Dictionary Learning and Sparse Coding on Grassmann Manifolds: An Extrinsic Solution

Mehrtash Harandi[1,2], Conrad Sanderson[1,3], Chunhua Shen[4], and Brian C. Lovell[5]

[1]NICTA, Australia
[2]College of Engineering and Computer Science, Australian National University, Australia
[3]Queensland University of Technology, Australia
[4]School of Computer Science, University of Adelaide, Australia
[5]The University of Queensland, Australia

Abstract

Recent advances in computer vision and machine learning suggest that a wide range of problems can be addressed more appropriately by considering non-Euclidean geometry. In this paper we explore sparse dictionary learning over the space of linear subspaces, which form Riemannian structures known as Grassmann manifolds. To this end, we propose to embed Grassmann manifolds into the space of symmetric matrices by an isometric mapping, which enables us to devise a closed-form solution for updating a Grassmann dictionary, atom by atom. Furthermore, to handle non-linearity in data, we propose a kernelised version of the dictionary learning algorithm. Experiments on several classification tasks (face recognition, action recognition, dynamic texture classification) show that the proposed approach achieves considerable improvements in discrimination accuracy, in comparison to state-of-the-art methods such as kernelised Affine Hull Method and graph-embedding Grassmann discriminant analysis.

1. Introduction

Linear subspaces of $\mathbb{R}^d$ can be considered as the core of many inference algorithms in computer vision and machine learning. For example, several state-of-the-art methods for matching videos or image sets model given data by subspaces [9, 11, 24]. Auto regressive and moving average models, which are typically employed to model dynamics in spatio-temporal processing, can also be expressed as linear subspaces [24]. More applications of linear subspaces in computer vision include, but are not limited to, chromatic noise filtering [23], biometrics [20], and domain adaptation [8].

Despite their wide applications and appealing properties (*e.g.*, the set of all reflectance functions produced by Lambertian objects lies in a linear subspace), subspaces lie on a special type of Riemannian manifold, namely the Grassmann manifold, which makes their analysis very challenging. This paper tackles and provides efficient solutions to the following two fundamental problems for learning on Grassmann manifolds:

1. *Sparse coding.* Given a signal $\mathbf{X}$ and a dictionary $\mathbb{D} = \{\mathbf{D}_i\}_{i=1}^N$ with N elements (also known as atoms), where $\mathbf{X}$ and $\mathbf{D}_i$ are linear subspaces, how $\mathbf{X}$ can be approximated by a combination of a "few" atoms in $\mathbb{D}$?
2. *Dictionary learning.* Given a set of measurements $\{\mathbf{X}_i\}_{i=1}^m$, how can a dictionary $\mathbb{D} = \{\mathbf{D}_i\}_{i=1}^N$ be learned to represent $\{\mathbf{X}_i\}_{i=1}^m$ sparsely ?

Our main motivation here is to develop new methods for analysing video data and image sets. This is inspired by the success of sparse signal modelling that suggests natural signals like images (and hence video and image sets as our concern here) can be efficiently approximated by superposition of atoms of a dictionary, where the coefficients of superposition are usually sparse (*i.e.*, most coefficients are zero). We generalise the traditional sparse coding, which operates on vectors, to sparse coding on subspaces. Sparse encoding with the dictionary of subspaces can then be seamlessly used for categorising video data. Before we present our main results, we want to highlight that the proposed algorithms outperform state-of-the-art methods on various recognition tasks and in particular has achieved the *highest* reported accuracy in classifying dynamic textures.

Related work. While significant steps have been taken to develop the theory of the sparse coding and dictionary learning in Euclidean spaces, similar problems on non-

Euclidean geometry have received comparatively little attention [10, 15]. To our best knowledge, among a handful of solutions devised on Riemannian manifolds, none is specialised for the Grassmann manifolds which motivates our study.

In [10], the authors addressed sparse coding and dictionary learning for the Riemannian structure of Symmetric Positive Definite (SPD) matrices or tensors. The solution was obtained by embedding the SPD manifold into Reproducing Kernel Hilbert Space (RKHS) using a Riemannian kernel. Another approach to learning a Riemannian dictionary is by exploiting the tangent bundle of the manifold, as for example in [15] for the manifold of probability distributions. Since the sparse coding has a trivial solution in this approach, an affine constraint has to be added to the problem [15]. While having an affine constraint along with sparse coding is welcome in specific tasks (*e.g.*, clustering [2]), in general, the resulting formulation is restrictive and no longer addresses the original problem. Also, working in successive tangent spaces, though common, values only a first-order approximation to the manifold at each step. Furthermore, switching back and forth to the tangent spaces of a Grassmann manifold (as required by this formulation) can be computationally very demanding for the problems that we are interested in (*e.g.*, video analysis). This in turns makes the applicability of such school of thought limited for the Grassmann manifolds arising in vision tasks.

Contributions. In light of the above discussion, in this paper, we introduce an extrinsic method for learning a Grassmann dictionary. To this end, we propose to embed Grassmann manifolds into the space of symmetric matrices by a diffeomorphism that preserves Grassmann projection distance (a special class of distances on Grassmann manifolds). We show how sparse coding can be accomplished in the induced space and devise a closed-form solution for updating a Grassmann dictionary atom by atom. Furthermore, in order to accommodate non-linearity in data, we propose a kernelised version of our dictionary learning algorithm. Our contributions are therefore three-fold:

1. We propose an extrinsic dictionary learning algorithm for data points on Grassmann manifolds by embedding the manifolds into the space of symmetric matrices.
2. We derive a kernelised version of the dictionary learning algorithm which can address the non-linearity in data.
3. We apply the proposed Grassmannian dictionary learning methods to several computer vision tasks where the data are videos or image sets. Our proposed algorithms outperform state-of-the-art methods on a wide range of classification tasks, including face recognition from image sets, action recognition and dynamic texture classification.

2. Background

Before presenting our algorithms, we review some concepts of Riemannian geometry of Grassmann manifolds, which provide the grounding for the proposed algorithms. Details on Grassmann manifolds and related topics can be found in [1].

