A Comprehensive Look at Coding Techniques on Riemannian Manifolds

Masoud Faraki, Mehrtash T. Harandi, and Fatih Porikli, *Fellow, IEEE*

***Abstract*—Core to many learning pipelines is visual recognition such as image and video classification. In such applications, having a compact yet rich and informative representation plays a pivotal role. An underlying assumption in traditional coding schemes (*e. g.* , sparse coding) is that the data geometrically comply with the Euclidean space. In other words, the data is presented to the algorithm in vector form and Euclidean axioms are fulfilled. This is of course restrictive in machine learning, computer vision and signal processing as shown by a large number of recent studies. This paper takes a further step and provides a comprehensive mathematical framework to perform coding in curved and non-Euclidean spaces, *i. e.*, Riemannian manifolds. To this end, we start by the simplest form of coding, namely bag of words. Then, inspired by the success of vector of locally aggregated descriptors in addressing computer vision problems, we will introduce its Riemannian extensions. Finally, we study Riemannian form of sparse coding, locality-constrained linear coding and collaborative coding. Through rigorous tests, we demonstrate the superior performance of our Riemannian coding schemes against state-of-the-art methods on several visual classification tasks including head pose classification, video-based face recognition and dynamic scene recognition.**

***Index Terms*—Riemannian Geometry, bag of words, vector of locally aggregated descriptors, sparse coding, locality-constrained linear coding, collaborative coding.**

I. Introduction

In this paper, we devise a frame-work to exploit state-of-the-art coding methods such as Vector of Locally Aggregated Descriptors (VLAD) [22] and Sparse Coding (SC) [40] where the local descriptors belong to a Riemannian manifold. Classical coding/aggregating techniques [22], [31], [24] are designed to work only with vectors (*i. e.*, local descriptors are points in $\mathbb{R}^n$). Specifically, compact VLAD codes have been shown to be exceptionally successful for face and texture classification [6], [10]. Lately, a few studies target the problem of coding/aggregation when the local descriptors are structured (*e. g.* , subspaces) and non-vectorial [17]. Inspired by the fact that describing images or videos by local descriptors is the method of choice [24], [22], [28] these days, we add a novel dimension to the applicability of such techniques by introducing a mathematical foundation for coding/aggregation of structured descriptors.

A diverse number of learning systems enjoy from representations of data that are compact yet discriminative, informative and robust to critical measurements. A notable example is the Diffusion Tensor Imaging (DTI) technique which represents each voxel in 3-D brain scans by a 3×3 Symmetric Positive Definite (SPD) matrix. It is now an accepted fact that analyzing the resulting diffusion tensors by vectorizing them deteriorates the performances heavily and can lead to solutions that are physically meaningless [1]. Another example of a structured descriptor is the Region Covariance Descriptor (RCovD) [36], successfully used in human detection [36], texture classification [9], human head pose estimation [35] and face recognition [38], [18], [19]. RCovDs offer compact and rich visual content representations by fusing various features while reducing the impact of noisy samples [36], [9]. Similarly, linear subspaces as structured descriptors offer a convenient platform to compensate for a wide range of image variations and have been used with promising results in image set and video classification [11], [17].

Despite their intriguing properties, analyzing the aforementioned descriptors is not straightforward as a result of their non-Eucliudean geometry. More specifically, diffusion tensors and RCovDs belong to the manifold of SPD matrices [27] and linear subspaces are points on the Grassmann manifold [8]. Although the two manifolds are Riemannian (*i. e.*, equipped with metrics), the lack of a vector space structure is a barrier for developing inference methods [2], [27], [36].

In this paper, we examine image and video-based recognition tasks where the local descriptors have the aforementioned Riemannian structures, namely the SPD or linear subspace structure. To be precise, we provide answers to the two following questions

- can we encode the local structured descriptors into a fixed length and discriminative vector?
- can we derive a universal mathematical scheme that enhances formulating the encoding problem into an elegant solution?

To this end, we begin by providing a solution to compute Riemannian version of the conventional VLAD, namely R-VLAD, using the geodesic distance of the underlying manifold as the nearness measure. Then, we clarify that the resulting codes are actually obtained from a new concept which we name Local Difference Vectors (LDV). Furthermore, analogues to the Higher-Order (HO-) VLAD [26], we also leverage higher order statistics of local structured descriptors for R-VLAD codes and make them more discriminative. Lastly, with the aid of the LDVs, we expand our Riemannian coding techniques and provide intrinsic solutions to Riemannian Sparse Coding (R-SC) and two of its variants, namely Riemannian version of the Locality-constrained Linear Coding [37] (R-

Authors are with the Research School of Engineering, Australian National University, Canberra, ACT 0200, Australia (e-mail:U5484403@anu.edu.au)

Masoud Faraki is also with Data61, CSIRO, Locked Bag 8001, Canberra, ACT 2601, Australia and Australian Centre for Robotic Vision located at Monash University, Scenic Blvd, Clayton VIC 3800, Melbourne, Australia (e-mail:masoud.faraki@{data61.csiro.au, roboticvision.org, monash.edu})

Mehrtash T. Harandi is also with Data61, CSIRO, Locked Bag 8001, Canberra, ACT 2601, Australia

LLC) and the Collaborative Coding [42] (R-CC).

With LDVs, we show that coding/aggregation with other metrics/closeness measures rather than geodesic distances is also possible. In other words, we do not confine ourselves to the geodesic distance case and develop the sister family of our methods by exploiting various well-known forms of similarity measures (*e. g.* , divergences) defined on the underlying manifolds. Our motivation is the fact that one can seamlessly use our general formulation with a metric suitable for a specific task at hand. For example, one may choose a divergence over the geodesic distance if computing geodesics is demanding. More specifically, we employ the Stein [32] and Jeffrey [39] divergences on the SPD manifold and the projection distance [16] on Grassmannian to obtain new variants of our solution. Last but not least, our contributions enable one to aggregate local descriptors residing on curved spaces. Therefore, conventional forms of coding/aggregation are indeed special cases of our universal scheme if the space is selected to be Euclidean.

II. Related Work

In this section, we review some relevant encoding methods to our proposals, such as Bag of Words (BoW), Vector of Locally Aggregated Descriptors (VLAD) and Sparse Coding (SC).

A. Bag of Words

While celebrating their third decade of birth, BoW [31]and its extensions [24] continue to be the baseline image and video representations. Various alternatives have been proposed to improve the discriminatory power of the original BoW model. Notable examples include Video Google [31] in which the resulting descriptor elements are graded by inverse document frequency terms and spatio-temporal pyramid matching [24] which considers the information about the spatial layout of features in the final image representation.

B. Vector of Locally Aggregated Descriptors

The VLAD descriptor, one of the main elements in this work, can be understood as a simpler version of the earlier Fisher Vectors (FV) derived from Fisher kernel [28]. Assuming that an incoming variable-sized set of descriptors follows a parametric generative model, FV can provide fixed-length descriptors by taking the gradients of the samples' likelihood with respect to the parameters of the distribution, weighted by the inverse square root of the Fisher information matrix. It has been shown that VLAD inherits the useful properties of FV by providing compact codes with relaxed assumptions on the origin of the samples and the scale of the output vector components (to be uniform) [22].

Peng *et al.* in [26] address the problem of enriching VLAD codes by higher-order statistics (called HVLAD) and supervised codebook learning (called SVLAD). The complimentary information in their HVLAD descriptor are second and third-order statistics which are obtained from covariance matrix and skewness measure of the points in each cluster. VLAD accuracy scores are further boosted by discriminatively learning the codebook in SVLAD.

C. Sparse Coding

Encoding a vector as linear combination of a few elements of an over-complete codebook is recognized as SC and has led to notable performances in various computer vision tasks [40], [44]. Another alternative to extend SC on non-linear spaces is through recasting the problem into Reproducing Kernel Hilbert Spaces (RKHS) via the kernel trick [7], [18], [17]. This leads to a convex quadratic problem which can be addressed conveniently. Another advantage of this method is that one could benefit from SC while having more separable samples in the resulting higher dimensional RKHS. Nevertheless, one is always obliged to find a valid kernel to be able to work on the manifold.

In [41], Xie *et al.* formulate the problem of sparse coding and dictionary learning on SPD manifolds using the Reimannian geodesic distances. To this end, they propose a coordinate-independent approach to reconstruct a given sample using affine linear combination of a small number of dictionary atoms. We will elaborate on this method more in § IV-C.

III. Background

In this section, we introduce some preliminary concepts such as Riemannian geometry and conventional VLAD coding which are of essential in our developments. Throughout the paper, we use bold lower-case letters (*e. g.* , $\boldsymbol{x}$) to show column vectors and bold upper-case letters (*e. g.* , $\boldsymbol{X}$) to show matrices. $[\cdot]_i$ is used to show the i-th element of a vector. $\mathbf{1}_n$ and $\mathbf{I}_n$ show vector of ones in $\mathbb{R}^n$ and the $n \times n$ identity matrix, respectively. ℓ_1 and ℓ_2 norms of a vector are denoted by $\|\boldsymbol{x}\|_1 = \sum_i |[\boldsymbol{x}]_i|$ and $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^T\boldsymbol{x}}$, respectively. The Frobenius norm of a matrix is shown by $\|\boldsymbol{X}\|_F = \sqrt{\mathrm{Tr}(\boldsymbol{X}^T\boldsymbol{X})}$, with $\mathrm{Tr}(\cdot)$ indicating the matrix trace. The determinant of a matrix is shown by $\det(\boldsymbol{X})$. Finally, $\log(\boldsymbol{X})$ is the principal logarithm of matrix $\boldsymbol{X}$.

