From Manifold to Manifold: Geometry-Aware Dimensionality Reduction for SPD Matrices

Mehrtash T. Harandi, Mathieu Salzmann, and Richard Hartley

Australian National University, Canberra, ACT 0200, Australia
NICTA[⋆], Locked Bag 8001, Canberra, ACT 2601, Australia

Abstract. Representing images and videos with Symmetric Positive Definite (SPD) matrices and considering the Riemannian geometry of the resulting space has proven beneficial for many recognition tasks. Unfortunately, computation on the Riemannian manifold of SPD matrices –especially of high-dimensional ones– comes at a high cost that limits the applicability of existing techniques. In this paper we introduce an approach that lets us handle high-dimensional SPD matrices by constructing a lower-dimensional, more discriminative SPD manifold. To this end, we model the mapping from the high-dimensional SPD manifold to the low-dimensional one with an orthonormal projection. In particular, we search for a projection that yields a low-dimensional manifold with maximum discriminative power encoded via an affinity-weighted similarity measure based on metrics on the manifold. Learning can then be expressed as an optimization problem on a Grassmann manifold. Our evaluation on several classification tasks shows that our approach leads to a significant accuracy gain over state-of-the-art methods.

Keywords: Riemannian geometry, SPD manifold, Grassmann manifold, dimensionality reduction, visual recognition.

1 Introduction

This paper introduces an approach to embedding the Riemannian structure of Symmetric Positive Definite (SPD) matrices into a lower-dimensional, more discriminative Riemannian manifold. SPD matrices are becoming increasingly pervasive in various domains. For instance, diffusion tensors naturally arise in medical imaging [16]. In computer vision, SPD matrices have been shown to provide powerful representations for images and videos via region covariances [20]. Such representations have been successfully employed to categorize textures [20,6], pedestrians [21], faces [15,6], actions and gestures [18].

SPD matrices can be thought of as an extension of positive numbers and form the interior of the positive semidefinite cone. It is possible to directly employ the Frobenius norm as a similarity measure between SPD matrices, hence analyzing
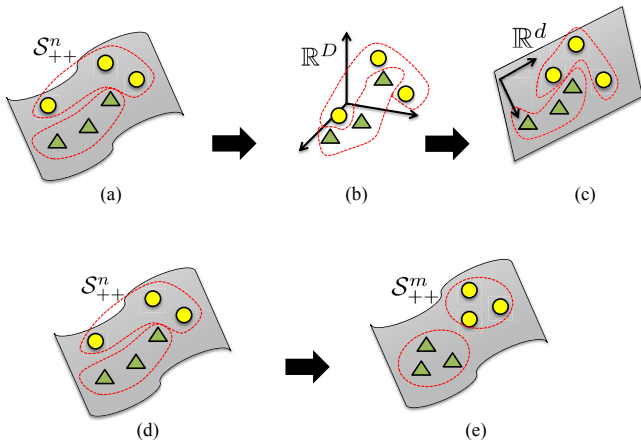
Fig. 1. Conceptual comparison of typical dimensionality reduction methods on the manifold [4,22] and our approach. **Top row (existing techniques):** The original manifold (a) is first flattened either via tangent space computation or by Hilbert space embedding. The flattened manifold (b) is then mapped to a lower-dimensional, optionally more discriminative space (c). The distortion incurred by the initial flattening may typically make this mapping more complicated. **Bottom row (our approach):** The original manifold (d) is directly transformed to a lower-dimensional, more discriminative manifold (e).

problems involving such matrices via Euclidean geometry. However, as several studies have shown, undesirable phenomena may occur when Euclidean geometry is utilized to manipulate SPD matrices [16,21,8]. One example of this is the *swelling effect* that occurs in diffusion tensor imaging (DTI), where a matrix represents the covariance of the local Brownian motion of water molecules [16]: When considering Euclidean geometry to interpolate between two diffusion tensors, the determinant of the intermediate matrices may become strictly larger than the determinants of both original matrices, which is a physically unacceptable behavior. In [16], a Riemannian structure for SPD matrices was introduced to overcome the drawbacks of the Euclidean representation. This Riemannian structure is induced by the Affine Invariant Riemmanian Metric (AIRM), and is referred to as the SPD or tensor manifold.

As shown in several studies [16,21,6,8], accounting for the geometry of SPD manifolds can have a highly beneficial impact. However, it also leads to challenges in developing effective and efficient inference methods. The main trends in analyzing SPD manifolds are to either locally flatten them via tangent space approximations [21,18], or embed them in higher-dimensional Euclidean spaces [6,2,8]. In both cases, the computational cost of the resulting methods increases dramatically with the dimension of the SPD matrices. As a consequence, very low-dimensional SPD matrices are typically employed (*e.g.*, region covariance descriptors obtained from a few low-dimensional features), with the exception

of a few studies where medium-size matrices were used [15,6]. While the matrices obtained from low-dimensional features have proven sufficient for specific problems, they are bound to be less powerful and discriminative than the high-dimensional features typically used in computer vision.

To overcome this limitation, here, we introduce an approach that lets us handle high-dimensional SPD matrices. In particular, from a high-dimensional SPD manifold, we construct a lower-dimensional, more discriminative SPD manifold. While some manifold-based dimensionality reduction techniques have been proposed [4,22], as illustrated in Fig. 1, they typically yield a Euclidean representation of the data and rely on flattening the manifold, which incurs distortions. In contrast, our approach directly works on the original manifold and exploits its geometry to learn a representation that **(i)** still benefits from useful properties of SPD manifolds, and **(ii)** can be used in conjunction with existing manifold-based recognition techniques to make them more practical and effective.

More specifically, given training SPD matrices, we search for a projection from their high-dimensional SPD manifold to a low-dimensional one such that the resulting representation maximizes an affinity-weighted similarity between pairs of matrices. In particular, we exploit the class labels to define an affinity measure, and employ either the AIRM, or the Stein divergence [19] to encode the similarity between two SPD matrices. Due to the affine invariance property of the AIRM and of the Stein divergence, any full rank projection would yield an equivalent representation. This allows us, without loss of generality, to model the projection with an orthonormal matrix, and thus express learning as an unconstrained optimization problem on a Grassmann manifold, which can be effectively optimized using a conjugate gradient method on the manifold.