Geometry of Grassmann manifolds. The space of $d \times p$ $(0 < p < d)$ matrices with orthonormal columns is a Riemannian manifold known as a Stiefel manifold $\mathrm{St}(p,d)$, *i.e.*, $\mathrm{St}(p,d) \triangleq \{\mathbf{X} \in \mathbb{R}^{d \times p} : \mathbf{X}^T\mathbf{X} = \mathbf{I}_p\}$ where $\mathbf{I}_p$ denotes the identity matrix of size $p \times p$ [1]. Grassmann manifold $\mathcal{G}(p,d)$ can be defined as a quotient manifold of $\mathrm{St}(p,d)$ with the equivalence relation $\sim$ being: $\mathbf{X}_1 \sim \mathbf{X}_2$ if and only if $\mathrm{Span}\,(\mathbf{X}_1) = \mathrm{Span}\,(\mathbf{X}_2)$, where $\mathrm{Span}(\mathbf{X})$ denotes the subspace spanned by columns of $\mathbf{X} \in \mathrm{St}(p,d)$.

Definition 1 *A Grassmann manifold $\mathcal{G}(p,d)$ consists of the set of all linear p-dimensional subspaces of $\mathbb{R}^d$.*

The Riemannian inner product, or metric for two tangent vectors Δ_1 and Δ_2 at $\mathbf{X}$ is defined as $\langle \Delta_1, \Delta_2 \rangle_{\mathbf{X}} = \mathrm{Tr}\left(\Delta_1^T(\mathbf{I}_d - \frac{1}{2}\mathbf{X}\mathbf{X}^T)\Delta_2\right) = \mathrm{Tr}\left(\Delta_1^T\Delta_2\right)$.Further properties of the Riemannian structure of Grassmannian are given in [1]. This Riemannian structure induces a geodesic distance on the Grassmann, namely the length of the shortest curve between two points (p-dimensional subspaces), denoted $\delta_g(\mathbf{X}_1, \mathbf{X}_2)$. The special orthogonal group $SO(d)$ (think of this as higher-dimensional rotations) acts transitively on $\mathcal{G}(p,d)$ by mapping one p-dimensional subspace to another. The geodesic distance may be thought of as the magnitude of the smallest rotation (element of $SO(d)$) that takes one subspace to the other. If $\Theta = [\theta_1, \theta_2, \ldots, \theta_p]$ is the sequence of principal angles [1] between two subspaces $\mathbf{X}_1$ and $\mathbf{X}_2$, then

$$\delta_g\left(\mathbf{X}_1, \mathbf{X}_2\right) = \|\Theta\|_2. \tag{1}$$

Definition 2 *Let $\mathbf{X}_1$ and $\mathbf{X}_2$ be two orthonormal matrices of size $d \times p$. The principal angles $0 \leq \theta_1 \leq \theta_2 \leq \cdots \leq \theta_p \leq \pi/2$ between two subspaces $\mathrm{Span}(\mathbf{X}_1)$ and $\mathrm{Span}(\mathbf{X}_2)$, are defined recursively by*

$$\begin{aligned} \cos(\theta_i) = & \max_{\boldsymbol{u}_i \in \mathrm{Span}(\mathbf{X}_1)} \max_{\boldsymbol{v}_i \in \mathrm{Span}(\mathbf{X}_2)} \boldsymbol{u}_i^T \boldsymbol{v}_i \\ \text{s.t.:} \quad & \|\boldsymbol{u}_i\|_2 = \|\boldsymbol{v}_i\|_2 = 1 \\ & \boldsymbol{u}_i^T\boldsymbol{u}_j = 0;\ j = 1, 2, \cdots, i-1 \\ & \boldsymbol{v}_i^T\boldsymbol{v}_j = 0;\ j = 1, 2, \cdots, i-1 \end{aligned} \tag{2}$$

In other words, the first principal angle θ_1 is the smallest angle between all pairs of unit vectors in the first and the second subspaces. The rest of the principal angles are defined similarly. The cosines of principal angles are the singular values of $\mathbf{X}_1^T\mathbf{X}_2$ [1].

2.1. Problem statement

Given a finite set of observations $\mathbb{X} = \{\boldsymbol{x}_i\}_{i=1}^m$, $\boldsymbol{x} \in \mathbb{R}^d$, dictionary learning in vector spaces optimises the objective function

$$f(\mathbb{D}) \triangleq \sum\nolimits_{i=1}^{m} l_E(\boldsymbol{x}_i, \mathbb{D}) \tag{3}$$

with $\mathbb{D}_{d\times N} = [\boldsymbol{d}_1|\boldsymbol{d}_2|\cdots|\boldsymbol{d}_N]$ being a dictionary of size N with atoms $\boldsymbol{d}_i \in \mathbb{R}^d$. Here, $l_E(\boldsymbol{x}, \mathbb{D})$ is a loss function and should be small if $\mathbb{D}$ is "good" at representing the signal $\boldsymbol{x}$. Aiming for sparsity, the ℓ_1-norm regularisation is usually employed to obtain the most common form of $l_E(\boldsymbol{x}, \mathbb{D})$ in the literature[1]:

$$l_E(\boldsymbol{x}, \mathbb{D}) \triangleq \min_{\boldsymbol{y}} \left\| \boldsymbol{x} - \sum\nolimits_{j=1}^{N} [\boldsymbol{y}]_j \boldsymbol{d}_j \right\|_2^2 + \lambda \|\boldsymbol{y}\|_1. \tag{4}$$

With this choice, finding the optimum $\mathbb{D}$ in Eqn. (3) is not trivial because of non-convexity, as can be easily seen by rewriting the dictionary learning to:

$$\min_{\{\boldsymbol{y}_i\}_{i=1}^m, \mathbb{D}} \sum\nolimits_{i=1}^{m} \left\| \boldsymbol{x}_i - \sum\nolimits_{j=1}^{N} [\boldsymbol{y}_i]_j \boldsymbol{d}_j \right\|_2^2 + \lambda \sum\nolimits_{i=1}^{m} \|\boldsymbol{y}_i\|_1.$$

A common approach to solving this is to alternate between the two sets of variables, $\mathbb{D}$ and $\mathbf{Y} = [\boldsymbol{y}_1|\boldsymbol{y}_2|\cdots|\boldsymbol{y}_m]$, as proposed for example by Aharon *et al.* [5]. More specifically, minimising Eqn. (4) over sparse codes $\boldsymbol{y}$ while dictionary $\mathbb{D}$ is fixed is a convex problem. Similarly, minimising the overall problem over $\mathbb{D}$ with fixed $\{\boldsymbol{y}_i\}_{i=1}^m$ is convex as well.