A. Riemannian Geometry

A *manifold* $\mathcal{M}$ is a Hausdorff topological space which locally resembles a Euclidean space $\mathbb{R}^m$. The *tangent space* is a vector space attached to a point $\boldsymbol{P} \in \mathcal{M}$, $T_{\boldsymbol{P}}\mathcal{M}$, which consists of the tangent vectors of all possible curves on the manifold passing through $\boldsymbol{P}$ [27]. A *Riemannian manifold* is differential and equipped with a metric on the tangent spaces. In fact, the structure of a Riemannian manifold is specified by the metric. A *Riemannian metric* is a continuous collection of dot products on the tangent space at each point of the manifold. It is usually chosen to provide robustness to some geometrical transformations. Furthermore, it enables one to define lengths and angles on the manifold.

Smooth curves connect points on a Riemannian manifold. Having the Riemannian metric at the disposal, one can compute instantaneous speed (direction and magnitude) and length of a given curve. The curves yielding the minimum distance for any two points of the manifold are called *geodesics* and their length is the *geodesic distance*.

On a Riemannian manifold $\mathcal{M}$, let $\overrightarrow{\boldsymbol{pq}} \in T_{\boldsymbol{P}}\mathcal{M}$ be a tangent vector. For geodesically complete manifolds (the case in our

paper), there exists a unique geodesic starting at $\boldsymbol{P}$ associated with this tangent vector and hence $\overrightarrow{\boldsymbol{pq}}$ can identify a point $\boldsymbol{Q} \in \mathcal{M}$. The *exponential map* $\exp_{\boldsymbol{P}}(\cdot) : T_{\boldsymbol{P}}\mathcal{M} \to \mathcal{M}$, guaranties that the length of the tangent vector is equal to the geodesic distance. The logarithm map $\log_{\boldsymbol{P}}(\cdot) = \exp_{\boldsymbol{P}}^{-1}(\cdot) : \mathcal{M} \to T_{\boldsymbol{P}}\mathcal{M}$, projects a point on the manifold to the tangent space $T_{\boldsymbol{P}}\mathcal{M}$, *i. e.*, $\overrightarrow{\boldsymbol{pq}} = \log_{\boldsymbol{P}}(\boldsymbol{Q})$. We note that, both maps vary as the point $\boldsymbol{P}$ moves along $\mathcal{M}$.

B. The Manifold of Symmetric Positive Definite Matrices

A real $d \times d$ matrix $\boldsymbol{C}$ is SPD if and only if $\boldsymbol{z}^T\boldsymbol{C}\boldsymbol{z} > 0$ for every non-zero vector $\boldsymbol{z} \in \mathbb{R}^d$. $\mathcal{S}_{++}^d$ denotes the space formed of these SPD matrices which is a Lie group with a manifold structure. This allows one to use relevant concepts of differential geometry, *e. g.* , geodesics, when addressing $\mathcal{S}_{++}^d$. The tangent space at a point $\boldsymbol{X} \in \mathcal{S}_{++}^d$ is the set of all $d \times d$ symmetric matrices. Formally,

$$T_{\boldsymbol{X}}\mathcal{S}_{++}^d \triangleq \{\boldsymbol{\Delta} \in \mathbb{R}^{d\times d} : \boldsymbol{\Delta} = \boldsymbol{\Delta}^T\} . \tag{1}$$

RCovDs are SPD matrices and therefore it is essential to utilize Riemannian geometry to analyze them. Formally, a $d\times d$ RCovD can be constructed from a set of r observations $\mathbb{O} = \{\boldsymbol{o}_i\}_{i=1}^r$, $\boldsymbol{o}_i \in \mathbb{R}^d$, extracted from a region in an image (or a block in a video) as follows

$$\boldsymbol{C}_I = \frac{1}{r-1}\sum_{i=1}^{r}(\boldsymbol{o}_i - \overline{\boldsymbol{o}})(\boldsymbol{o}_i - \overline{\boldsymbol{o}})^T , \tag{2}$$

where $\overline{\boldsymbol{o}} = \frac{1}{r}\sum_{i=1}^r \boldsymbol{o}_i$.

$\mathcal{S}_{++}^d$ is mostly studied with the Riemannian structure induced by the Affine Invariant Riemannian Metric (AIRM) [27].

Definition 1. *The geodesic distance $\delta_G : \mathcal{S}_{++}^d \times \mathcal{S}_{++}^d \to \mathbb{R}^+$ derived from the AIRM is given by*

$$\delta_G(\boldsymbol{X}, \boldsymbol{Y}) \triangleq \|\log(\boldsymbol{X}^{-1/2}\boldsymbol{Y}\boldsymbol{X}^{-1/2})\|_F . \tag{3}$$

To measure similarities on SPD manifolds, two other types of symmetric Bregman divergences, namely the Stein [32] and the Jeffrey [39] divergences are also popular.

Definition 2. *One symmetric form of Bregman divergence is the Stein metric $\delta_S : \mathcal{S}_{++}^d \times \mathcal{S}_{++}^d \to \mathbb{R}^+$ which is defined as*

$$\delta_S^2(\boldsymbol{X}, \boldsymbol{Y}) \triangleq \ln\det\left(\frac{\boldsymbol{X}+\boldsymbol{Y}}{2}\right) - \frac{1}{2}\ln\det(\boldsymbol{X}\boldsymbol{Y}) . \tag{4}$$

Definition 3. *Another symmetric form of Bregman divergence is the Jeffrey (J or symmetric KL) divergence $\delta_J : \mathcal{S}_{++}^d \times \mathcal{S}_{++}^d \to \mathbb{R}^+$ given by*

$$\delta_J^2(\boldsymbol{X}, \boldsymbol{Y}) \triangleq \frac{1}{2}\operatorname{Tr}(\boldsymbol{X}^{-1}\boldsymbol{Y}) + \frac{1}{2}\operatorname{Tr}(\boldsymbol{Y}^{-1}\boldsymbol{X}) - d . \tag{5}$$

C. The Grassmann Manifold

To have a better understanding of the Grassmann manifold, we first define Stiefel manifold. The Stiefel manifold $S(p, d)$ is the set of $d \times p$, $0 < p < d$, matrices with orthonormal columns. More formally,

$$S(p, d) \triangleq \{\boldsymbol{X} \in \mathbb{R}^{d\times p} : \boldsymbol{X}^T\boldsymbol{X} = \mathbf{I}_p\} . \tag{6}$$

A point on the Grassmann manifold $\mathcal{G}_d^p$ is a subspace spanned by the columns of a full rank $d \times p$ matrix [8]. In other words, given that matrices are called equivalent if their columns span the same subspace of order p, equivalence classes of matrices of size $d \times p$ with orthonormal columns represent points on $\mathcal{G}_d^p$. The tangent space at a point $\boldsymbol{X} \in \mathcal{G}_d^p$ admits

$$T_{\boldsymbol{X}}\mathcal{G}_d^p \triangleq \{\boldsymbol{\Delta} \in \mathbb{R}^{d\times p} : \boldsymbol{X}^T\boldsymbol{\Delta} + \boldsymbol{\Delta}^T\boldsymbol{X} = 0\} . \tag{7}$$

The geodesic distance between two subspaces (or points on the Grassmannian) is defined as the magnitude of the smallest rotation that takes one point to the other [17].

Definition 4. *On the Grassmann manifold, the geodesic distance between two subspaces $\boldsymbol{X}$ and $\boldsymbol{Y}$ is defined as*

$$\delta_G(\boldsymbol{X}, \boldsymbol{Y}) \triangleq \|\Theta\| , \tag{8}$$

with $\Theta = [\theta_1, \theta_2, \cdots, \theta_p]$ denoting the vector of principal angles between the two subspaces [8].

Beside the geodesic distance on $\mathcal{G}_d^p$, the projection metric is also widely used.

Definition 5. *The projection distance, $\delta_P : \mathcal{G}_d^p \times \mathcal{G}_d^p \to \mathbb{R}^+$, between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is defined as [17], [16]*

$$\delta_P^2(\boldsymbol{X}, \boldsymbol{Y}) \triangleq \|\boldsymbol{X}\boldsymbol{X}^T - \boldsymbol{Y}\boldsymbol{Y}^T\|_F^2 . \tag{9}$$

D. Conventional Coding Methods

Let us assume that a set of local descriptors $\mathcal{X} = \{\boldsymbol{x}_t\}_{t=1}^m, \boldsymbol{x}_t \in \mathbb{R}^d$ extracted from an image or video and a codebook $\mathcal{D}$ with atoms $\{\boldsymbol{d}_i\}_{i=1}^k$, $\boldsymbol{d}_i \in \mathbb{R}^d$ are at our disposal. Coding algorithms represent each query point $\boldsymbol{x}$ as some function of codebook atoms $\boldsymbol{d}_i$. Furthermore, some additional constraints might be added to objective functions to impose useful structure on the codes and subsequently obtain a more discriminative representation.

1) VLAD: To review the VLAD method, we begin by studying its function in Euclidean spaces through its predecessor, the Fisher Vector (FV) [28]. FV encodes the set $\mathcal{X}$ into a high-dimensional vector representation by fitting a parametric generative model in the form of a Gaussian Mixture Model (GMM) with k components to the local descriptors, *i. e.*,

$$p(\boldsymbol{x}_t|\lambda) = \sum_{i=1}^{k}\omega_i\mathcal{N}(\boldsymbol{x}_t|\,\boldsymbol{\mu}_i, \Sigma_i) ,$$

where $\lambda = \{\omega_i, \boldsymbol{\mu}_i, \Sigma_i\}$ are the mixture weight, mean and covariance of the Gaussian components, respectively.