We demonstrate the benefits of our approach on several tasks where the data can be represented with high-dimensional SPD matrices. In particular, our method outperforms state-of-the-art techniques on three classification tasks: image-based material categorization and face recognition, and action recognition from 3D motion capture sequences. A Matlab implementation of our algorithm is available from the first author's webpage.

2 Related Work

We now discuss in more details the three techniques that also tackle dimensionality reduction of manifold-valued data.

Principal Geodesic Analysis (PGA) was introduced in [4] as a generalization of Principal Component Analysis (PCA) to Riemannian manifolds. PGA identifies the tangent space whose corresponding subspace maximizes the variability of the data on the manifold. PGA, however, is equivalent to flattening the Riemannian manifold by taking its tangent space at the Karcher, or Fréchet, mean of the data. As such, it does not fully exploit the structure of the manifold. Furthermore, PGA, as PCA, cannot exploit the availability of class labels, and may therefore be sub-optimal for classification.

In [22], the Covariance Discriminative Learning (CDL) algorithm was proposed to embed the SPD manifold into a Euclidean space. In contrast to PGA,

CDL utilizes class labels to learn a discriminative subspace using Partial Least Squares (PLS) or Linear Discriminant Analysis (LDA). However, CDL relies on mapping the SPD manifold to the space of symmetric matrices via the principal matrix logarithm. While this embedding has some nice properties (*e.g.*, diffeomorphism), it can also be thought of as embedding the SPD manifold into its tangent space at the identity matrix. Therefore, although supervised, CDL also exploits data potentially distorted by the use of a single tangent space, as PGA.

Finally, in [5], several Nonlinear Dimensionality Reduction techniques were extended to their Riemannian counterparts. This was achieved by introducing various Riemannian geometry concepts, such as Karcher mean, tangent spaces and geodesics, in Locally Linear Embedding (LLE), Hessian LLE and Laplacian Eigenmaps. The resulting algorithms were applied to several unsupervised clustering tasks. Although these methods can, in principle, be employed for supervised classification, they are limited to the transductive setting since they do not define any parametric mapping to the low-dimensional space.

In this paper, we learn a mapping from a high-dimensional SPD manifold to a lower-dimensional one without relying on tangent space approximations of the manifold. Our approach therefore accounts for the structure of the manifold and can simultaneously exploit class label information. The resulting mapping lets us effectively handle high-dimensional SPD matrices for classification purposes. Furthermore, by mapping to another SPD manifold, our approach can serve as a pre-processing step to other Riemannian-based approaches, such as the manifold sparse coding of [6], thus making them practical to work with more realistic, high-dimensional features. Note that, while our formulation is inspired from graph embedding methods in Euclidean spaces, *e.g.*, [24], here we work with data lying on more challenging non-linear manifolds.

To the best of our knowledge, this is the first work that shows how a high-dimensional SPD manifold can be transformed into another SPD manifold with lower intrinsic dimension. Note that a related idea, but with a very different approach, was introduced in [9] to decompose high-dimensional spheres into submanifolds of decreasing dimensionality.

3   Riemannian Geometry of SPD Manifolds

In this section, we discuss some notions of geometry of SPD manifolds. Throughout this paper we will use the following notation: $\mathcal{S}_{++}^n$ is the space of real $n \times n$ SPD matrices; $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ is the identity matrix; $\mathrm{GL}(n)$ is the general linear group, *i.e.*, the group of real invertible $n \times n$ matrices.

Definition 1. *A real and symmetric matrix $\mathbf{X} \in \mathbb{R}^{n \times n}$ is said to be SPD if $\mathbf{v}^T \mathbf{X} \mathbf{v}$ is positive for any non-zero $\mathbf{v} \in \mathbb{R}^n$.*

The space of $n \times n$ SPD matrices is obviously not a vector space since multiplying an SPD matrix by a negative scalar results in a matrix which does not belong to $\mathcal{S}_{++}^n$. Instead, $\mathcal{S}_{++}^n$ forms the interior of a convex cone in the n^2-dimensional Euclidean space. The $\mathcal{S}_{++}^n$ space is mostly studied when endowed

with a Riemannian metric and thus forms a Riemannian manifold [16]. A natural way to measure closeness on a manifold is by considering the geodesic distance between two points on the manifold. Such a distance is defined as the length of the shortest curve connecting the two points. The shortest curves are known as geodesics and are analogous to straight lines in $\mathbb{R}^n$. The Affine Invariant Riemannian Metric (AIRM) is probably the most popular Riemannian structure for analyzing SPD matrices [16]. Let $\boldsymbol{P}$ be a point on $\mathcal{S}_{++}^n$. The AIRM for two tangent vectors $\boldsymbol{v}, \boldsymbol{w} \in T_{\boldsymbol{P}}\mathcal{S}_{++}^n$ is defined as

$$\langle \boldsymbol{v}, \boldsymbol{w} \rangle_{\boldsymbol{P}} := \langle \boldsymbol{P}^{-1/2}\boldsymbol{v}\boldsymbol{P}^{-1/2}, \boldsymbol{P}^{-1/2}\boldsymbol{w}\boldsymbol{P}^{-1/2} \rangle = \mathrm{Tr}\left(\boldsymbol{P}^{-1}\boldsymbol{v}\boldsymbol{P}^{-1}\boldsymbol{w}\right) . \quad (1)$$

Definition 2. *The geodesic distance* $\delta_g : \mathcal{S}_{++}^n \times \mathcal{S}_{++}^n \to [0, \infty)$ *induced by the AIRM is defined as*

$$\delta_g^2(\boldsymbol{X}, \boldsymbol{Y}) = \|\log(\boldsymbol{X}^{-1/2}\boldsymbol{Y}\boldsymbol{X}^{-1/2})\|_F^2 , \quad (2)$$

where $\log(\cdot)$ *is the matrix principal logarithm.*

More recently, Sra introduced the Stein metric on SPD manifolds [19]:

Definition 3. *The Stein metric* $\delta_S : \mathcal{S}_{++}^n \times \mathcal{S}_{++}^n \to [0, \infty)$ *is a symmetric type of Bregman divergence and is defined as*

$$\delta_S^2(\boldsymbol{X}, \boldsymbol{Y}) = \ln\det\left(\frac{\boldsymbol{X} + \boldsymbol{Y}}{2}\right) - \frac{1}{2}\ln\det(\boldsymbol{XY}) . \quad (3)$$

The Stein metric shows several similarities to the geodesic induced by the AIRM while being less expensive to compute [3]. In addition to the properties studied by Sra [19], we provide the following important theorem which relates the length of curves under the two metrics.