Directly translating the dictionary learning problem to non-flat Grassmann manifolds results in writing Eqn. (4) as:

$$l_{\mathcal{G}}(\mathbf{X}, \mathbb{D}) \triangleq \min_{\boldsymbol{y}} \left\| \mathbf{X} \ominus \biguplus\nolimits_{j=1}^{N} [\boldsymbol{y}]_j \odot \mathbf{D}_j \right\|_{\mathcal{G}}^2 + \lambda \|\boldsymbol{y}\|_1. \tag{5}$$

Here $\mathbf{X} \in \mathbb{R}^{d\times p}$ and $\mathbf{D}_j \in \mathbb{R}^{d\times p}$ are points on the Grassmann manifold $\mathcal{G}_{p,d}$, while the operators $\ominus$ and $\biguplus$ are Grassmann replacements for subtraction and summation in vector spaces (and hence should be commutative and associative). Furthermore, $\odot$ is the replacement for scalar multiplication and $\|\cdot\|_{\mathcal{G}}$ is the geodesic distance on Grassmann manifolds.

There are several difficulties in solving Eqn. (5). Firstly, $\ominus$, $\biguplus$ and $\odot$ need to be appropriately defined. While the Euclidean space is closed under the subtraction and addition (and hence a sparse solution $\sum_{j=1}^{N} [\boldsymbol{y}]_j \boldsymbol{d}_j$ is a point in that space), Grassmann manifolds are not closed under normal matrix subtraction and addition. More importantly, fixing sparse codes $\boldsymbol{y}_i$ does not result in a convex cost function for dictionary learning, *i.e.* $\sum_{i=1}^{m} l_{\mathcal{G}}(\mathbf{X}_i, \mathbb{D})$ is not convex because of the distance function $\|\cdot\|_{\mathcal{G}}$ in Eqn. (5).

3. Grassmann Dictionary Learning (GDL)

In this work, we propose to embed Grassmann manifolds into the space of symmetric matrices via mapping $\Pi : \mathcal{G}(p, d) \to \mathrm{Sym(d)}, \Pi(\mathbf{X}) = \mathbf{X}\mathbf{X}^{\mathrm{T}}$. The embedding $\Pi(\mathbf{X})$ is diffeomorphism [14] (a one-to-one, continuous, differentiable mapping with a continuous, differentiable inverse) and has been used for example in subspace tracking [22]. It has been used in [2] for clustering and in [9] and [11] to develop discriminant analysis on Grassmann manifolds amongst the others. The induced space can be understood as a smooth, compact submanifold of $\mathrm{Sym(d)}$ of dimension $d(d-p)$ [14]. A natural metric on $\mathrm{Sym(d)}$ is induced by the Frobenius norm, $\|\mathbf{A}\|_F^2 = \mathrm{Tr}(\mathbf{A}\mathbf{A}^T)$ which we will exploit to *convexify* Eqn. (5). Here $\mathrm{Tr}(\cdot)$ is the matrix trace operator. As such, we define:

$$\delta_s(\mathbf{X}_1, \mathbf{X}_2) = \|\Pi(\mathbf{X}_1) - \Pi(\mathbf{X}_2)\|_F = \|\mathbf{X}_1\mathbf{X}_1^T - \mathbf{X}_2\mathbf{X}_2^T\|_F$$

as the metric in the induced space. Before explaining our solution, we note that δ_s is related to the Grassmann manifold in several aspects. This provides motivation and grounding for the follow-up formulation and is discussed briefly here.

Theorem 3 *The mapping $\Pi(\mathbf{X})$ forms an isometry from $(\mathcal{G}(p, d), \delta_p)$ onto the $(\mathrm{Sym(d)}, \delta_{\mathrm{s}})$ where the projection distance between two points on the Grassmann manifold $\mathcal{G}(p, d)$ is defined as $\delta_p^2(\mathbf{X}_1, \mathbf{X}_2) = \sum_{i=1}^{p} \sin^2\theta_i$.*

We refer the reader to [9] for the proof of this theorem.

Theorem 4 *The length of any given curve is the same under δ_s and δ_g up to a scale of $\sqrt{2}$.*

The proof of this theorem is in the Appendix.

Since $\Pi(\mathbf{X})$ is in the manifold of symmetric matrices, matrix subtraction and addition can be considered for $\ominus$ and $\biguplus$ in Eqn. (5). Therefore, we recast the dictionary learning task as optimising the empirical cost function $f(\mathbb{D}) \triangleq \sum_{i=1}^{m} l(\mathbf{X}_i, \mathbb{D})$ with

$$l(\mathbf{X}, \mathbb{D}) \triangleq \min_{\boldsymbol{y}} \left\| \mathbf{X}\mathbf{X}^T - \sum_{j=1}^{N} [\boldsymbol{y}]_j \mathbf{D}_j \mathbf{D}_j^T \right\|_F^2 + \lambda \|\boldsymbol{y}\|_1. \tag{6}$$

It is this projection mapping $\Pi(\mathbf{X}) = \mathbf{X}\mathbf{X}^T$ that leads to a simple and efficient learning approach to our problem.

3.1. Sparse Coding

Finding the sparse codes with fixed $\mathbb{D}$ is straightforward, as expanding the Frobenius norm term in Eqn. (6) results in a convex function in $\boldsymbol{y}$:

$$\left\| \mathbf{X}\mathbf{X}^T - \sum_{j=1}^{N} [\boldsymbol{y}]_j \mathbf{D}_j \mathbf{D}_j^T \right\|_F^2 = \mathrm{Tr}(\mathbf{X}^T\mathbf{X}\mathbf{X}^T\mathbf{X}) +$$

$$\sum_{j,r=1}^{N} [\boldsymbol{y}]_j [\boldsymbol{y}]_r \, \mathrm{Tr}(\mathbf{D}_r^T \mathbf{D}_j \mathbf{D}_j^T \mathbf{D}_r) - 2 \sum_{j=1}^{N} [\boldsymbol{y}]_j \, \mathrm{Tr}(\mathbf{D}_j^T \mathbf{X}\mathbf{X}^T \mathbf{D}_j).$$

[1] The notation $[\cdot]_i$ and $[\cdot]_{i,j}$ is used to demonstrate elements in position i and (i, j) in a vector and matrix, respectively.