The FV descriptor is obtained by computing the gradients of the log-likelihood of the model with respect to its parameters (also known as the *score functions* in statistics). It leads to a representation that captures the contribution of the individual parameters to the generative process. Related to VLAD, is the first order differences between members of the set $\mathcal{X}$ and each of the GMM centers, *i. e.*,

$$\nabla_{\mu_i}\log p(\mathcal{X}|\lambda) = \sum_{t=1}^{m}\gamma_i(\boldsymbol{x}_t)\Sigma_i^{-1}(\boldsymbol{\mu}_i - \boldsymbol{x}_t) , \tag{10}$$

where $\gamma_i(\boldsymbol{x}_t)$ denotes the soft-assignment of $\boldsymbol{x}_t$ to the i-th Gaussian, *i. e.*,

$$\gamma_i(\boldsymbol{x}_t) = \frac{\omega_i\mathcal{N}(\boldsymbol{x}_t|\,\boldsymbol{\mu}_i, \Sigma_i)}{\sum_{j=1}^{k}\omega_j\mathcal{N}(\boldsymbol{x}_t|\,\boldsymbol{\mu}_j, \Sigma_j)} .$$

VLAD partitions the input vector space $\mathbb{R}^d$ into k clusters by learning a codebook $\mathcal{D}$ with atoms $\{\boldsymbol{d}_i\}_{i=1}^k$. Then, a descriptor $V \in \mathbb{R}^{kd}$ is generated by stacking k Local Difference Vectors

(LDV) $\boldsymbol{v}_i$ aggregating the differences $\boldsymbol{d}_i - \boldsymbol{x}_t$ in each cluster. More formally for the set $\mathcal{X}$,

$$\boldsymbol{v}_i = \sum_{\boldsymbol{x}_t \in \boldsymbol{d}_i} \boldsymbol{d}_i - \boldsymbol{x}_t \ , \tag{11}$$

where $\boldsymbol{x} \in \boldsymbol{d}_i$ denotes that the nearest codeword to $\boldsymbol{x}$ is $\boldsymbol{d}_i$.

Comparing Eq. (10) to Eq. (11), one can observe that

1) VLAD equally characterizes the distribution of local descriptors with respect to the centers. Hence, VLAD can be conceived as a non-probabilistic version of the FV.
2) In contrast to FV, VLAD assumes that the covariance matrices of the Gaussians are diagonal and fixed, *i. e.*, $\Sigma_i = \sigma \mathbf{I}_d, \ \forall i \in \{1, 2, \cdots, k\}$.

2) Sparse Coding: In Euclidean spaces, the idea of sparse coding is to reconstruct the input $\boldsymbol{x}$ through a linear combination of codebook elements, *i. e.*, $\boldsymbol{x} = \sum_{i=1}^{k} \boldsymbol{d}_i [\boldsymbol{y}]_i$, such that a small number of codewords is involved [40]. The problem of coding the single query input $\boldsymbol{x}_t$ can be formulated as solving the following minimization problem

$$\min_{\boldsymbol{y}} \ \left\| \boldsymbol{x}_t - \sum_{i=1}^{k} \boldsymbol{d}_i [\boldsymbol{y}]_i \right\|^2 + \lambda \|\boldsymbol{y}\|_1 \ , \tag{12}$$

where λ is the sparsity-promoter regularizer.

Since the codebook $\mathcal{D}$ is usually selected to be over-complete, *i. e.*, $k > d$, the regularization is essential to ensure that the under-determined system has a unique solution. Moreover, generally pooling methods such as average pooling or max pooling are performed on the resulting set $\mathcal{Y} = \{\boldsymbol{y}_t\}_{t=1}^{m}, \boldsymbol{y}_t \in \mathbb{R}^k$, to generate the final representation for the query set $\mathcal{X}$.

3) Locality-Constrained Linear Coding: Locality-Constrained Linear Coding (LLC) applies locality constraint to select similar atoms to the query and learns an affine combination of them to reconstruct the query [37]. An approximated LLC algorithm is proposed by Wang *et al.* [37] which first performs a K-nearest-neighbor (Knn) search and then analytically solves a constrained least squares problem. The affine combination of weights $\sum_{i=1}^{k} [\boldsymbol{y}]_i = 1$ (or equivalently $\mathbf{1}^T \boldsymbol{y} = 1$) is considered to ensure a shift invariant code is obtained

$$\begin{aligned} &\min_{\boldsymbol{y}} \left\| \boldsymbol{x}_t - \sum_{\boldsymbol{d}_i \in Knn(\boldsymbol{x}_t)} \boldsymbol{d}_i [\boldsymbol{y}]_i \right\|^2, \\ &\text{s.t.} \quad \mathbf{1}^{\mathrm{T}} \boldsymbol{y} = 1. \end{aligned} \tag{13}$$

4) Collaborative Coding: Zhang *et al.* [42] show that collaboratively reconstructing the query vector by codewords is effective for face recognition problem. To generate the face representation, a regularized least squares problem is solved as follows

$$\min_{\boldsymbol{y}} \ \left\| \boldsymbol{x}_t - \sum_{i=1}^{k} \boldsymbol{d}_i [\boldsymbol{y}]_i \right\|^2 + \lambda \|\boldsymbol{y}\|^2 \ , \tag{14}$$

where λ is the regularizer parameter.

Similar to the LLC coding, an analytic solution is obtained by zeroing out the derivative with respect to the variable $\boldsymbol{y}$. The induced sparsity is weaker than the original sparse coding method as the ℓ_2 norm is used for regularization.

IV. Riemannian Coding

In this section, we present our coding methods on Riemannian manifolds. In what follows, we assume that $\mathcal{X} = \{\boldsymbol{X}_t\}_{t=1}^{m}, \ \boldsymbol{X}_t \in \mathcal{M}$ is a set of local descriptors extracted from a query visual content (*e. g.* , an image) and $\mathcal{D} = \{\boldsymbol{D}_i\}_{i=1}^{k}, \ \boldsymbol{D}_i \in \mathcal{M}$ represents a codebook on a Riemannian manifold $\mathcal{M}$. Moreover, let $\delta(\cdot, \cdot) : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$ be a measure of similarity defined on $\mathcal{M}$.

A. Riemannian Bag of Words

In the simplest and most straightforward model, for the query set $\mathcal{X}$ and the codebook $\mathcal{D}$, a representation $\boldsymbol{y}$ is obtained by BoW algorithm using the hard assignment strategy [31]. In this case, a histogram $\boldsymbol{y} \in \mathbb{R}^k$ is obtained by assigning each query point $\boldsymbol{X}_t$ to its closest codeword from the set $\mathcal{D}$ using the given measure δ in $\mathcal{M}$. The i-th dimension of $\boldsymbol{y}$, $[\boldsymbol{y}]_i$, is obtained using $[\boldsymbol{y}]_i = \#\big(\boldsymbol{X}_t \in \boldsymbol{D}_i\big)$, where $\#(\cdot)$ denotes the number of occurrences. This obviously requires $m \times k$ comparisons. In the end, in order to add robustness to the number of extracted local descriptors, the resulting histogram is ℓ_2 normalized via $\widehat{\boldsymbol{y}} = \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|}$.

B. Riemannian Vector of Locally Aggregated Descriptors

The key inspiration in VLAD coding is that it has been successfully used in addressing many challenging tasks such as image retrieval [22], [15] and scene recognition [15]. The interest has even influenced the deep learning community [15], [6]. Besides, the discriminative representation obtained by VLAD is the result of rudimentary vector addition and subtraction. Another important merit is the reliance on small codebooks which further simplifies the learning stage and increases the popularity of VLAD.

Here, we develop a general framework for Riemannian VLAD (R-VLAD). To this end, we first start by devising R-VLAD on $\mathcal{M}$ when the similarity measure is the geodesic distance, *i. e.*, δ_G. We then discuss our universal solution in which any arbitrary similarity measure can take the role of δ_G and derive faster variants of R-VLAD. We conclude this section by introducing an approach to enrich R-VLAD by encoding more information about the distribution of the local descriptors and name it Higher Order R-VLAD (HO-R-VLAD).

1) R-VLAD: the geodesic distance scenario: A closer look at the signature generation steps of the conventional VLAD reveals that the LDVs are actually the gradient of the ℓ^2 norm (or simply the Euclidean distance). By discarding the associated normalization terms in the FV algorithm we arrive at equity of the FV and VLAD. Having said that, it is easy to conclude that R-VLAD signature on $\mathcal{M}$ is attained when the following tools are at the disposal

- metric δ which measures the nearness of the local descriptors to the codewords.
- addition and subtraction operators on $\mathcal{M}$.

Since $\mathcal{M}$ is a metric space, a natural way to address the first requirement is to choose the geodesic distance $\delta_G : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$. To address the second requirement, we note that on $\mathcal{M}$,

one can identify a vector $\overrightarrow{AB}$ as a vector of the tangent space at $\boldsymbol{A}$, *i. e.*, $T_{\boldsymbol{A}}\mathcal{M}$. As a result, the subtraction operator on $\mathcal{M}$ can be obtained using the logarithm map, $\log_{\boldsymbol{A}}(\cdot) : \mathcal{M} \to T_{\boldsymbol{A}}\mathcal{M}$. The concept of vector subtraction via the logarithm map has been used before. For instance, in [27] for addressing the problem of interpolation, in [21] for sparse coding and in [13] for performing dimensionality reduction. This discussion hints towards proposing the R-VLAD on a Riemannian manifold as follows

- utilize the geodesic distance of $\mathcal{M}$ to identify the nearest local descriptors to the codewords.
- exploit the tangent space attached to each codeword on $\mathcal{M}$, to build a Riemannian LDV for the codewords.