Theorem 1. *The length of any given curve is the same under* δ_g *and* δ_s *up to a scale of* $2\sqrt{2}$.

Proof. Given in supplementary material. □

One of the motivations for projecting a higher-dimensional SPD manifold to a lower-dimensional one is to preserve the properties of δ_g^2 and δ_S^2 [16,19]. One important such property, especially in computer vision, is affine invariance [16].

Property 1 (Affine invariance). For any $\boldsymbol{M} \in \mathrm{GL}(n)$,

$$\delta_g^2(\boldsymbol{X}, \boldsymbol{Y}) = \delta_g^2(\boldsymbol{MXM}^T, \boldsymbol{MYM}^T),$$
$$\delta_S^2(\boldsymbol{X}, \boldsymbol{Y}) = \delta_S^2(\boldsymbol{MXM}^T, \boldsymbol{MYM}^T).$$

This property postulates that the metric between two SPD matrices is unaffected by the action of the affine group. In the specific case where the SPD matrices are region covariance descriptors [20], this implies that the distance between two descriptors will remain unchanged after an affine transformation of the image features, such as a change of illumination when using RGB values. Note that, in addition to this specific implication, we will also exploit the affine invariance property for a different purpose when deriving our learning algorithm in the next section.

4   Geometry-Aware Dimensionality Reduction

We now describe our approach to learning an embedding of high-dimensional SPD matrices to a more discriminative, low-dimensional SPD manifold. More specifically, given a matrix $\boldsymbol{X} \in \mathcal{S}_{++}^n$, we seek to learn the parameters $\boldsymbol{W} \in \mathbb{R}^{n \times m}$, $m < n$, of a generic mapping $f : \mathcal{S}_{++}^n \times \mathbb{R}^{n \times m} \to \mathcal{S}_{++}^m$, which we define as

$$f(\boldsymbol{X}, \boldsymbol{W}) = \boldsymbol{W}^T \boldsymbol{X} \boldsymbol{W}. \tag{4}$$

Clearly, if $\mathcal{S}_{++}^n \ni \boldsymbol{X} \succ 0$ and $\boldsymbol{W}$ has full rank, $\mathcal{S}_{++}^m \ni \boldsymbol{W}^T \boldsymbol{X} \boldsymbol{W} \succ 0$.

Given a set of SPD matrices $\mathcal{X} = \{\boldsymbol{X}_1, \cdots, \boldsymbol{X}_p\}$, where each matrix $\boldsymbol{X}_i \in \mathcal{S}_{++}^n$, our goal is to find a transformation $\boldsymbol{W}$ such that the resulting low-dimensional SPD manifold preserves some interesting structure of the original data. Here, we encode this structure via an undirected graph defined by a real symmetric affinity matrix $\boldsymbol{A} \in \mathbb{R}^{p \times p}$. The element $\boldsymbol{A}_{ij}$ of this matrix measures some notion of affinity between matrices $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, and may be negative. We will discuss the affinity matrix in more details in Section 4.2.

Given $\boldsymbol{A}$, we search for an embedding such that the affinity between pairs of SPD matrices is reflected by a measure of similarity on the low-dimensional SPD manifold. In this paper, we propose to make use of either the AIRM or the Stein metric to encode (dis)similarities between SPD matrices. For each pair (i, j) of training samples, this lets us write a cost function of the form

$$\mathcal{J}_{ij}(\boldsymbol{W}; \boldsymbol{X}_i, \boldsymbol{X}_j) = \boldsymbol{A}_{ij} \delta^2 \left(\boldsymbol{W}^T \boldsymbol{X}_i \boldsymbol{W}, \boldsymbol{W}^T \boldsymbol{X}_j \boldsymbol{W} \right), \tag{5}$$

where δ is either δ_g or δ_S. These pairwise costs can then be grouped together in a global empirical cost function

$$L(\boldsymbol{W}) = \sum_{i,j} \mathcal{J}_{ij}(\boldsymbol{W}; \boldsymbol{X}_i, \boldsymbol{X}_j), \tag{6}$$

which we seek to minimize w.r.t. $\boldsymbol{W}$.

To avoid degeneracies and ensure that the resulting embedding forms a valid SPD manifold, *i.e.*, $\boldsymbol{W}^T \boldsymbol{X} \boldsymbol{W} \succ 0$, $\forall \boldsymbol{X} \in \mathcal{S}_{++}^n$, we need $\boldsymbol{W}$ to have full rank. Here, we enforce this requirement by imposing orthonormality constraints on $\boldsymbol{W}$, *i.e.*, $\boldsymbol{W}^T \boldsymbol{W} = \mathbf{I}_m$. Note that, with either the AIRM or the Stein divergence, this entails no loss of generality. Indeed, any full rank matrix $\tilde{\boldsymbol{W}}$ can be expressed as $\boldsymbol{M}\boldsymbol{W}$, with $\boldsymbol{W}$ an orthonormal matrix and $\boldsymbol{M} \in \mathrm{GL}(n)$. The affine invariance property of the AIRM and of the Stein metric therefore guarantees that

$$\mathcal{J}_{ij}(\tilde{\boldsymbol{W}}; \boldsymbol{X}_i, \boldsymbol{X}_j) = \mathcal{J}_{ij}(\boldsymbol{M}\boldsymbol{W}; \boldsymbol{X}_i, \boldsymbol{X}_j) = \mathcal{J}_{ij}(\boldsymbol{W}; \boldsymbol{X}_i, \boldsymbol{X}_j) .$$

Finally, learning can be expressed as the minimization problem

$$\boldsymbol{W}^* = \underset{\boldsymbol{W} \in \mathbb{R}^{n \times m}}{\arg\min} \sum_{i,j} \boldsymbol{A}_{ij} \delta^2 \left(\boldsymbol{W}^T \boldsymbol{X}_i \boldsymbol{W}, \boldsymbol{W}^T \boldsymbol{X}_j \boldsymbol{W} \right) \quad \text{s.t. } \boldsymbol{W}^T \boldsymbol{W} = \mathbf{I}_m . \tag{7}$$

In the next section, we describe an effective way of solving (7) via optimization on a (different) Riemannian manifold.