Sparse codes can be obtained without explicit embedding of the manifold to $\mathrm{Sym(d)}$ using $\Pi(\mathbf{X})$. This can be seen by defining $[\mathcal{K}(\mathbf{X},\mathbb{D})]_i = \mathrm{Tr}(\mathbf{D}_i^T\mathbf{X}\mathbf{X}^T\mathbf{D}_i) = \|\mathbf{X}^T\mathbf{D}_i\|_F^2$ as an N dimensional vector storing the similarity between signal $\mathbf{X}$ and dictionary atoms in the induced space and $[\mathbb{K}(\mathbb{D})]_{i,j} = \mathrm{Tr}(\mathbf{D}_i^T\mathbf{D}_j\mathbf{D}_j^T\mathbf{D}_i) = \|\mathbf{D}_i^T\mathbf{D}_j\|_F^2$ as an $N \times N$ symmetric matrix encoding the similarities between dictionary atoms (which can be computed offline). Then, the sparse coding in Eqn. (6) can be written as:

$$l(\mathbf{X},\mathbb{D}) = \min_{\boldsymbol{y}} \boldsymbol{y}^T\mathbb{K}(\mathbb{D})\boldsymbol{y} - 2\boldsymbol{y}^T\mathcal{K}(\mathbf{X},\mathbb{D}) + \lambda\|\boldsymbol{y}\|_1, \tag{7}$$

which is a quadratic problem. Clearly the symmetric matrix $\mathbb{K}(\mathbb{D})$ is positive semidefinite. So the quadratic problem is convex.

3.2. Dictionary Update

To update dictionary atoms we break the minimisation problem into N sub-minimisation problems by independently updating each atom, in line with general practice in dictionary learning [5]. More specifically, fixing sparse codes and ignoring the terms that are irrelevant to dictionary atoms, the dictionary learning problem can be seen as finding $\min\limits_{\mathbb{D}} \sum_{r=1}^{N} \mathcal{J}(r)$, where:

$$\begin{aligned}\mathcal{J}(r) \quad = \quad & \sum\nolimits_{i=1}^{m}\sum\nolimits_{j=1,j\neq r}^{N} [\boldsymbol{y}_i]_r[\boldsymbol{y}_i]_j \,\mathrm{Tr}(\mathbf{D}_r^T\mathbf{D}_j\mathbf{D}_j^T\mathbf{D}_r) \\ & -2\sum\nolimits_{i=1}^{m} [\boldsymbol{y}_i]_r \,\mathrm{Tr}(\mathbf{D}_r^T\mathbf{X}_i\mathbf{X}_i^T\mathbf{D}_r). \end{aligned}\tag{8}$$

Imposing the orthogonality of $\mathbf{D}_r$ results in the following minimisation sub-problem for updating $\mathbf{D}_r$:

$$\mathbf{D}_r^* = \underset{\mathbf{D}_r}{\mathrm{argmin}}\ \mathcal{J}(r), \quad \text{s.t.} \quad \mathbf{D}_r^T\mathbf{D}_r = \mathbf{I}_p. \tag{9}$$

A closed-form solution for the above minimisation problem can be obtained by the method of Lagrange multipliers and forming $L(r,\zeta) = \mathcal{J}(r) + \zeta\left(\mathbf{D}_r^T\mathbf{D}_r - \mathbf{I}_p\right)$. The gradient of $L(r,\zeta)$ is:

$$\begin{aligned}\nabla_{\mathbf{D}_r} L(r,\zeta) = 2\sum\nolimits_{i=1}^{m}\sum\nolimits_{j=1,j\neq r}^{N} [\boldsymbol{y}]_r[\boldsymbol{y}]_j\mathbf{D}_j\mathbf{D}_j^T\mathbf{D}_r \\ - 4\sum\nolimits_{i=1}^{m} [\boldsymbol{y}_i]_r\mathbf{X}_i\mathbf{X}_i^T\mathbf{D}_r + 2\zeta\mathbf{D}_r. \end{aligned}\tag{10}$$

The solution of Eqn. (9) can be sought by finding the roots of (10), *i.e.*, $\nabla_{\mathbf{D}_r} L(r,\zeta) = 0$, which is an eigen-value problem. Therefore, the solution of (9) can be obtained by computing p eigenvectors of $\mathcal{S}$, where

$$\mathcal{S} = \sum_{i=1}^{m}\sum_{j=1,j\neq r}^{N} [\boldsymbol{y}_i]_r[\boldsymbol{y}_i]_j\mathbf{D}_j\mathbf{D}_j^T - 2\sum_{i=1}^{m}[\boldsymbol{y}_i]_r\mathbf{X}_i\mathbf{X}_i^T. \tag{11}$$

Note that the mapping $\Pi(\mathbf{X})$ meets the requirement of a Grassmann kernel [9, 12]. Consequently, it is possible to interpret the above solution as a kernel method. Nevertheless, we believe that the explanation through manifold of symmetric matrices provides more insight into the solution and also avoids possible confusion with the following section, where we present an explicitly kernelised version of GDL.

4. Kernelised GDL

In this section we propose the kernel extension of the GDL method (KGDL) to model complex nonlinear structures in the original data.

Assume a population of sets in the form of $\mathbb{X} = \{\mathbf{X}_i\}_{i=1}^m$, with $\mathbf{X}_i = \left\{\boldsymbol{x}_j^i\right\}_{j=1}^{m_i}$; $\boldsymbol{x}_j^i \in \mathbb{R}^d$ and a kernel function $k(\cdot,\cdot)$ is given. Therefore, a real-valued function on $\mathbb{R}^d \times \mathbb{R}^d$ with the property that a mapping $\phi : \mathbb{R}^d \to \mathcal{H}$ into a dot product Hilbert space $\mathcal{H}$ exists, such that for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$ we have $\langle \boldsymbol{x}, \boldsymbol{x}'\rangle_{\mathcal{H}} = \phi(\boldsymbol{x})^T\phi(\boldsymbol{x}') = k(\boldsymbol{x},\boldsymbol{x}')$ [21]. As before, we are interested in optimising $f(\mathbb{D}) \triangleq \sum_{i=1}^{m} l_\Psi(\mathbf{X}_i,\mathbb{D})$ where $l_\Psi(\mathbf{X},\mathbb{D})$ is the kernel version of (6) as depicted below:

$$\begin{aligned}& l_\Psi(\mathbf{X},\mathbb{D}) \triangleq \\ & \min_{\boldsymbol{y}} \left\|\Psi(\mathbf{X})\Psi(\mathbf{X})^T - \sum_{j=1}^{N}[\boldsymbol{y}]_j\Psi(\mathbf{D}_j)\Psi(\mathbf{D}_j)^T\right\|_F^2 + \lambda\|\boldsymbol{y}\|_1. \end{aligned}\tag{12}$$