It is worth noting that, no further consideration (such as parallel transport) is required here, as the pole of the tangent space ($\boldsymbol{D}_i$ in our case) is fixed. In other words, the outputs of the logarithm map are compatible with each other [1]. As such, Eq. (11) on $\mathcal{M}$ has the following form

$$\boldsymbol{v}_i = \sum_{\boldsymbol{X}_t \in \boldsymbol{D}_i} \log_{\boldsymbol{D}_i}(\boldsymbol{X}_t) \;, \tag{15}$$

where $\log_{\boldsymbol{D}_i}(\cdot)$ is the logarithm map to the tangent space $T_{\boldsymbol{D}_i}$.

Although being perfectly accurate, the computational load of δ_G seems to be the sticking point as it leads to complex and slow algorithms, especially in our case where we have several local descriptors per query image/video. To alleviate this limitation, several studies recommend faster alternatives with excellent theoretical properties and similar results in practice [39], [5], [2], [16]. This motivates us to engage other valid metrics and devise a universal form for our R-VLAD.

2) R-VLAD: arbitrary metric scenario: For an arbitrary metric δ, the second requirement should only be taken into account. Since in the Euclidean case, the LDV can be imagined as the gradient of the distance function (see § III-D1), it is natural to define the LDV on $\mathcal{M}$ as $\sum_{\boldsymbol{X}_t \in \boldsymbol{D}_i} \nabla_{\boldsymbol{D}_i} \delta^2(\boldsymbol{D}_i, \boldsymbol{X}_t)$[2]. This idea is reinforced even more by the following theorem.

Theorem 1. *For a Riemannian manifold $\mathcal{M}$, the gradient of the geodesic distance function, $\delta_G : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$ is*

$$\nabla_{\boldsymbol{X}} \delta_G^2(\boldsymbol{X}, \boldsymbol{Y}) = -2 \log_{\boldsymbol{X}}(\boldsymbol{Y}). \tag{16}$$

Proof. The interested reader is referred to [34]. □

In practice, choosing $\nabla_{\boldsymbol{D}_i} \delta^2(\boldsymbol{D}_i, \boldsymbol{X}_t)$ for LDV, does not provide a solution. The main reason is that, the norm of $\nabla_{\boldsymbol{X}} \delta_G^2(\boldsymbol{X}, \boldsymbol{Y})$ is directly related to the metric δ_G. More formally,

$$\|\nabla_{\boldsymbol{X}} \delta_G^2(\boldsymbol{X}, \boldsymbol{Y})\|^2 = 4\| \log_{\boldsymbol{X}}(\boldsymbol{Y})\|^2 = 4\delta_G^2(\boldsymbol{X}, \boldsymbol{Y}).$$

This is indeed inherited to the Euclidean space case where the metric is selected to be the geodesic distance (or equivalently the Euclidean distance). However, this is not valid for any arbitrary metric as illustrated by the following counterexample.

[1]To be precise, this argument is valid as long as $\boldsymbol{x}_t$ is not in the cut locus of $\boldsymbol{c}_i$. This is of course not a very restricting assumption as in many manifolds (*e. g.* , the SPD manifold) the cut locus is indeed empty.

[2]On an abstract Riemannian manifold $\mathcal{M}$, the gradient of a smooth real function f at a point $\boldsymbol{X} \in \mathcal{M}$, denoted by $\nabla_x f$, is the element of $T_x\mathcal{M}$ satisfying $\langle \nabla_x f, \zeta \rangle_x = Df_x[\zeta]$ for all $\zeta \in T_x\mathcal{M}$, where $Df_x[\zeta]$ denotes the directional derivative of f at x in the direction of ζ.

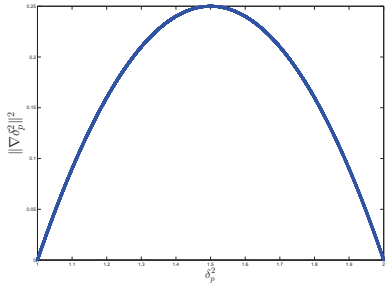

Fig. 1: Behavior of the squared norm of the gradient against distance function for the projection distance on $\mathcal{G}_3^2$.

Algorithm 1 The proposed R-VLAD technique

Input:
- local descriptors $\mathcal{X} = \{\boldsymbol{X}_t\}_{t=1}^m, \boldsymbol{X}_t \in \mathcal{M}$, obtained from a query image/video,
- codebook $\mathcal{D} = \{\boldsymbol{D}_i\}_{i=1}^k, \boldsymbol{d}_k \in \mathcal{M}$

Output:
- $\boldsymbol{V}(\mathcal{X})$ the R-VLAD representation of $\mathcal{X}$

1: **for** $i = 1 \to k$ **do**
2: Find all $\boldsymbol{X}_t \in \boldsymbol{D}_i$, the closest points from the query set $\mathcal{X}$ to $\boldsymbol{D}_i$
3: Compute $\boldsymbol{v}_i$, the i-th LDV, using Eq. (17)
4: **end for**
5: Concatenate the resulting LDVs to construct the final signature, *i. e.*, $\boldsymbol{V}(\mathcal{X}) = \left[\boldsymbol{v}_1^T, \boldsymbol{v}_2^T, \cdots, \boldsymbol{v}_k^T\right]^T$

Example 1. *Fig. 1 illustrates the behavior of* $\|\nabla_{\boldsymbol{X}} \delta^2(\boldsymbol{X}, \boldsymbol{Y})\|^2$, i. e., *squared norm of gradient by varying distance function* $\delta^2(\boldsymbol{X}, \boldsymbol{Y})$, *for the projection metric on* $\mathcal{G}_3^2$ *(equations are provided in § IV-B4). Interestingly,* $\|\nabla_{\boldsymbol{X}} \delta^2(\boldsymbol{X}, \boldsymbol{Y})\|^2$ *will start decreasing as the point* $\boldsymbol{Y}$ *gets farther away from the point* $\boldsymbol{X}$. *As a result, during coding, a point which should contribute significantly to the descriptor, acts as an unimportant point. This deteriorates the discriminatory power.*

The above example provides us with the following guideline for constructing an LDV on a Riemannian manifold.

- the length of the LDV must represent the metric.

As such, we propose the following form of LDV for our general R-VLAD descriptor. Algorithm 1 summarizes the steps on the R-VLAD coding.

$$\boldsymbol{v}_i = \sum_{\boldsymbol{X}_t \in \boldsymbol{D}_i} \psi_\delta(\boldsymbol{D}_i, \boldsymbol{X}_t) \;, \tag{17}$$

where $\psi_\delta(\boldsymbol{D}, \cdot) : \mathcal{M} \times \mathcal{M} \to T_{\boldsymbol{D}}\mathcal{M}$ is defined as

$$\psi_\delta(\boldsymbol{D}_i, \boldsymbol{X}_t) = \delta(\boldsymbol{D}_i, \boldsymbol{X}_t) \frac{\nabla_{\boldsymbol{D}_i} \delta^2(\boldsymbol{D}_i, \boldsymbol{X}_t)}{\|\nabla_{\boldsymbol{D}_i} \delta^2(\boldsymbol{D}_i, \boldsymbol{X}_t)\|} \;.$$

Remark 1. *To avoid having a concentrated distribution around zero in our experiments, we normalize the R-VLAD descriptors using the following two steps. First, a power normalization is applied utilizing the function* $y : \mathbb{R} \to \mathbb{R}$, $y(x) = sign(x)\sqrt{|x|}$, *where* x *is an element of the descriptor and* $|\cdot|$ *shows absolute value. Second, an* ℓ_2 *normalization is performed to make the energy of the codes uniform. This post-processing is compatible with the recommendations in [22] and could increase the discriminatory power of the final descriptors.*

Table I presents the gradients of all the studied metrics

TABLE I: Gradients of the metrics on the SPD and Grassmann manifolds.

Metric	$\nabla_{\boldsymbol{X}}\delta^2$
geodesic	$2\boldsymbol{X}^{1/2}\log(\boldsymbol{X}^{-1/2}\boldsymbol{Y}\boldsymbol{X}^{-1/2})\boldsymbol{X}^{1/2}$
Stein	$\boldsymbol{X}(\boldsymbol{X}+\boldsymbol{Y})^{-1}\boldsymbol{X}-\frac{1}{2}\boldsymbol{X}$
Jeffrey	$\frac{1}{2}\boldsymbol{X}\big(\boldsymbol{Y}^{-1}-\boldsymbol{X}^{-1}\boldsymbol{Y}\boldsymbol{X}^{-1}\big)\boldsymbol{X}$
geodesic	No analytic form
projection	$-4\big(\mathbf{I}_d-\boldsymbol{X}\boldsymbol{X}^T\big)\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{X}$

(required by Eq. (17)). In the following two parts, we devise the R-VLAD technique for the SPD and the Grassmann manifolds.

3) R-VLAD on SPD Manifold: On $\mathcal{S}^d_{++}$, the gradient of a function $f:\mathcal{S}^d_{++}\to\mathbb{R}$ at $\boldsymbol{X}$ admits the following form [33]

$$\nabla_{\boldsymbol{X}} f = \boldsymbol{X}\mathrm{sym}(Df)\boldsymbol{X}, \tag{18}$$

where $\mathrm{sym}(\boldsymbol{X})=0.5(\boldsymbol{X}+\boldsymbol{X}^T)$ and Df is the derivative of the function $f:\mathbb{R}^{d\times d}\to\mathbb{R}$ w.r.t $\boldsymbol{X}$.