4.1   Optimization on Grassmann Manifolds

Recent advances in optimization methods formulate problems with orthogonality constraints as optimization problems on Stiefel or Grassmann manifolds [1]. More specifically, the geometrically correct setting for the minimization problem $\min L(\boldsymbol{W})$ with the orthogonality constraint $\boldsymbol{W}^T\boldsymbol{W} = \mathbf{I}_m$ is, in general, on a Stiefel manifold. However, if the cost function $L(\boldsymbol{W})$ possesses the property that for any rotation matrix $\boldsymbol{R}$ (i.e., $\boldsymbol{R} \in \mathrm{SO}(m), \boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{R}^T\boldsymbol{R} = \mathbf{I}_m$), $L(\boldsymbol{W}) = L(\boldsymbol{W}\boldsymbol{R})$, then the problem is on a Grassmann manifold.

Since both the AIRM and the Stein metric are affine invariant, we have

$$\mathcal{J}(\boldsymbol{X}_i, \boldsymbol{X}_j, \boldsymbol{W}) = \mathcal{J}(\boldsymbol{X}_i, \boldsymbol{X}_j, \boldsymbol{W}\boldsymbol{R}),$$

and thus $L(\boldsymbol{W}) = L(\boldsymbol{W}\boldsymbol{R})$, which therefore identifies (7) as an (unconstrained) optimization problem on the Grassmann manifold $\mathcal{G}(m, n)$.

In particular, here, we utilize a nonlinear Conjugate Gradient (CG) method on Grassmann manifolds to minimize (7). A brief description of the steps of this algorithm is provided in supplementary material. For a more detailed treatment, we refer the reader to [1]. As for now, we just confine ourselves to saying that nonlinear CG on Grassmann manifolds requires the $n \times m$ Jacobian matrix of $L(\boldsymbol{W})$ w.r.t. $\boldsymbol{W}$. For the Stein metric, this Jacobian matrix can be obtained by noting that

$$D_{\boldsymbol{W}} \ln \det \left(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{W}\right) = 2\boldsymbol{X}\boldsymbol{W}\left(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{W}\right)^{-1}, \tag{8}$$

which lets us identify the Jacobian of the Stein divergence as

$$D_{\boldsymbol{W}}\delta_S^2\left(\boldsymbol{W}^T\boldsymbol{X}_i\boldsymbol{W}, \boldsymbol{W}^T\boldsymbol{X}_j\boldsymbol{W}\right) = (\boldsymbol{X}_i + \boldsymbol{X}_j)\boldsymbol{W}(\boldsymbol{W}^T\frac{\boldsymbol{X}_i + \boldsymbol{X}_j}{2}\boldsymbol{W})^{-1}$$
$$- \boldsymbol{X}_i\boldsymbol{W}(\boldsymbol{W}^T\boldsymbol{X}_i\boldsymbol{W})^{-1} - \boldsymbol{X}_j\boldsymbol{W}(\boldsymbol{W}^T\boldsymbol{X}_j\boldsymbol{W})^{-1}.$$

For the AIRM, we can exploit the fact that $\mathrm{Tr}\left(\log(\boldsymbol{X})\right) = \ln \det \left(\boldsymbol{X}\right), \forall \boldsymbol{X} \in \mathcal{S}_{++}^n$. We can then derive the Jacobian by utilizing Eq. 8, which yields

$$D_{\boldsymbol{W}}\left(\delta_g^2\left(\boldsymbol{W}^T\boldsymbol{X}_i\boldsymbol{W}, \boldsymbol{W}^T\boldsymbol{X}_j\boldsymbol{W}\right)\right) = D_{\boldsymbol{W}}\left(\left\|\log\left((\boldsymbol{W}^T\boldsymbol{X}_j\boldsymbol{W})^{-1/2}\boldsymbol{W}^T\boldsymbol{X}_i\boldsymbol{W}(\boldsymbol{W}^T\boldsymbol{X}_j\boldsymbol{W})^{-1/2}\right)\right\|_F^2\right)$$
$$= 2D_{\boldsymbol{W}}\left\{\mathrm{Tr}\left(\log\left((\boldsymbol{W}^T\boldsymbol{X}_j\boldsymbol{W})^{-1/2}\boldsymbol{W}^T\boldsymbol{X}_i\boldsymbol{W}(\boldsymbol{W}^T\boldsymbol{X}_j\boldsymbol{W})^{-1/2}\right)\right)\right\}\cdot$$
$$\cdot \log\left((\boldsymbol{W}^T\boldsymbol{X}_j\boldsymbol{W})^{-1/2}\boldsymbol{W}^T\boldsymbol{X}_i\boldsymbol{W}(\boldsymbol{W}^T\boldsymbol{X}_j\boldsymbol{W})^{-1/2}\right)$$
$$= 2D_{\boldsymbol{W}}\left(\ln\det\left(\boldsymbol{W}^T\boldsymbol{X}_i\boldsymbol{W}(\boldsymbol{W}^T\boldsymbol{X}_j\boldsymbol{W})^{-1}\right)\right)\log\left((\boldsymbol{W}^T\boldsymbol{X}_j\boldsymbol{W})^{-1/2}\boldsymbol{W}^T\boldsymbol{X}_i\boldsymbol{W}(\boldsymbol{W}^T\boldsymbol{X}_j\boldsymbol{W})^{-1/2}\right)$$
$$= 4\left(\boldsymbol{X}_i\boldsymbol{W}(\boldsymbol{W}^T\boldsymbol{X}_i\boldsymbol{W})^{-1} - \boldsymbol{X}_j\boldsymbol{W}(\boldsymbol{W}^T\boldsymbol{X}_j\boldsymbol{W})^{-1}\right)\log\left((\boldsymbol{W}^T\boldsymbol{X}_j\boldsymbol{W})^{-1/2}\boldsymbol{W}^T\boldsymbol{X}_i\boldsymbol{W}(\boldsymbol{W}^T\boldsymbol{X}_j\boldsymbol{W})^{-1/2}\right).$$

The pseudo-code for our SPD manifold learning (SPD-ML) method is given in Algorithm 1, where $\nabla_{\boldsymbol{W}}L(\boldsymbol{W})$ denotes the gradient on the manifold obtained from the Jacobian $D_{\boldsymbol{W}}L(\boldsymbol{W})$, and $\tau(\boldsymbol{H}, \boldsymbol{W}_0, \boldsymbol{W}_1)$ denotes the parallel transport of tangent vector $\boldsymbol{H}$ from $\boldsymbol{W}_0$ to $\boldsymbol{W}_1$ (see supplementary material for details).

