel dictionaries for object recognition. *IEEE Trans. Image Process.*, 22(12):5123–5135, 2013. 1, 2, 5, 6

[38] J. Wang, J.-F. Cai, Y. Shi, and B. Yin. Incoherent dictionary learning for sparse representation based image denoising. In *ICIP*, pages 4582–4586. IEEE, 2014. 2

[39] X. Wei, H. Shen, and M. Kleinsteuber. An adaptive dictionary learning approach for modeling dynamical textures. In *ICASSP*, pages 3567–3571, May 2014. 1, 3

[40] F. Woolfe and A. Fitzgibbon. Shift-invariant dynamic texture recognition. In *ECCV*, pages 549–562. Springer, 2006. 3

[41] Y. Xu, S. Huang, H. Ji, and C. Fermüller. Scale-space texture description on sift-like textons. *Comput. Vis. Image. Und.*, 116(9):999–1013, 2012. 8

[42] Y. Xu, Y. Quan, H. Ling, and H. Ji. Dynamic texture classification using dynamic fractal analysis. In *ICCV*, pages 1219–1226. IEEE, 2011. 3, 8

[43] Y. Xu, Y. Quan, Z. Zhang, H. Ling, and H. Ji. Classifying dynamic textures via spatiotemporal fractal analysis. *Pattern Recogn.*, 2015. 3, 8

[44] L. Zhang, W.-D. Zhou, P.-C. Chang, J. Liu, Z. Yan, T. Wang, and F.-Z. Li. Kernel sparse representation-based classifier. *IEEE Trans. Signal Process.*, 60(4):1684–1695, 2012. 1, 2

[45] Q. Zhang and B. Li. Discriminative K-SVD for dictionary learning in face recognition. In *CVPR*, pages 2691–2698. IEEE, 2010. 5, 7

[46] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):915–928, 2007. 3, 8

[47] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *J. Comp. Graph. Stat.*, 15(2):265–286, 2006. 5