Here, $\Psi(\mathbf{Z}) = [\boldsymbol{\psi}_1|\boldsymbol{\psi}_2|\cdots|\boldsymbol{\psi}_p]$ denotes a subspace of order p in the space $\mathcal{H}$ associated to samples of set $\mathbf{Z} = \{\boldsymbol{z}_i\}_{i=1}^{m_z}$. That is, $\boldsymbol{\psi}_j = \sum_{i=1}^{m_z} a_{i,j}\phi(\boldsymbol{z}_i)$ and $\Psi(\mathbf{Z}) = \mathbf{\Phi}(\mathbf{Z})\mathbf{A}_{\mathbf{Z}}$ with $[\mathbf{A}_{\mathbf{Z}}]_{i,j} = a_{i,j}$.

We note that the coefficients $a_{i,j}$ are given by the KPCA [21] method for input sets (*i.e.*, $\mathbf{X}_i$) while in the case of the dictionary atoms $\mathbf{D}_j$, they need to be determined by the KGDL algorithm. We also note that $\Psi(\mathbf{Z})^T\Psi(\mathbf{W}) = \mathbf{A}_Z^T\mathbf{K}(\mathbf{Z},\mathbf{W})\mathbf{A}_W$ with $\mathbf{K}(\mathbf{Z},\mathbf{W})$ being the kernel matrix of size $m_z \times m_w$ between sets $\mathbf{Z}$ and $\mathbf{W}$, *i.e.*, $[\mathbf{K}(\mathbf{Z},\mathbf{W})]_{i,j} = \phi(\boldsymbol{z}_i)^T\phi(\boldsymbol{w}_j) = k(\boldsymbol{z}_i,\boldsymbol{w}_j)$.

Obtaining sparse codes $\boldsymbol{y}$ is a straightforward task since $\left\|\Psi(\mathbf{X})\Psi(\mathbf{X})^T - \sum_{j=1}^{N}[\boldsymbol{y}]_j\Psi(\mathbf{D}_j)\Psi(\mathbf{D}_j)^T\right\|_F^2$ is a convex function in $\boldsymbol{y}$. Similar to the linear GDL method, dictionary is updated atom by atom (*i.e.*, atoms are assumed to be independent) by fixing sparse codes. As such, the cost function to update $\Psi(\mathbf{D}_r)$ can be defined as:

$$\begin{aligned}\mathcal{J}_\Psi(r) = & \sum_{i,j}[\boldsymbol{y}_i]_r[\boldsymbol{y}_i]_j\,\mathrm{Tr}(\Psi(\mathbf{D}_r)^T\Psi(\mathbf{D}_j)\Psi(\mathbf{D}_j)^T\Psi(\mathbf{D}_r)) \\ & -2\sum_{i=1}^{m}[\boldsymbol{y}_i]_r\,\mathrm{Tr}(\Psi(\mathbf{D}_r)^T\Psi(\mathbf{X}_i)\Psi(\mathbf{X}_i)^T\Psi(\mathbf{D}_r)), \\ = & \sum_{i=1}^{m}\sum_{j=1,j\neq r}^{N}[\boldsymbol{y}_i]_r[\boldsymbol{y}_i]_j\,\mathrm{Tr}(\mathbf{A}_{\mathbf{D}_r}^T\mathbf{B}(\mathbf{D}_r,\mathbf{D}_j)\mathbf{A}_{\mathbf{D}_r}) \\ & -2\sum_{i=1}^{m}[\boldsymbol{y}_i]_r\,\mathrm{Tr}(\mathbf{A}_{\mathbf{D}_r}^T\mathbf{B}(\mathbf{D}_r,\mathbf{X}_i)\mathbf{A}_{\mathbf{D}_r}). \end{aligned}\tag{13}$$

where $\mathbf{B}(\mathbf{X},\mathbf{Z}) = \mathbf{K}(\mathbf{X},\mathbf{Z})\mathbf{A}_{\mathbf{Z}}\mathbf{A}_{\mathbf{Z}}^T\mathbf{K}(\mathbf{Z},\mathbf{X})$. The orthogonality constraint in $\mathcal{H}$ can be written as:

$$\Psi(\mathbf{D}_r)^T\Psi(\mathbf{D}_r) = \mathbf{A}_{\mathbf{D}_r}^T\mathbf{K}(\mathbf{D}_r,\mathbf{D}_r)\mathbf{A}_{\mathbf{D}_r} = \mathbf{I}_p. \quad (14)$$

$\Psi(\mathbf{D}_r)$ is fully determined if $\mathbf{A}_{\mathbf{D}_r}$ and $\mathbf{K}(\cdot,\mathbf{D}_r)$ are known. If we assume that dictionary atoms are independent, then according to the representer theorem [21], $\Psi(\mathbf{D}_r)$ is a linear combination of all the $\mathbf{X}_i$ that have used it. Therefore, $\mathbf{K}(\cdot,\mathbf{D}_r) = \mathbf{K}(\cdot,\bigcup_i \mathbf{X}_i),\ y_{i,r} \neq 0$ which leaves us with identifying $\mathbf{A}_{\mathbf{D}_r}$ via the following minimisation problem:

$$\begin{aligned}
\min_{\mathbf{A}_{D_r}} \quad & \sum_{i=1}^{m}\sum_{j=1,j\neq r}^{N} y_{i,r}y_{i,j}\,\mathrm{Tr}(\mathbf{A}_{\mathbf{D}_r}^T\mathbf{B}(\mathbf{D}_r,\mathbf{D}_j)\mathbf{A}_{\mathbf{D}_r}) \\
& -2\sum\nolimits_{i=1}^{m} y_{i,r}\,\mathrm{Tr}(\mathbf{A}_{\mathbf{D}_r}^T\mathbf{B}(\mathbf{D}_r,\mathbf{X}_i)\mathbf{A}_{\mathbf{D}_r}) \\
\text{s.t.} \quad & \mathbf{A}_{\mathbf{D}_r}^T\mathbf{K}(\mathbf{D}_r,\mathbf{D}_r)\mathbf{A}_{\mathbf{D}_r} = \mathbf{I}_p.
\end{aligned}$$