4.2   Designing the Affinity Matrix

Different criteria can be employed to build the affinity matrix $\boldsymbol{A}$. In this work, we focus on classification problems on $\mathcal{S}_{++}^n$ and therefore exploit class labels to

---

Algorithm 1. SPD Manifold Learning (SPD-ML)

Input:
A set of SPD matrices $\{\boldsymbol{X}_i\}_{i=1}^p$, $\boldsymbol{X}_i \in \mathcal{S}_{++}^n$
The corresponding labels $\{y_i\}_{i=1}^p$, $y_i \in \{1, 2, \cdots, C\}$
The dimensionality m of the induced manifold

Output:
The mapping $\boldsymbol{W} \in \mathcal{G}(m, n)$

Generate $\boldsymbol{A}$ using (9), (10) and (11)
$\boldsymbol{W}_{old} \leftarrow \boldsymbol{I}_{n \times m}$ (*i.e.*, the truncated identity matrix)
$\boldsymbol{W} \leftarrow \boldsymbol{W}_{old}$
$\boldsymbol{H}_{old} \leftarrow \boldsymbol{0}$
repeat
    $\boldsymbol{H} \leftarrow -\nabla_{\boldsymbol{W}} L(\boldsymbol{W}) + \eta\tau(\boldsymbol{H}_{old}, \boldsymbol{W}_{old}, \boldsymbol{W})$
    Line search along the geodesic $\gamma(t)$ from $\boldsymbol{W} = \gamma(0)$ in the direction $\boldsymbol{H}$ to find
    $\boldsymbol{W}^* = \underset{\boldsymbol{W}}{\mathrm{argmin}}\ L(\boldsymbol{W})$
    $\boldsymbol{H}_{old} \leftarrow \boldsymbol{H}$
    $\boldsymbol{W}_{old} \leftarrow \boldsymbol{W}$
    $\boldsymbol{W} \leftarrow \boldsymbol{W}^*$
until convergence

---

construct $\boldsymbol{A}$. Note, however, that our framework is general and also applies to unsupervised or semi-supervised settings. For example, in an unsupervised scenario, $\boldsymbol{A}$ could be built from pairwise similarities (distances) on $\mathcal{S}_{++}^n$. Solving (7) could then be understood as finding a mapping where nearby data pairs on the original manifold $\mathcal{S}_{++}^n$ remain close in the induced manifold $\mathcal{S}_{++}^m$.

Let us assume that each point $\boldsymbol{X}_i \in \mathcal{S}_{++}^n$ belongs to one of C possible classes and denote its class label by y_i. Our aim is to define an affinity matrix that encodes the notions of intra-class and inter-class distances, and thus, when solving (7), yields a mapping that minimizes the intra-class distances while simultaneously maximizing the inter-class distances (*i.e.*, a discriminative mapping).

More specifically, let $\{(\boldsymbol{X}_i, y_i)\}_{i=1}^p$ be the set of p labeled training points, where $\boldsymbol{X}_i \in \mathcal{S}_{++}^n$ and $y_i \in \{1, 2, \cdots, C\}$. The affinity of the training data on $\mathcal{S}_{++}^n$ can be modeled by building a within-class similarity graph $\boldsymbol{G}_w$ and a between-class similarity graph $\boldsymbol{G}_b$. In particular, we define $\boldsymbol{G}_w$ and $\boldsymbol{G}_b$ as binary matrices constructed from nearest neighbor graphs. This yields

$$\boldsymbol{G}_w(i, j) = \begin{cases} 1, \text{ if } \boldsymbol{X}_i \in N_w(\boldsymbol{X}_j) \text{ or } \boldsymbol{X}_j \in N_w(\boldsymbol{X}_i) \\ 0, \qquad\qquad\qquad \text{otherwise} \end{cases} \tag{9}$$

$$\boldsymbol{G}_b(i, j) = \begin{cases} 1, \text{ if } \boldsymbol{X}_i \in N_b(\boldsymbol{X}_j) \text{ or } \boldsymbol{X}_j \in N_b(\boldsymbol{X}_i) \\ 0, \qquad\qquad\qquad \text{otherwise} \end{cases} \tag{10}$$

where $N_w(\boldsymbol{X}_i)$ is the set of ν_w nearest neighbors of $\boldsymbol{X}_i$ that share the same label as y_i, and $N_b(\boldsymbol{X}_i)$ contains the ν_b nearest neighbors of $\boldsymbol{X}_i$ having different labels. The affinity matrix $\boldsymbol{A}$ is then defined as

$$\boldsymbol{A} = \boldsymbol{G}_w - \boldsymbol{G}_b \ , \tag{11}$$

which resembles the Maximum Margin Criterion (MMC) of [11]. In practice, we set ν_w to the minimum number of points in each class and, to balance the influence of $\boldsymbol{G}_w$ and $\boldsymbol{G}_b$, choose $\nu_b \leq \nu_w$, with the specific value found by cross-validation. We analyze the influence of ν_b in supplementary material.

4.3   Discussion in Relation to Region Covariance Descriptors

In our experiments, we exploited Region Covariance Matrices (RCMs) [20] as image descriptors. Here, we discuss some interesting properties of our algorithm when applied to these specific SPD matrices.

There are several reasons why RCMs are attractive to represent images and videos. First, RCMs provide a natural way to fuse various feature types. Second, they help reducing the impact of noisy samples in a region via their inherent averaging operation. Third, RCMs are independent of the size of the region, and can therefore easily be utilized to compare regions of different sizes. Finally, RCMs can be efficiently computed using integral images [21,18].