As for the $D\delta_S^2$ and $D\delta_J^2$, we can infer the gradients from [5], required in the R-VLAD algorithm as depicted in Table I[3].

Computational Complexity: The computational load of building R-VLAD descriptor is dictated by the computational load of the incorporated metric δ^2 as wells as its gradient. Besides, one should consider the complexity of codebook learning on $\mathcal{M}$. The computational loads of computing δ_G^2, δ_J^2 and δ_S^2 are $4d^3$, $8/3d^3$ and d^3, respectively [5]. Computing the gradient of δ_G^2 needs an eigenvalue decomposition (for computing principal matrix logarithm) which adds up to a total of $9d^3$ flops for δ_G^2 (considering the matrix multiplications). For δ_J^2 and δ_S^2, computing the gradients only requires a matrix inversion which can be computed in $O(d^3)$. Therefore, the computational complexity of R-VLAD with δ_J^2 and δ_S^2 is $O(17/3d^3)$ and $O(4d^3)$, respectively.

4) R-VLAD on Grassmannian: The gradient of a function on the Grassmann manifold, *i. e.*, $f:\mathcal{G}_d^p\to\mathbb{R}$ admits the following form

$$\nabla_{\boldsymbol{X}} f = \Big(\mathbf{I}_d-\boldsymbol{X}\boldsymbol{X}^T\Big)\,Df, \tag{19}$$

where Df is a $d\times p$ matrix of partial derivatives of f w.r.t the elements of $\boldsymbol{X}$. Formally

$$[Df]_{i,j}=\frac{\partial f}{\partial[\boldsymbol{X}]_{i,j}}.$$

On the Grassmann manifold, the logarithm and exponential maps do not have an analytic form. Nevertheless, numerical methods exist for computing both mappings. Furthermore, to compute R-VLAD using the geodesic distance, we particularly exploit the developments presented in [4]. As long as the gradient of the projection metric is considered, noting that $\delta_P^2(\boldsymbol{X},\boldsymbol{Y})=2p-2\|\boldsymbol{X}^T\boldsymbol{Y}\|_F^2$ along with Eq. (19) leads us to the following analytic form

$$\nabla_{\boldsymbol{X}}\delta_P^2(\boldsymbol{X},\boldsymbol{Y}) = -4\Big(\mathbf{I}_d-\boldsymbol{X}\boldsymbol{X}^T\Big)\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{X}. \tag{20}$$

[3]Note that in Table 3 of [5] a scalar factor of 0.5 is wrongly dropped from the Jeffrey divergence (KLDM according to [5]). Also please note that the gradient reported in [5] is the Euclidean gradient not the Riemannian as required here.

Computational Complexity: We note that δ_G^2 on the Grassmann manifold is computed using Singular Value Decomposition (SVD). Therefore, obtaining δ_G^2 needs dp^2+p^3 flops on $\mathcal{G}_d^p$. In comparison, the load of computing δ_P^2 on Grassmannian is dp^2. As for δ_G^2, computing the gradient (or equivalently the logarithm map) needs a matrix inversion of size $p\times p$, two matrix multiplications of size $d\times p$ and a thin SVD of size $d\times p$. Computing thin SVD using an efficient implementation like the Golub-Reinsch [14] takes $14dp^2+8p^3$ flops. As such, a total of $O\big(10p^3+17dp^2\big)$ flops is required for one local descriptor. As for δ_p^2, computing the gradient according to the Table I demands $4dp^2$ operations. This results in a total of $5dp^2$ flops for δ_p^2.

5) Boosted R-VLAD: In this part, we present a variant of R-VLAD which in most cases further boosts the classification accuracy. We first note that the original VLAD formulation only considers simple first-order statistics of the LDVs to generate the final descriptor. Peng *et al.* [26] address this issue and introduce coding of higher-order statistics into the VLAD framework. Assuming training data is clustered using a codebook, the idea is to compute two additional super vectors associated to each cluster, capturing the deviation of the LDVs from qualitative measures, namely the diagonal elements of the covariance matrix and the skewness of the training samples. Similar in spirit to VLAD, the two forms of high-order statistics are coded as complementary information.

Here, we further expand this idea to exploit complementary information and adapt it to our R-VLAD descriptor. To this end, we use the definition of LDV in § IV-B2. Let the vector $\boldsymbol{\sigma_i}$ denotes diagonal elements of a covariance matrix constructed from the LDVs associated to $\boldsymbol{D}_i$ (training samples that are the closest to $\boldsymbol{D}_i$). In our case, the j-th element of the second-order super vector is computed as follows

$$[\boldsymbol{v}_i^{o^2}]_j=\frac{1}{\#(\boldsymbol{X}_t\in\boldsymbol{D}_i)}\sum_{\boldsymbol{X}_t\in\boldsymbol{D}_i}\big[\psi_\delta(\boldsymbol{D}_i,\boldsymbol{X}_t)\big]_j^2-\big[\sigma_i\big]_j^2, \tag{21}$$

with ψ defined below Eq. (17).

As for encoding the third-order statistics, skewness takes up the role of the diagonal elements of $\boldsymbol{\sigma_i}$

$$[\boldsymbol{v}_i^{o^3}]_j=\frac{\frac{1}{\#(\boldsymbol{X}_t\in\boldsymbol{D}_i)}\sum_{\boldsymbol{X}_t\in\boldsymbol{D}_i}\big[\psi_\delta(\boldsymbol{D}_i,\boldsymbol{X}_t)\big]_j^3}{\Big[\frac{1}{\#(\boldsymbol{X}_t\in\boldsymbol{D}_i)}\sum_{\boldsymbol{X}_t\in\boldsymbol{D}_i}\big[\psi_\delta(\boldsymbol{D}_i,\boldsymbol{X}_t)\big]_j^2\Big]^{\frac{3}{2}}}-\big[\Gamma_i\big]_j, \tag{22}$$

where Γ_i is the skewness vector of the training LDVs belonging to the i-th codeword.

The two super vectors $\boldsymbol{v}_i^{o^2}$ and $\boldsymbol{v}_i^{o^3}$ are concatenated and augmented to the original R-VLAD to form the final image/video signature. The power normalization is also performed in the end. We will dub this solution as Higher Order R-VLAD (HO-R-VLAD) in our experiments.

C. Riemannian Sparse Coding

As discussed earlier (see § III-D2), the goal of sparse coding is to find a sparse vector of coefficients $\boldsymbol{y}$ in a way that a query point $\boldsymbol{x}$ is as close as possible to the linear combination $\sum_{i=1}^k \boldsymbol{d}_i[\boldsymbol{y}]_i$. While in $\mathbb{R}^n$, this problem seems to be well formulated, the difficulty arises when the query point

(and subsequently each $\boldsymbol{d}_i$) belongs to $\mathcal{M}$, mainly because a universal coordinate system does not exist on $\mathcal{M}$. One natural modification to the notion of usual sparse coding is introduced by [41] in which the term $\boldsymbol{x}_t - \sum_{i=1}^{k} \boldsymbol{d}_i[\boldsymbol{y}]_i$ in Eq. (12) is generalized for $\boldsymbol{X} \in \mathcal{M}$. The affine constraint $\mathbf{1}^T\boldsymbol{y} = 1$ is imposed to the code to avoid having the trivial solution $\boldsymbol{y} = 0$. The sparse coding using the geodesic distance is cast as

$$\min_{\boldsymbol{y}} \sum_{i=1}^{k} \left\| \log_{\boldsymbol{X}} \left(\boldsymbol{D}_i\right) \right\|^2 [\boldsymbol{y}]_i + \lambda \|\boldsymbol{y}\|_1 \,, \quad \text{s.t.} \quad \mathbf{1}^{\mathrm{T}}\boldsymbol{y} = 1. \tag{23}$$

where $\log_{\boldsymbol{X}}(\cdot)$ is the logarithm map to the tangent space $T_{\boldsymbol{X}}$ and λ is the sparsity-promoter regularizer [40].

With the aid of LDVs defined in § IV-B2, we generalize the affine sparse coding scheme to be used with an arbitrary metric δ. Our idea is to perform coding by minimizing the following objective function

$$\min_{\boldsymbol{y}} \sum_{i=1}^{k} \left\| \psi_\delta(\boldsymbol{X}, \boldsymbol{D}_i) \right\|^2 [\boldsymbol{y}]_i + \lambda \|\boldsymbol{y}\|_1, \quad \text{s.t.} \quad \mathbf{1}^{\mathrm{T}}\boldsymbol{y} = 1. \tag{24}$$

where ψ is defined below Eq. (17).

Similar to $\mathbb{R}^n$, the final descriptor of the set $\mathcal{X}$ is obtained by pooling the resulting $\{\boldsymbol{y}_t\}_{t=1}^m$ codes. We refer to this method as Riemannian Sparse Coding (R-SC) in our experiments.

D. Riemannian Locality-constrained Linear Coding

Similar in spirit to sparse coding is the Locality-constrained Linear Coding (LLC) [37] in which the sparsity is a by-product of the locality constraint. LLC is easy to compute and gives superior image classification performance than many sophisticated approaches [37]. The locality constraint is applied to select similar atoms of a codebook for coding. Like sparse coding, the goal is to learn a linear combination of the chosen atoms to reconstruct each query point.