The solution of the above problem is given by the eigenvectors of the generalised eigenvalue problem $\mathcal{S}_\Psi \boldsymbol{v} = \lambda\mathbf{K}(\mathbf{D}_r,\mathbf{D}_r)\boldsymbol{v}$, where:

$$\mathcal{S}_\Psi = \sum_{i=1}^{m}[\boldsymbol{y}_i]_r\Big(\sum_{j=1,j\neq r}^{N}[\boldsymbol{y}_i]_j\mathbf{B}(\mathbf{D}_r,\mathbf{D}_j) - 2\mathbf{B}(\mathbf{D}_r,\mathbf{X}_i)\Big). \quad (15)$$

5. Further Discussion

The solution proposed in (6) considers $\Pi(\mathbf{X})$ as a mapping and solves sparse coding extrinsically, meaning $\sum_i[\boldsymbol{y}]_i\mathbf{D}_i\mathbf{D}_i^T$ is not necessarily a point on $\mathcal{G}(p,d)$. If, however, it is required that the linear combination of elements $\sum_i[\boldsymbol{y}]_i\mathbf{D}_i\mathbf{D}_i^T$ actually be used to represent a point on the Grassmannian, it can be accomplished as follows. The **Eckart-Young** theorem [7] states that the matrix of rank p closest in Frobenius norm to a given matrix $\mathbf{X}$ is found by dropping all the singular values beyond the p-th one. This operation (along with equalization of the singular values) can easily be applied to a linear combination of matrices $\mathbf{D}_i\mathbf{D}_i^T$ to obtain a point on the Grassmann manifold. Thus, in a very concrete sense, the linear combination of elements $\mathbf{D}_i\mathbf{D}_i^T$, although not equaling any point on the Grassmann manifold, does *represent* such an element, the closest point lying on the manifold itself.

Furthermore, (6) follows the general principle of sparse coding in that the over-completeness of $\mathbb{D}$ will approximate $\mathbf{X}\mathbf{X}^T$ and $\sum_i[\boldsymbol{y}]_i\mathbf{D}_i\mathbf{D}_i^T$ could be safely expected to be closely tied to a Grassmannian point. Since $d \times d$ symmetric matrices of rank p with the extra property of being idempotent[2] are equivalent to points on $\mathcal{G}(p,d)$, an intrinsic version of (6) can be written as:

$$\min_{\boldsymbol{y}} \Big\|\mathbf{X}\mathbf{X}^T - \mathrm{Proj}\Big(\sum_{j=1}^{N}[\boldsymbol{y}]_j\mathbf{D}_j\mathbf{D}_j^T\Big)\Big\|_F^2 + \lambda\|\boldsymbol{y}\|_1, \quad (16)$$

where $\mathrm{Proj}(\cdot)$ is the operator that projects a symmetric matrix onto a Grassmann manifold (by forcing the idempotency and rank properties). Based on Eckart-Young theorem, $\mathrm{Proj}(\cdot)$ can be obtained through Singular Value Decomposition (SVD). The involvement of SVD, especially in vision applications, makes solving (16) tedious and challenging. We acknowledge that seeking efficient ways of solving (16) is interesting but beyond this paper.

6. Experiments

In this section we compare and contrast the performance of the proposed GDL and KGDL methods against several state-of-the-art methods: Discriminant analysis of Canonical Correlation (DCC) [17], kernel version of Affine Hull Method (KAHM) [3], Grassmann Discriminant Analysis (GDA) [9], and Graph-embedding Grassmann Discriminant Analysis (GGDA) [11]. We evaluate the performance on the tasks of **(i)** face recognition from image sets, **(ii)** dynamic texture classification and **(iii)** action recognition.

DCC is an iterative learning method that maximises a measure of discrimination between image sets where the distance between sets is expressed by canonical correlations. In KAHM, images are considered as points in a linear or affine feature space, while image sets are characterised by a convex geometric region (affine or convex hull) spanned by their feature points. GDA can be considered as an extension of kernel discriminant analysis over Grassmann manifolds [9]. In GDA, a transform over the Grassmann manifold is learned to simultaneously maximise a measure of inter-class distances and minimise intra-class distances. GGDA can be considered as an extension of GDA, where a local discriminant transform over Grassmann manifolds is learned. This is achieved by incorporating local similarities/dissimilarities through within-class and between-class similarity graphs.

Based on preliminary experiments, the Gaussian kernel [21] was used in KGDL for all tests. In GDL and KGDL methods, the dictionary is used to generate sparse codes for training and testing data. The sparse codes are then fed to a SVM for classification. The size of the dictionary is found empirically and the highest accuracy is reported here. In all experiments, the input data has the form of image sets. An image set $\mathbb{F} = \{\boldsymbol{f}_i\}_{i=1}^{b}; \boldsymbol{f}_i \in \mathbb{R}^d$, with $\boldsymbol{f}_i$ being the vectorised descriptor of frame i, can be modelled by a subspace (and hence as a point on a Grassmann manifold) through any orthogonalisation procedure like SVD. More specifically, let $\mathbb{F} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ be the SVD of $\mathbb{F}$. The dominant p left singular-vectors (p columns of $\mathbf{U}$ corresponding to the maximum singular values) represent an optimised subspace of order p (in the mean square sense) for $\mathbb{F}$ and can be seen as a point on manifold $\mathcal{G}p,d$. To select the optimum value of p, (order) of subspaces, the performance of a NN classifier on Grassmann manifolds will be evaluated for various val-

[2] A matrix $\mathbf{P}$ is called idempotent if $\mathbf{P}^2 = \mathbf{P}$.

Figure 1: Examples of YouTube celebrity dataset (grayscale versions of images were used in our experiments).

ues of ‘p’ and the value resulted in maximum performance will be chosen for constructing subspaces for each task.

6.1. Face Recognition

While face recognition from a single still image has been extensively studied, recognition based on a group of still images is relatively new. A popular choice for modelling image-sets is by representing them through linear subspaces [9, 11]. For the task of image-set face recognition, we used the YouTube celebrity dataset [16]. See Fig. 1 for examples. Face recognition on this dataset is very challenging, since the videos are compressed with a very high compression ratio and most of them are low-resolution.