Let I be a $W \times H$ image, and $\mathbb{O} = \{\boldsymbol{o}_i\}_{i=1}^r$, $\boldsymbol{o}_i \in \mathbb{R}^n$ be a set of r observations extracted from I, $e.g.$, $\boldsymbol{o}_i$ concatenates intensity values, gradients along the horizontal and vertical directions, filter responses,... for image pixel i. Let $\mu = \frac{1}{r} \sum_{i=1}^r \boldsymbol{o}_i$ be the mean value of the observations. Then image I can be represented by the $n \times n$ RCM

$$\boldsymbol{C}_I = \frac{1}{r-1} \sum_{i=1}^r (\boldsymbol{o}_i - \mu)(\boldsymbol{o}_i - \mu)^T = \boldsymbol{O} \boldsymbol{J} \boldsymbol{J}^T \boldsymbol{O}^T \ , \tag{12}$$

where $\boldsymbol{J} = r^{-3/2}(r\mathbf{I}_r - \mathbf{1}_{r \times r})$. To have a valid RCM, $r \geq n$, otherwise $\boldsymbol{C}_I$ would have zero eigenvalues, which would make both δ_g^2 and δ_S^2 indefinite.

After learning the projection $\boldsymbol{W}$, the low-dimensional representation of image I is given by $\boldsymbol{W}^T \boldsymbol{O} \boldsymbol{J} \boldsymbol{J}^T \boldsymbol{O}^T \boldsymbol{W}$. This reveals two interesting properties of our learning scheme. **1)** The resulting representation can also be thought of as an RCM with $\boldsymbol{W}^T \boldsymbol{O}$ as a set of low-dimensional observations. Hence, in our framework, we can create a valid $\mathcal{S}_{++}^m$ manifold with only m observations instead of at least n in the original input space. This is not the case for other algorithms, which require having matrices on $\mathcal{S}_{++}^n$ as input. In supplementary material, we study the influence of the number of observations on recognition accuracy. **2)** Applying $\boldsymbol{W}$ directly the set of observations reduces the computation time of creating the final RCM on $\mathcal{S}_{++}^m$. This is due to the fact that the computational complexity of computing an RCM is quadratic in the dimensionality of the features.

5   Empirical Evaluation

In this section, we study the effectiveness of our SPD manifold learning approach. In particular, as mentioned earlier, we focus on classification and present results on two image datasets and one motion capture dataset. In all our experiments, the dimensionality of the low-dimensional SPD manifold was determined by cross-validation. Below, we first briefly describe the different classifiers used in these experiments, and then discuss our results.

Classification algorithms: The SPD-ML algorithm introduced in Section 4 allows us to obtain a low-dimensional, more discriminative SPD manifold from a high-dimensional one. Many different classifiers can then be used to categorize the data on this new manifold. In our experiments, we make use of two such classifiers. First, we employ a simple nearest neighbor classifier based on the manifold metric (either AIRM or Stein). This simple classifier clearly evidences the benefits of mapping the original Riemannian structure to a lower-dimensional one. Second, we make use of the Riemannian sparse coding algorithm of [6] (RSR). This algorithm exploits the notion of sparse coding to represent a query SPD matrix using a codebook of SPD matrices. In all our experiments, we formed the codebook purely from the training data, *i.e.*, no dictionary learning was employed. Note that RSR relies on a kernel derived from the Stein metric. We therefore only applied it to the Stein metric-based version of our algorithm. We refer to the different algorithms evaluated in our experiments as:

NN-Stein: Stein metric-based Nearest Neighbor classifier.
NN-AIRM: AIRM-based Nearest Neighbor classifier.
NN-Stein-ML: Stein metric-based Nearest Neighbor classifier on the low-dimensional SPD manifold obtained with our approach.
NN-AIRM-ML: AIRM-based Nearest Neighbor classifier on the low-dimensional SPD manifold obtained with our approach.
RSR: Riemannian Sparse Representation [6].
RSR-ML: Riemannian Sparse Representation on the low-dimensional SPD manifold obtained with our approach.

In addition to these methods, we also provide the results of the PLS-based Covariance Discriminant Learning (CDL) technique of [22], as well as of the state-of-the-art baselines of each specific dataset.

5.1   Material Categorization

For the task of material categorization, we used the UIUC dataset [12]. The UIUC material dataset contains 18 subcategories of materials taken in the wild from four general categories (see Fig. 2): *bark*, *fabric*, *construction materials*, and *outer coat of animals*. Each subcategory has 12 images taken at various scales. Following standard practice, half of the images from each subcategory was randomly chosen as training data, and the rest was used for testing. We report the average accuracy over 10 different random partitions.

Fig. 2. Samples from the UIUC material dataset

Small RCMs, such as those used for texture recognition in [6], are hopeless here due to the complexity of the task. Recently, SIFT features [13] have been shown to be robust and discriminative for material classification [12]. Therefore, we constructed RCMs of size 155×155 using 128 dimensional SIFT features (from grayscale images) and 27 dimensional color descriptors. To this end, we resized all the images to 400×400 and computed dense SIFT descriptors on a regular grid with 4 pixels spacing. The color descriptors were obtained by simply stacking colors from 3×3 patches centered at the grid points. Each grid point therefore yields one 155-dimensional observation o_i in Eq. 12. The parameters for this experiments were set to $\nu_w = 6$ (minimum number of samples in a class), and $\nu_b = 3$ obtained by 5-fold cross-validation.

Table 1 compares the performance of our different algorithms and of the state-of-the-art method on this dataset (SD) [12]. The results show that appropriate manifold-based methods (*i.e.*, RSR and CDL) with the original 155×155 RCMs already outperform SD, while NN on the same manifold yields worse performance. However, after applying our learning algorithm, NN not only outperforms SD significantly, but also outperforms both CDL and RSR. RSR on the learned SPD manifold (RSR-ML) further boosts the accuracy to 66.6%.

To further evidence the importance of geometry-aware dimensionality reduction, we replaced our low-dimensional RCMs with RCMs obtained by applying PCA directly on the 155 dimensional features. The AIRM-based NN classifier used on these RCMs gave 42.1% accuracy (best performance over different PCA dimensions). While this is better than the performance in the original feature space (*i.e.*, 35.6%), it is significantly lower than the accuracy of our NN-AIRM-ML approach (*i.e.*, 58.3%). Finally, note that performing NN-AIRM on the original data required 490s on a 3GHz machine with Matlab. After our dimensionality reduction scheme, this only took 9.7s.