Similar to LLC in $\mathbb{R}^n$, we have the luxury of a closed-form solution for our non-linear LLC. Having a metric δ at our disposal, for a query point $\boldsymbol{X} \in \mathcal{M}$, we first find $n \ll k$ nearest neighbors from atoms of $\mathcal{D}$ and then construct matrix $\boldsymbol{C}$ by stacking $\psi_\delta(\boldsymbol{X}, \boldsymbol{D}_i)$ selected vectors as its columns, *i. e.*, $\boldsymbol{C} = \left[\psi_\delta(\boldsymbol{X}, \boldsymbol{D}_1) | \psi_\delta(\boldsymbol{X}, \boldsymbol{D}_2) | \cdots | \psi_\delta(\boldsymbol{X}, \boldsymbol{D}_n)\right]$. Then, the LLC code $\boldsymbol{y}$ is obtained by solving the following constrained least squares problem

$$\min_{\boldsymbol{y}} \left\| \boldsymbol{C}\boldsymbol{y} \right\|^2, \quad \text{s.t.} \quad \mathbf{1}^{\mathrm{T}}\boldsymbol{y} = 1. \tag{25}$$

Here, again the affine constraint $\mathbf{1}^T\boldsymbol{y} = 1$ is imposed to avoid having the trivial solution $\boldsymbol{y} = \mathbf{0}$. As such, using the Lagrange multipliers technique, the code $\boldsymbol{y}$ is obtained in closed-form as

$$\boldsymbol{y} = \frac{\left(\boldsymbol{C}^T\boldsymbol{C}\right)^{-1}\mathbf{1}}{\mathbf{1}^T\left(\boldsymbol{C}^T\boldsymbol{C}\right)^{-1}\mathbf{1}} \,. \tag{26}$$

In practice, a numerically stable way to minimize Eq. (26) is obtained through solving the set of n linear equations $\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{y} = 0$ followed by rescaling the coefficients $\boldsymbol{y}_i$ to ensure that $\mathbf{1}^T\boldsymbol{y} = 1$ [29]. We will dub this solution as Riemannian Locality-constrained Linear Coding (R-LLC) in our experiments.

E. Riemannian Collaborative Coding

In contrast to LLC, Collaborative Coding (CC) uses all dictionary atoms to represent the query sample. In the original CC, a regularized least squares problem is solved. Using δ, for a query point $\boldsymbol{X} \in \mathcal{M}$, we first construct matrix $\boldsymbol{C}$ by stacking $\psi_\delta(\boldsymbol{X}_t, \boldsymbol{D}_i)$ vectors as its columns, *i. e.*, $\boldsymbol{C} = \left[\psi_\delta(\boldsymbol{X}, \boldsymbol{D}_1) | \psi_\delta(\boldsymbol{X}, \boldsymbol{D}_2) | \cdots | \psi_\delta(\boldsymbol{X}, \boldsymbol{D}_k)\right]$. Then, the code $\boldsymbol{y} \in \mathbb{R}^k$ is obtained by solving the following constrained regularized least squares problem

$$\min_{\boldsymbol{y}} \left\| \boldsymbol{C}\boldsymbol{y} \right\|^2 + \lambda \|\boldsymbol{y}\|^2, \quad \text{s.t.} \quad \mathbf{1}^{\mathrm{T}}\boldsymbol{y} = 1. \tag{27}$$

To obtain the solution, again we use the Lagrange multipliers technique. Following a similar procedure to R-LLC, we obtain $\boldsymbol{y}$ as

$$\boldsymbol{y} = \frac{\left(\boldsymbol{C}^T\boldsymbol{C} + \lambda\boldsymbol{I}_k\right)^{-1}\mathbf{1}}{\mathbf{1}^T\left(\boldsymbol{C}^T\boldsymbol{C} + \lambda\boldsymbol{I}_k\right)^{-1}\mathbf{1}} \,. \tag{28}$$

We will dub this solution as Riemannian Collaborative Coding (R-CC) in our experiments.

V. K-Means on Riemannian Manifolds

Before delving into experiments and for the sake of completeness, we provide details of learning a Riemannian codebook using different metrics introduced previously in §III. Like many other codebook learning algorithms, mean computation is a fundamental building block in our proposal. Therefore, we define the Fréchet mean which is incorporated in our Riemannian codebook learning.

Definition 6. *The Fréchet mean for a set of points $\{\boldsymbol{X}_i\}_{i=1}^n$, $\boldsymbol{X}_i \in \mathcal{M}$ is the minimizer of the cost function*

$$\boldsymbol{D}^* \triangleq \arg\min_{\boldsymbol{D}} \sum_{i=1}^{n} \delta^2(\boldsymbol{D}, \boldsymbol{X}_i) \,, \tag{29}$$

where $\delta : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$ is the associated metric.

Generally, an analytic solution for Eq. (29) may not exist and hence iterative solutions that employ the logarithm and exponential maps must be used [27]. In case of high-dimensional manifolds, this could not be manageable.Therefore, one reason in generalizations introduced in the previous section is that for some metrics Eq. (29) has analytic solution.

Similar in concept to the standard k-means algorithm, we train a codebook using an iterative scheme. The Riemannian k-means algorithm initiates by choosing k random points from the available training data and calling them cluster centers. In one step, all the training samples are assigned to their closest cluster center using the metric δ. Then, in the next step, the cluster centers are re-estimated using the Fréchet mean.

On the SPD manifold and for the δ_G, the Fréchet mean is obtained by an iterative algorithm (see [27] for more details). In case of the Stein metric, we make use of the below theorem.

Theorem 2. *The Fréchet mean for a set of SPD matrices* $\{\boldsymbol{X}_i\}_{i=1}^n \in \mathcal{S}_{++}^d$ *with* δ_S *is computed iteratively through*

$$\boldsymbol{\mu}^{(t+1)} = \left[\frac{1}{n}\sum_{i=1}^{n}\Big(\frac{\boldsymbol{X}_i + \boldsymbol{\mu}^{(t)}}{2}\Big)^{-1}\right]^{-1}. \tag{30}$$

Proof. See [5] for the proof. □

Unlike δ_G and δ_S which we do not have an analytic solution, for the Jeffrey divergence, we have the luxury of computing the Fréchet mean in closed-form.

Theorem 3. *The Fréchet mean for a set of SPD matrices* $\{\boldsymbol{X}_i\}_{i=1}^n \in \mathcal{S}_{++}^d$ *with* δ_J *is*

$$\boldsymbol{\mu} = \boldsymbol{P}^{-1/2}(\boldsymbol{P}^{1/2}\boldsymbol{Q}\boldsymbol{P}^{1/2})^{1/2}\boldsymbol{P}^{-1/2}\,, \tag{31}$$

where $\boldsymbol{P} = \sum_i \boldsymbol{X}_i^{-1}$ *and* $\boldsymbol{Q} = \sum_i \boldsymbol{X}_i$.

Proof. See [10] for the proof. □

Similarly, the projection metric admits the following property.

Theorem 4. *The Fréchet mean for a set of points* $\{\boldsymbol{X}_i\}_{i=1}^n$, $\boldsymbol{X}_i \in \mathcal{G}_d^p$, *under* δ_P *is the* p *leading (largest) eigenvectors of* $\sum_{i=1}^n \boldsymbol{X}_i\boldsymbol{X}_i^T$.

Proof. See [10] for the proof. □

VI. Experiments

In this part, we provide empirical evaluation of our coding algorithms versus the baseline and state-of-the-art methods for a number of visual recognition tasks defined on the SPD and Grassmann manifolds. Unless otherwise stated, a number of overlapping blocks/cubes are extracted from images/videos. Then, from each block/cube, we generate an RCovD or a linear subspace, hence the block/cube corresponds to a point on the SPD or the Grassmann manifold.

The straightforward Log-Euclidean (LE) [2] alternative of devising VLAD on a Riemannian manifold constitutes our first type of base-line. Basically in the LE modeling, the manifold is embedded into a vector space through a fixed tangent space (centered at the identity matrix in our case). Furthermore, we will consider the popular BoW representation of an image or video as another base-line method. Different algorithms tested in this section are referred to as

BoW$_{LE}$: Riemannian BoW algorithm trained by flattening the underlying manifold via the identity tangent space.

R-BoW$_G$: Riemannian BoW algorithm using geodesic distance.

R-VLAD$_{LE}$: Similar in spirit to the **BoW**$_{LE}$, but we assess the performance of VLAD model, instead of BoW.

R-VLAD$_{G/J/S/P}$: R-VLAD model utilizing geodesic distance, the Jeffrey, Stein, or projection metrics.

HO-R-VLAD$_{G/J/S/P}$: Higher-Order R-VLAD utilizing geodesic distance, the Jeffrey, Stein, or projection metrics.

R-SC$_{G/J/S/P}$: Riemannian sparse codingusing geodesic distance, the Jeffrey, Stein, or projection metrics.

R-LLC$_{G/J/S/P}$: Riemannian LLC coding using geodesic distance, the Jeffrey, Stein, or projection metrics.

R-CC$_{G/J/S/P}$: Riemannian CC coding using geodesic distance, the Jeffrey, Stein, or projection metrics.

A. The Manifold of SPD Matrices

Here, an image is described by a set of local RCovDs. In particular, having a block $I(x, y)$ of size $W \times H$ at the disposal, let $\mathbb{O} = \{\boldsymbol{o}_i\}_{i=1}^r$, $\boldsymbol{o}_i \in \mathbb{R}^d$, be a set of r observations from $I(x, y)$. Then, using Eq. (2), the block can be described by a $d \times d$ RCovD. We use a simple Nearest Neighbor (NN) classifier to classify the final descriptors (such as BoW or R-VLAD). This clearly shows the benefits of our proposal.