To create an image set from a video, we used a cascaded face detector [25] to extract face regions from each video, followed by resizing regions to 96×96 and describing them via histogram of Local Binary Patterns (LBP) [19]. Then each image set (corresponding to a video) was represented by a linear subspace of order 5. Our data included 1471 image-sets which ware randomly split into 1236 training and 235 testing points. The process of random splitting was repeated ten times and the average classification accuracy is reported. The results in Table 1 show that the proposed GDL and KGDL approaches outperform the competitors. KGDL achieved the highest accuracy of 73.91, more than 3 percentage points better than GDL.

Table 1: Average recognition rate on the YouTube celebrity dataset.

Method	CRR
DCC [17]	60.21 ± 2.9
KAHM [3]	67.49 ± 3.5
GDA [9]	58.72 ± 3.0
GGDA [11]	61.06 ± 2.2
GDL	70.47 ± 1.7
KGDL	$\mathbf{73.91 \pm 1.9}$

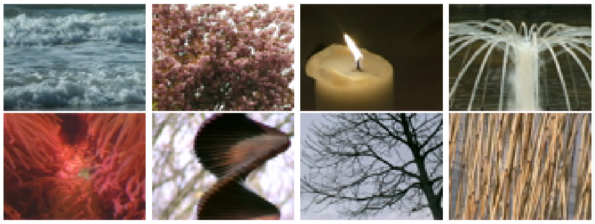

Figure 2: Example classes of DynTex++ dataset (grayscale images were used in our experiment).

6.2. Dynamic Texture Classification

Dynamic textures are videos of moving scenes that exhibit certain stationary properties in the time domain [6].Such videos are pervasive in various environments, such as sequences of rivers, clouds, fire, swarms of birds, humans in crowds. In our experiment, we used the challenging DynTex++ dataset [6], which is comprised of 36 classes, each of which contains 100 sequences with a fixed size of $50 \times 50 \times 50$ (see Fig. 2 for example classes). We split the dataset into training and testing sets by randomly assigning half of the videos of each class to the training set and using the rest as query data. The random split was repeated twenty times; average accuracy is reported.

To generate points on the Grassmann manifold, we used histogram of LBP from Three Orthogonal Planes (LBP-TOP) [27] which, takes into account the dynamics within the videos. To this end, we split each video to subvideos of length 10, with a 7 frames overlap. Each subvideo then described by a histogram of LBP-TOP features. From the subvideo descriptors, we extracted a subspace of order 5 as the video representation on Grassmann manifold.

In addition to DCC [17], KAHM [3], GDA [9] and GGDA [11], the proposed GDL and KGDL approaches were compared against Distance Learning Pegasos (DL-Pegasos) [6]. DFS can be seen as concatenation of two components: (i) a volumetric component that encodes the stochastic self-similarities of dynamic textures as 3D volumes, (ii) a multi-slice dynamic component that captures structures of dynamic textures on 2D slices along various views of the 3D volume. DL-Pegasos uses three descriptors (LBP, HOG and LDS) and learns how the descriptors can be linearly combined to best discriminate between dynamic texture classes.

The overall classification results are presented in Table 2. The proposed KGDL approach obtains the highest average recognition rate. To our best knowledge this is the highest reported result on DynTex++ dataset.

6.3. Ballet Dataset

The Ballet dataset contains 44 videos of 8 actions collected from an instructional ballet DVD [26]. The dataset

Table 2: Average recognition rate on the DynTex++ dataset.

Method	CRR
DL-PEGASOS [6]	63.7
DCC [17]	53.2
KAHM [3]	82.8
GDA [9]	81.2
GGDA [11]	84.1
GDL	90.3
KGDL	**92.8**

Figure 3: Examples from the Ballet dataset [26].

consists of 8 complex motion patterns performed by 3 subjects, and is very challenging due to the significant intra-class variations in terms of speed, spatial and temporal scale, clothing and movement.

We extracted 2400 image sets by grouping every 6 frames that exhibited the same action into one image set. We descried each image set by a subspace of order 4 with Histogram of Oriented Gradient (HOG) as frame descriptor [4]. Available samples were randomly split into training and testing set (the number of image sets in both sets was even). The process of random splitting was repeated ten times and the average classification accuracy is reported.

Table 3 shows that both GDL and KGDL algorithms have superior performance as compared to the state-of-the-art methods DCC, KAHM, GDA and KGDA. The difference between KGDL and the closest state-of-the-art competitor, *i.e.*, GGDA, is roughly ten percentage points.

Please note that in all our experiments, we randomly initialized the dictionary ten times and picked the one with minimum reconstruction error over the training set. We performed an extra experiment on ONE SPLIT of YT dataset and evaluated 10 random initialisations. The mean and std for the GDL were 72.21% and 1.6 respectively, while the performance for the dictionary with min reconstruction error was observed to be 73.19%.

Table 3: Average recognition rate on the Ballet dataset.

Method	CRR
DCC [17]	41.95 ± 9.6
KAHM [3]	70.05 ± 0.9
GDA [9]	67.33 ± 1.1
GGDA [11]	73.54 ± 2.0
GDL	79.64 ± 1.1
KGDL	$\mathbf{83.53 \pm 0.8}$

Regarding the intrinsic approach which appreciates the geodesic distance in deriving sparse codes, we performed an extra experiment on YT, following [15]. To learn the dictionary, the intrinsic approach required 26706s on an i7-Quad core platform as compared to 955sec for our algorithm. In terms of performance, our algorithm outperformed the intrinsic method (70.47% as compared to 68.51%). While this sounds counter-intuitive, we conjecture this might be due to the affine constraint required by the intrinsic approach for generating sparse codes.

7. Main Findings and Future Directions

With the aim of learning a Grassmann dictionary, we first proposed to embed Grassmann manifolds into the space of symmetric matrices by an isometric projection. We have then shown how sparse codes can be determined in the induced space and devised a closed-form solution for updating a Grassmann dictionary, atom by atom. Finally, we proposed a kernelised version of the dictionary learning algorithm, to handle non-linearity in data.

Experiments on several classification tasks (face recognition from image sets, action recognition and dynamic texture classification) show that the proposed approaches achieve notable improvements in discrimination accuracy, in comparison to state-of-the-art methods such as affine hull method [3], Grassmann discriminant analysis (GDA) [9], and graph-embedding Grassmann discriminant analysis [11].