### 5.2 Action Recognition from Motion Capture Data

As a second experiment, we tackled the problem of human action recognition from motion capture sequences using the HDM05 database [14]. This database contains the following 14 actions: 'clap above head', 'deposit floor', 'elbow to knee', 'grab high', 'hop both legs', 'jog', 'kick forward', 'lie down floor', 'rotate both arms backward', 'sit down chair', 'sneak', 'squat', 'stand up lie' and 'throw basketball' (see Fig. 3 for an example). The dataset provides the 3D locations of 31 joints over time acquired at the speed of 120 frames per second. We describe an action of a K joints skeleton observed over m frames by its joint covariance

Table 1. Mean recognition accuracies with standard deviations for the UIUC material dataset [12]

Method	Accuracy
SD [12]	$43.5\% \pm N/A$
CDL [22]	$52.3\% \pm 4.3$
NN-Stein	$35.8\% \pm 2.6$
NN-Stein-ML	$58.1\% \pm 2.8$
NN-AIRM	$35.6\% \pm 2.6$
NN-AIRM-ML	$58.3\% \pm 2.3$
RSR [6]	$52.8\% \pm 2.1$
RSR-ML	$\mathbf{66.6\% \pm 3.1}$

Table 2. Recognition accuracies for the HDM05-MOCAP dataset [14]

Method	Accuracy
CDL [22]	79.8%
NN-Stein	61.7%
NN-Stein-ML	68.6%
NN-AIRM	62.8%
NN-AIRM-ML	67.6%
RSR [6]	76.1%
RSR-ML	**81.9%**
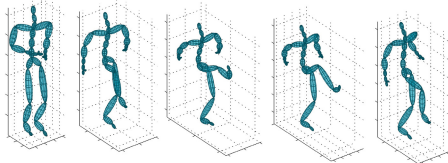
Fig. 3. Kicking action from the HDM05 motion capture sequences database [14]

descriptor [7], which is an SPD matrix of size $3K \times 3K$. This matrix is computed as in Eq. 12 by taking $\boldsymbol{o}_i$ as the 93-dimensional vector concatenating the 3D coordinates of the 31 joints in frame i.

In our experiments, we used 2 subjects for training (*i.e.*, 'bd' and 'mm') and the remaining 3 subjects for testing (*i.e.*, 'bk', 'dg' and 'tr')[1]. This resulted in 118 training and 188 test sequences for this experiment. The parameters of our method were set to $\nu_w = 5$ (minimum number of samples in one class), and $\nu_b = 5$ by cross-validation.

We report the performance of the different methods on this dataset in Table 2. Again we can see that the accuracies of NN and RSR are significantly improved by our learning algorithm, and that our RSR-ML approach achieves the best accuracy of 81.9%. As on the UIUC dataset, we also evaluated the performance RCMs built by reducing the dimensionality of the features using PCA. This yielded an accuracy of 63.3% with an AIRM-based NN classifier (best performance over different PCA dimensions). Again, while this slightly outperforms the accuracy of NN-AIRM (*i.e.*, 62.8%), it remains clearly inferior to the performance of our NN-AIRM-ML algorithm (*i.e.*, 67.6%).

---

[1] Note that this differs from the setup in [7], where 3 subjects were used for training and 2 for testing. However, with the setup of [7] where an accuracy of 95.41% was reported, all our algorithms resulted in about 99% accuracy.
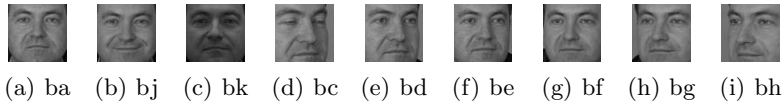
(a) ba   (b) bj   (c) bk   (d) bc   (e) bd   (f) be   (g) bf   (h) bg   (i) bh

Fig. 4. Samples from the FERET dataset [17]

5.3   Face Recognition

For face recognition, we used the 'b' subset of the FERET dataset [17], which contains 1800 images from 200 subjects. Following common practice [6], we used cropped images, downsampled to 64×64. Fig. 4 depicts samples from the dataset.

We performed six experiments on this dataset. In all these experiments, the training data was composed of frontal faces with expression and illumination variations (*i.e.*, images marked as 'ba', 'bj' and 'bk'). The six experiments correspond to using six different non-frontal viewing angles as test data (*i.e.*, images marked as 'bc','bd', 'be', 'bf', 'bg' and 'bh', respectively).

To represent a face image, we block diagonally concatenated three different 43×43 RCMs: one obtained from the entire image, one from the left half and one from the right half. This resulted in an RCM of size 129×129 for each image. Each 43×43 RCM was computed from the features

$$\boldsymbol{o}_{x,y} = [\; I(x,y),\; x,\; y,\; |G_{0,0}(x,y)|,\; \cdots,\; |G_{4,7}(x,y)|\;]\;,$$

where $I(x,y)$ is the intensity value at position (x,y), $G_{u,v}(x,y)$ is the response of a 2D Gabor wavelet [10] centered at (x,y) with orientation u and scale v, and $|\cdot|$ denotes the magnitude of a complex value. Here, following [6], we generated 40 Gabor filters at 8 orientations and 5 scales.

In addition to our algorithms, we evaluated the state-of-the-art Sparse Representation based Classification (SRC) [23] and its Gabor-based extension (GSRC) [25]. For SRC, we reduced the dimensionality of the data using PCA and chose the dimensionality that gave the best performance. For GSRC, we followed the recommendations of the authors to set the downsampling factor in the Gabor filters, but found that better results could be obtained with a larger λ than the recommended one, and thus report these better results obtained with $\lambda = 0.1$. The parameters for our approach were set to $\nu_w = 3$ (minimum number of samples in one class), and $\nu_b = 1$ by cross-validation.

Table 3 reports the performance of the different methods. Note that both CDL and RSR outperform the Euclidean face recognition systems SRC and GSRC. Note also that even a simple Stein-based NN on 129×129 RCMs performs roughly on par with GSRC and better than SRC. More importantly, the representation learned with our SPD-ML algorithm yields significant accuracy gains when used with either NN or RSR for all different viewing angles, with more than 10% improvement for some poses.