1) Head Pose Classification: We consider the task of head pose (orientation) classification utilizing two datasets, namely Heads Of Coffee break (HOCoffee) and Queen Mary University of London (QMUL) datasets [35]. The HOCoffee dataset is gathered for the purpose of autonomously detecting social interactions and presents 18,117 outdoor images in low-resolution, taken by a head detector . The QMUL head dataset has 19,292 images, captured in an airport terminal. Images of both datasets are of size 50×50 pixels and split into a predefined training/test partition. There are 9,522 training and 8,595 test images in the HOCoffee dataset spanning 6 different orientations: front, front-right, front-left, right, left and back. For the QMUL dataset, 10,517 images are used for training and the remaining 8,775 images are considered for testing. The images are uniformly partitioned into 5 classes: front, right, left, back and background. The classification task is quite challenging since the datasets feature non-homogeneous illumination and severe occlusions.

As for image descriptor, following [35], we utilized a Difference Of Offset Gaussian (DOOG) filter-bank along color and image gradients for both datasets. The corresponding feature vector at image pixel (x, y) is

$$\boldsymbol{o}_{x,y} = \Big[I_L(x,y), I_a(x,y), I_b(x,y), \sqrt{I_x^2 + I_y^2}, \\ arctan\Big(\frac{|I_x|}{|I_y|}\Big), G_1(x,y), \cdots, G_8(x,y)\Big]\,,$$

where $I_c(x, y)$, $c \in \{L, a, b\}$, is the CIELab color information, I_x and I_y denote luminance derivatives, and $G_i(x, y)$ shows the response of the i-th DOOG centered at $I_L(x, y)$. Thereby, RCovDs are on $\mathcal{S}_{++}^{13}$.

The first column of Table II reports recognition accuracy numbers of all the mentioned methods on the HOCoffee dataset. Several conclusions can be drawn here. Firstly, even the simple **R-BOW**$_G$ is superior to the previous state-of-the-art algorithm, demonstrating the advantageous of local approaches. Compared to sparse coding techniques, R-VLAD coding with all the metrics achieve higher performances. **HO-R-VLAD**$_G$ is the overall winner when the classification accuracy is considered. However, the correct classification numbers of **HO-R-VLAD**$_S$ and **HO-R-VLAD**$_J$ are on par or slightly worse than that of the **HO-R-VLAD**$_G$'s value, while being at least 65 times faster in the training stage and 27 times faster in coding phase (especially for the Jeffrey metric). In comparison to the Log-Euclidean methods, we observe that the proposed R-VLAD method is considerably superior,

TABLE II: Classification accuracy numbers in % for the HOCoffee and QMUL datasets [35].

Method	HOCoffee	QMUL
WARCO	80.8 [35]	91.2 [35]
BOW$_{LE}$	81.6	87.2
R-BOW$_{G}$	81.8	87.6
VLAD$_{LE}$	82.4	87.8
R-SC$_{G}$	83.2	91.7
R-SC$_{S}$	83.1	91.6
R-SC$_{J}$	82.9	91.2
R-LLC$_{G}$	84.0	92.1
R-LLC$_{S}$	83.8	91.7
R-LLC$_{J}$	83.7	91.5
R-CC$_{G}$	82.7	90.6
R-CC$_{S}$	82.6	90.5
R-CC$_{J}$	82.4	90.1
R-VLAD$_{G}$	85.0	92.5
R-VLAD$_{S}$	84.9	92.5
R-VLAD$_{J}$	84.5	92.2
HO-R-VLAD$_{G}$	**85.3**	**92.9**
HO-R-VLAD$_{S}$	85.0	92.7
HO-R-VLAD$_{J}$	84.7	92.5

suggesting the fact that the underlying Riemannian structure is better exploited in R-VLAD.

Among sparse coding family methods, R-LLC using the geodesic distance obtains the highest accuracy. Here, collaborative construction of codes using all codebook atoms as in the variants of R-CC yields slightly inferior recognition accuracy. However, the accuracy numbers are still greater than the state-of-the-art methods.

The second column of Table II, reports recognition accuracies of all the considered methods for the QMUL dataset. Similar to our previous experiment, regardless of the metric, the performance is improved by considering the higher order information. The **HO-R-VLAD**$_{G}$ achieves the highest classification accuracy which is about 1.7 percentage points greater than [35]. Moreover, the **R-VLAD**$_{S}$ works on par with the **R-VLAD**$_{G}$ while both are the preferred techniques to the **R-VLAD**$_{J}$ in terms of classification accuracy. Among the sparse coding family methods, the highest accuracy is obtained when the geodesic distance is used while sparse coding with the Jeffrey divergence yields the lowest accuracy number.

Furthermore, we tested the performance of the conventional VLAD (**VLAD**$_{E}$) by varying codebook sizes to obtain descriptors with the same size or greater than that of R-VLAD's descriptors. We observed that our R-VLAD methods comfortably outperform **VLAD**$_{E}$. For example, the highest accuracy number of **VLAD**$_{E}$ on the HOCoffee and QMUL datasets are 79.9% and 85.7%, respectively.

B. Grassmannian Manifold

As our empirical evaluation on Grassmannian manifolds, we choose the application of recognition from videos by image-set modeling of the videos to create Grassmannian points. Similar to the experiments on the SPD manifolds, local descriptors of numerous small spatio-temporal blocks of a video are extracted. Then each cube is described by a linear subspace using SVD. We use a linear SVM classifier to further improve performances.

1) Dynamic Texture Classification: For our first experiment on the Grassmann manifold, we tackled the problem of dynamic texture classification. To this end, we used the Dyntex++ dataset [12] whose samples contain certain stationarity properties in time domain. DynTex++ has 3600 ($50 \times 50 \times 50$) videos of moving scenes, spanning 36 categories.

To extract local Grassmannian points, videos were first split into cubes of size $15 \times 15 \times 15$ with 5 pixels/frames overlap in spatial/temporal axis. Then, the cubes were described by the Local Binary Pattern in Three Orthogonal Planes (LBP-TOP) [43]. Finally, from the extracted features, we generated subspaces of order 6 using SVD. Therefore, our local descriptors belonged to $\mathcal{G}^6_{177}$. In total, 512 Grassmannian points were obtained from each dynamic texture video.

Here, we used the experimental set-up adopted in [3]. More specifically, videos of each category were randomly divided into training and testing sets with equal number of videos within each set. This process was repeated 10 times. Table III reports average accuracy numbers along standard deviations of all the considered methods.

Table III shows that the **R-VLAD**$_{P}$ outperforms the accuracy number of the previous best method of [3] by more than 5 percentage points. Similar to the previous experiments, again R-VLAD outperforms the VLAD using the Log-Euclidean solution. However, the Log-Euclidean VLAD performs better than the method of [3]. Moreover, HO-R-VLAD boosts the accuracy values when the geodesic or projection metrics are utilized as similarity measures. HO-R-VLAD with projection metric, the **HO-R-VLAD**$_{P}$, achieves the highest average recognition rate of 97.8%.

2) Dynamic Scene Categorization: We conducted another experiment to classify videos of dynamic scenes (similar to the dynamic texture videos) utilizing the Maryland "In-The-Wild" dataset [30]. The dataset is very challenging due to the web nature of videos, having significant camera motions, scene cuts, differences in appearance, scale, frame rate, illumination conditions and viewpoint. The videos span 13 classes (*e. g.* , Avalanche) with 10 videos in each category.

Following the standard setup used in [11], we utilized the FC7 features of the CNN of Zhou *et al.* [45] trained on the Places dataset [45] which has 205 scene classes and 2,5 millions number of images. The utilized features are 4096 dimension which we subsequently reduce them to 400. To extract local Grassmannian points, we grouped every 6 consecutive frames of the videos with 90% overlap and generated the linear subspaces. As such each local descriptor belongs to $\mathcal{G}^6_{400}$. A leave-one-video-out validation protocol is considered for consistency with previous study in [11].

The recognition accuracies for all the studied algorithms are summarized in the third column of Table III. To the best of our knowledge, the recent work of faraki *et al.* [11] has achieved the highest accuracy on this dataset. Our HO-R-VLAD using the grassmannian geodesic metric outperforms this state-of-the-art by 1.5%. Furthermore, the HO-R-VLAD using the projection metric obtains the top accuracy number,

TABLE III: Recognition accuracies in % for the Maryland [30], Dyntex++ [12] and YTC [23] datasets.

Method	Dyntex++	Maryland	YTC
Previous Best	92.4 [3]	90.0 [11]	72.6 ± 5.1 [20]
BOW$_{LE}$	81.1 ± 0.5	84.6	55.3 ± 2.9
R-BOW$_{G}$	92.4 ± 0.5	85.4	64.5 ± 5.1
VLAD$_{LE}$	93.3 ± 0.4	86.9	65.2 ± 2.8
R-SC$_{G}$	96.0 ± 0.4	87.7	74.1 ± 3.0
R-SC$_{P}$	96.1 ± 0.2	88.5	74.8 ± 5.2
R-LLC$_{G}$	96.3 ± 0.3	88.5	75.7 ± 3.2
R-LLC$_{P}$	96.3 ± 0.2	90.0	76.7 ± 4.8
R-CC$_{G}$	95.4 ± 0.5	86.9	75.2 ± 2.9
R-CC$_{P}$	95.8 ± 0.4	87.7	75.4 ± 2.0
R-VLAD$_{G}$	96.7 ± 0.3	90.0	75.6 ± 2.5
R-VLAD$_{P}$	97.6 ± 0.4	90.8	**79.9 ± 3.6**
HO-R-VLAD$_{G}$	97.0 ± 0.7	91.5	78.7 ± 3.8
HO-R-VLAD$_{P}$	**97.8 ± 0.4**	**93.1**	79.8 ± 3.7

outperforming [11] by more than 3 percentage points. Notably, HO-R-VLAD is superior to the R-VLAD using both metrics. Compared to the Log-Euclidean solution, R-VLAD is preferable, indicating the advantage of our proposal.