In this work a Grassmann dictionary is learned such that a reconstruction error is minimised. This is not necessarily the optimum solution when labelled data is available. To benefit from labelled data, it has recently been proposed to consider a discriminative penalty term along with the reconstruction error term in the optimisation process [18]. We are currently pursuing this line of research and seeking solutions for discriminative dictionary learning on Grassmann manifolds. Moreover, the Frobenius norm used in our work is a special type of matrix Bregman divergence. Studying more involved cost functions derived from Bregman divergences is an interesting avenue to explore. On a side note, Bregman divergences induced from logdet function (*e.g.*, Burg or symmetrical ones like Stein) cannot be directly used here because $\mathbf{X}\mathbf{X}^T$ is rank deficient.

Acknowledgements

NICTA is funded by the Australian Government as represented by the *Department of Broadband, Communications and the Digital Economy*, as well as the Australian Research Council through the *ICT Centre of Excellence* program. This work is funded in part through an ARC Discovery grant DP130104567. C. Shen's participation was in part supported by ARC Future Fellowship F120100969.

Appendix

Here, we prove Theorem 4 (from Section 3), *i.e.*, the length of any given curve is the same under δ_g and δ_s up to a scale of $\sqrt{2}$.

Proof. Without any assumption on differentiability, let (M, d) be a metric space. A curve in M is a continuous function $\gamma : [0, 1] \rightarrow M$ and joins the starting point $\gamma(0) = x$ to the end point $\gamma(1) = y$. The intrinsic metric $\widehat{\delta}$ is defined as the infimum of the lengths of all paths from x to y. If the intrinsic metrics induced by two metrics d_1 and d_2 are identical to scale ξ, then the length of any given curve is the same under both metrics up to ξ [13].

Theorem 5 *If $d_1(x, y)$ and $d_2(x, y)$ are two metrics defined on a metric space M such that*

$$\lim_{d_1(x,y)\rightarrow 0} \frac{d_2(x, y)}{d_1(x, y)} = 1. \tag{17}$$

uniformly (with respect to x and y), then their intrinsic metrics are identical [13].

Therefore, we need to study the behaviour of

$$\lim_{\delta_g(\mathbf{X},\mathbf{Y})\rightarrow 0} \frac{\delta_s(\mathbf{X}, \mathbf{Y})}{\delta_g(\mathbf{X}, \mathbf{Y})}$$

to prove our theorem on curve lengths. We note that $\delta_s^2(\mathbf{X}, \mathbf{Y}) = 2\sum_{i=1}^{p} \sin^2 \theta_i$. Since $\sin \theta_i \rightarrow \theta_i$ for $\theta_i \rightarrow 0$, we have

$$\lim_{\delta_g(\mathbf{X},\mathbf{Y})\rightarrow 0} \frac{\delta_s^2(\mathbf{X}, \mathbf{Y})}{\delta_g^2(\mathbf{X}, \mathbf{Y})} = \lim_{\delta_g(\mathbf{X},\mathbf{Y})\rightarrow 0} \frac{2\sum_{i=1}^{p} \sin^2 \theta_i}{\sum_{i=1}^{p} \theta_i^2} = 2\ ,$$

■

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, USA, 2008.

[2] H. E. Cetingul and R. Vidal. Intrinsic mean shift for clustering on stiefel and grassmann manifolds. In *CVPR*, pages 1896–1902, 2009.

[3] H. Cevikalp and B. Triggs. Face recognition based on image sets. In *CVPR*, pages 2567–2573, 2010.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[5] M. Elad. *Sparse and Redundant Representations - From Theory to Applications in Signal and Image Processing*. Springer, 2010.

[6] B. Ghanem and N. Ahuja. Maximum margin distance learning for dynamic texture recognition. In *Proc. Eur. Conf. Comput. Vis.*, volume 6312, pages 223–236, 2010.

[7] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[8] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Proc. Int. Conf. Comput. Vis.*, pages 999–1006, 2011.

[9] J. Hamm and D. D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proc. Int. Conf. Mach. Learn.*, pages 376–383, 2008.

[10] 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 *Proc. Eur. Conf. Comput. Vis.*, pages 216–229, 2012.

[11] M. T. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell. Graph embedding discriminant analysis on Grassmannian manifolds for improved image set matching. In *CVPR*, pages 2705–2712, 2011.

[12] M. T. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell. Kernel analysis on grassmann manifolds for action recognition. *Pattern Recognition Letters*, 34(15):1906 – 1915, 2013.

[13] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *Int. J. Comput. Vis.*, 2013.

[14] J. T. Helmke, Knut Hüper. Newtons's method on Grassmann manifolds. *Preprint: [arXiv:0709.2205]*, 2007.

[15] J. Ho, Y. Xie, and B. Vemuri. On a nonlinear generalization of sparse coding and dictionary learning. In *Proc. Int. Conf. Mach. Learn.*, pages 1480–1488, 2013.

[16] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *CVPR*, pages 1–8, 2008.

[17] T.-K. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *IEEE Trans. Pattern Anal. & Mach. Intell.*, 29(6):1005–1018, 2007.

[18] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE Trans. Pattern Anal. & Mach. Intell.*, 34(4):791–804, 2012.

[19] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. & Mach. Intell.*, 24:971–987, July 2002.

[20] C. Sanderson, M. T. Harandi, Y. Wong, and B. C. Lovell. Combined learning of salient local descriptors and distance metrics for image set face verification. In *Int. Conf. Adv. Video and Signal-Based Surveillance (AVSS)*, pages 294–299, 2012.

[21] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[22] A. Srivastava and E. Klassen. Bayesian and geometric subspace tracking. *Advances in Applied Probability*, 36(1):43–56, 2004.

[23] R. Subbarao and P. Meer. Nonlinear mean shift over Riemannian manifolds. *Int. J. Comput. Vis.*, 84(1):1–20, 2009.

[24] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *IEEE Trans. Pattern Anal. & Mach. Intell.*, 33(11):2273–2286, 2011.

[25] P. Viola and M. J. Jones. Robust real-time face detection. *Int. J. Comput. Vis.*, 57(2):137–154, 2004.

[26] Y. Wang and G. Mori. Human action recognition by semilatent topic models. *IEEE Trans. Pattern Anal. & Mach. Intell.*, 31(10):1762–1774, 2009.

[27] 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.