Table 3. Recognition accuracies for the FERET face dataset [17]

Method	bc	bd	be	bf	bg	bh	average acc.
SRC [23]	9.5%	37.5%	77.0%	88.0%	48.5%	11.0%	45.3% ± 3.3
GSRC [25]	35.5%	77.0%	93.5%	97.0%	79.0%	38.0%	70.0% ± 2.7
CDL [22]	35.0%	87.5%	**99.5%**	**100.0%**	91.0%	34.5%	74.6% ± 3.1
NN-Stein	29.0%	75.5%	94.5%	98.0%	83.5%	34.5%	69.2% ± 3.0
NN-Stein-ML	40.5%	88.5%	97.0%	99.0%	91.5%	44.5%	76.8% ± 2.7
NN-AIRM	28.5%	72.5%	93.0%	97.5%	83.0%	35.0%	68.3% ± 3.0
NN-AIRM-ML	39.0%	84.0%	96.0%	99.0%	90.5%	45.5%	75.7% ± 2.6
RSR [6]	36.5%	79.5%	96.5%	97.5%	86.0%	41.5%	72.9% ± 2.7
RSR-ML	**49.0%**	**90.5%**	98.5%	**100%**	**93.5%**	**50.5%**	**80.3% ± 2.4**

6   Conclusions and Future Work

We have introduced a learning algorithm to map a high-dimensional SPD manifold into a lower-dimensional, more discriminative one. To this end, we have exploited a graph embedding formalism with an affinity matrix that encodes intra-class and inter-class distances, and where the similarity between two SPD matrices is defined via either the Stein divergence or the AIRM. Thanks to their invariance to affine transformations, these metrics have allowed us to model the mapping from the high-dimensional manifold to the low-dimensional one with an orthonormal projection. Learning could then be expressed as the solution to an optimization problem on a Grassmann manifold. Our experimental evaluation has demonstrated that the resulting low-dimensional SPD matrices lead to state-of-the art recognition accuracies on several challenging datasets.

In the future, we plan to extend our learning scheme to the unsupervised and semi-supervised scenarios. Finally, we believe that this work is a first step towards showing the importance of preserving the Riemannian structure of the data when performing dimensionality reduction, and thus going from one manifold to another manifold of the same type. We therefore intend to study how this framework can be applied to other types of Riemannian manifolds.

References

1. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2008)
2. Caseiro, R., Henriques, J.F., Martins, P., Batista, J.: Semi-intrinsic mean shift on riemannian manifolds. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 342–355. Springer, Heidelberg (2012)
3. Cherian, A., Sra, S., Banerjee, A., Papanikolopoulos, N.: Jensen-bregman logdet divergence with application to efficient similarity search for covariance matrices. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(9), 2161–2174 (2013)

4. Fletcher, P.T., Lu, C., Pizer, S.M., Joshi, S.: Principal geodesic analysis for the study of nonlinear statistics of shape. IEEE Transactions on Medical Imaging 23(8), 995–1005 (2004)
5. Goh, A., Vidal, R.: Clustering and dimensionality reduction on riemannian manifolds. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–7. IEEE (2008)
6. Harandi, M.T., Sanderson, C., Hartley, R., Lovell, B.C.: Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 216–229. Springer, Heidelberg (2012)
7. Hussein, M.E., Torki, M., Gowayyed, M.A., El-Saban, M.: Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In: Proc. Int. Joint Conference on Artificial Intelligence, IJCAI (2013)
8. Jayasumana, S., Hartley, R., Salzmann, M., Li, H., Harandi, M.: Kernel methods on the riemannian manifold of symmetric positive definite matrices. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2013)
9. Jung, S., Dryden, I.L., Marron, J.: Analysis of principal nested spheres. Biometrika 99(3), 551–568 (2012)
10. Lee, T.S.: Image representation using 2d Gabor wavelets. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(10), 959–971 (1996)
11. Li, H., Jiang, T., Zhang, K.: Efficient and robust feature extraction by maximum margin criterion. IEEE Transactions on Neural Networks 17(1), 157–165 (2006)
12. Liao, Z., Rock, J., Wang, Y., Forsyth, D.: Non-parametric filtering for geometric detail extraction and material representation. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2013)
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
14. Müller, M., Röder, T., Clausen, M., Eberhardt, B., Krüger, B., Weber, A.: Documentation: Mocap database HDM05. Tech. Rep. CG-2007-2, Universität Bonn (2007)
15. Pang, Y., Yuan, Y., Li, X.: Gabor-based region covariance matrices for face recognition. IEEE Transactions on Circuits and Systems for Video Technology 18(7), 989–993 (2008)
16. Pennec, X., Fillard, P., Ayache, N.: A riemannian framework for tensor computing. Int. Journal of Computer Vision (IJCV) 66(1), 41–66 (2006)
17. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The feret evaluation methodology for face-recognition algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(10), 1090–1104 (2000)
18. Sanin, A., Sanderson, C., Harandi, M., Lovell, B.: Spatio-temporal covariance descriptors for action and gesture recognition. In: IEEE Workshop on Applications of Computer Vision (WACV), pp. 103–110 (2013)
19. Sra, S.: A new metric on the manifold of kernel matrices with application to matrix geometric means. In: Proc. Advances in Neural Information Processing Systems (NIPS), pp. 144–152 (2012)
20. Tuzel, O., Porikli, F., Meer, P.: Region covariance: A fast descriptor for detection and classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 589–600. Springer, Heidelberg (2006)
21. Tuzel, O., Porikli, F., Meer, P.: Pedestrian detection via classification on riemannian manifolds. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(10), 1713–1727 (2008)

22. Wang, R., Guo, H., Davis, L.S., Dai, Q.: Covariance discriminative learning: A natural and efficient approach to image set classification. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2496–2503. IEEE (2012)
23. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(2), 210–227 (2009)
24. Yan, S., Xu, D., Zhang, B., Zhang, H.J., Yang, Q., Lin, S.: Graph embedding and extensions: a general framework for dimensionality reduction. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(1), 40–51 (2007)
25. Yang, M., Zhang, L.: Gabor feature based sparse representation for face recognition with Gabor occlusion dictionary. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 448–461. Springer, Heidelberg (2010)