3) Face Recognition from Videos: For our last experiment, we tackled the problem of face recognition from videos. For this task, we examined the YouTube Celebrity (YTC) dataset [23] which comes with 1910 videos of 47 subjects. The dataset is very challenging due to high variation of poses, illumination conditions and facial expressions along with high compression ratio of the images. For our evaluation, we followed the widely used setup in Deep Reconstruction Models (DRM) [20]. Specifically, face areas are first extracted from all frames of the videos. This is followed by dividing each face region into distinct blocks and extracting the histogram of Local Binary Patterns (LBP) [25]. Concatenated LBP features form the final descriptor.

As for the evaluation protocol, different protocols exist for the YTC. Here again, we followed the cross validation protocol adopted in [20]. More specifically, all videos are divided into 5 folds (equally and with minimum overlap) where each fold has 9 videos (with 3 and 6 randomly chosen videos for training and testing, respectively). Linear subspaces of order 6 constitute our descriptors.

The last column of Table III shows the mean correct recognition rates along with the standard deviations of all the methods. The results are self-explanatory. The **R-VLAD**$_{G}$ and **R-VLAD**$_{P}$ comfortably outperform the DRM method. Furthermore, the accuracy gap between the Log-Euclidean solution and R-VLAD exceeds 10 percentage points. While encoding higher-order information improves the accuracy of **R-VLAD**$_{G}$, the highest accuracy number of 79.9% is obtained by R-VLAD when the projection metric is utilized.

Here, R-LLC coding is still the preferred coding method among the sparse coding schemes. Furthermore, collaborative coding improves the performance over simple sparse coding with both metrics, indicating this type of coding is more useful -as originally devised- for face recognition task.

Computational Load: To give the reader a better picture on the computational load of our coding methods, we recorded the average coding times for 100 descriptors on $\mathcal{S}_{++}^{13}$ and $\mathcal{G}_{177}^{6}$. These are indeed examples of the SPD and Grassmann manifolds which we had in our experiments. Table IV shows the recorded times using Matlab on a quad-core computer, when different metrics are used in our coding methods. Since the extra computational load in **HO-R-VLAD** (over **R-VLAD**) is negligible, we removed that coding from the table.

VII. Conclusions

In this paper, we studied Riemannian coding/aggregating methods such as VLAD [22] and sparse coding [40]. In doing so, we were motivated by the favorable outcomes of coding/aggregating methods in conventional Euclidean spaces and excellent discriminatory power of visual descriptors on Riemannian manifolds. Particularly, we considered local RCovDs and linear subspaces extracted from images and videos, as structured points on the manifold of SPD matrices and the Grassmann manifolds, respectively. Aside from an extensive formulation, we developed a family of methods that benefits from various forms of similarity measures defined on the underlying manifolds. A comprehensive set of empirical evaluation on various challenging computer vision problems supported our proposal.

References

[1] A. L. Alexander, J. E. Lee, M. Lazar, and A. S. Field. Diffusion tensor imaging of the brain. *Neurotherapeutics*, 4(3):316–329, 2007. 1

[2] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 29(1):328–347, 2007. 1, 5, 8

[3] M. Baktashmotlagh, M. Harandi, B. C. Lovell, and M. Salzmann. Discriminative non-linear stationary subspace analysis for video classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(12):2353 – 2366, 2014. 9, 10

[4] E. Begelfor and M. Werman. Affine invariance revisited. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2087–2094, 2006. 6

[5] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos. 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, 2012. 5, 6, 8

[6] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recognition and segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3828–3836, 2015. 1, 4

[7] X. Deng, F. Da, and H. Shao. Efficient 3d face recognition using local covariance descriptor and riemannian kernel sparse coding. *Computers & Electrical Engineering*, 2017. 2

[8] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. 20(2):303–353, 1998. 1, 3

[9] M. Faraki, M. T. Harandi, and F. Porikli. Material classification on symmetric positive definite manifolds. In *Proc. IEEE Winter Conference on Applications of Computer Vision*, pages 749–756, 2015. 1

[10] M. Faraki, M. T. Harandi, and F. Porikli. More about vlad: A leap from euclidean to riemannian manifolds. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4951–4960. IEEE, 2015. 1, 8

[11] M. Faraki, M. T. Harandi, and F. Porikli. Image set classification by symmetric positive semi-definite matrices. In *Proc. IEEE Winter Conference on Applications of Computer Vision*, pages 749–756, 2016. 1, 9, 10

[12] B. Ghanem and N. Ahuja. Maximum margin distance learning for dynamic texture recognition. In *Proc. European Conference on Computer Vision*, pages 223–236, 2010. 9, 10

[13] A. Goh and R. Vidal. Clustering and dimensionality reduction on Riemannian manifolds. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE, 2008. 5

[14] G. H. Golub and C. F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. 6

TABLE IV: Computational loads of our coding methods in $\mathcal{S}_{++}^{13}$ (first row) and in $\mathcal{G}_{177}^{6}$ (second row) in seconds.

R-SC$_G$	**R-SC**$_S$	**R-SC**$_J$	**R-LLC**$_G$	**R-LLC**$_S$	**R-LLC**$_J$	**R-CC**$_G$	**R-CC**$_S$	**R-CC**$_J$	**R-VLAD**$_G$	**R-VLAD**$_S$	**R-VLAD**$_J$
7.1	5.9	6.3	6.8	5.0	6.1	7.0	5.2	6.2	2.1	0.4	0.1
R-SC$_G$	**R-SC**$_P$		**R-LLC**$_G$	**R-LLC**$_P$		**R-CC**$_G$	**R-CC**$_P$		**R-VLAD**$_G$	**R-VLAD**$_P$	
31.6	25.6		25.8	19.6		27.8	21.3		0.3	0.2	

[15] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *Proc. European Conference on Computer Vision*, pages 392–407. Springer, 2014. 4

[16] J. Hamm and D. D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proc. Int. Conference on Machine Learning*, pages 376–383. ACM, 2008. 2, 3, 5

[17] M. Harandi, R. Hartley, C. Shen, B. Lovell, and C. Sanderson. Extrinsic methods for coding and dictionary learning on grassmann manifolds. *Int. Journal of Computer Vision*, 114(2-3):113–136, 2015. 1, 2, 3

[18] M. T. Harandi, R. Hartley, B. Lovell, and C. Sanderson. Sparse coding on symmetric positive definite manifolds using bregman divergences. *IEEE transactions on neural networks and learning systems*, 27(6):1294–1306, 2016. 1, 2

[19] W. Hariri, H. Tabia, N. Farah, A. Benouareth, and D. Declercq. 3d facial expression recognition using kernel methods on riemannian manifold. *Engineering Applications of Artificial Intelligence*, 64:25–32, 2017. 1

[20] M. Hayat, M. Bennamoun, and S. An. Deep reconstruction models for image set classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(4):713–727, 2015. 10

[21] J. Ho, Y. Xie, and B. Vemuri. On a nonlinear generalization of sparse coding and dictionary learning. In *Proc. Int. Conference on Machine Learning*, pages 1480–1488, 2013. 5

[22] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2012. 1, 2, 4, 5, 10

[23] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 10

[24] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178. IEEE, 2006. 1, 2

[25] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002. 10

[26] X. Peng, L. Wang, Y. Qiao, and Q. Peng. Boosting vlad with supervised dictionary learning and high-order statistics. In *Proc. European Conference on Computer Vision*, volume 8691, pages 660–674. Springer, 2014. 1, 2, 6

[27] X. Pennec, P. Fillard, and N. Ayache. A Riemannian framework for tensor computing. *Int. Journal of Computer Vision*, 66(1):41–66, 2006. 1, 2, 3, 5, 7

[28] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 1, 2, 3

[29] L. K. Saul and S. T. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003. 7

[30] N. Shroff, P. Turaga, and R. Chellappa. Moving vistas: Exploiting motion for describing scenes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1911–1918. IEEE, 2010. 9, 10

[31] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. Int. Conference on Computer Vision*, pages 1470–1477. IEEE, 2003. 1, 2, 4

[32] S. Sra. A new metric on the manifold of kernel matrices with application to matrix geometric means. In *Proc. Advances in Neural Information Processing Systems*, pages 144–152, 2012. 2, 3

[33] S. Sra and R. Hosseini. Conic geometric optimisation on the manifold of positive definite matrices. *arXiv:1312.1039*, 2014. 6

[34] R. Subbarao and P. Meer. Nonlinear mean shift over Riemannian manifolds. *Int. Journal of Computer Vision*, 84(1):1–20, 2009. 5

[35] D. Tosato, M. Spera, M. Cristani, and V. Murino. Characterizing humans on Riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1972–1984, 2013. 1, 8, 9

[36] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on Riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1713–1727, 2008. 1

[37] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3360–3367. IEEE, 2010. 1, 4, 7

[38] R. Wang, H. Guo, L. S. Davis, and Q. Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2496–2503. IEEE, 2012. 1

[39] Z. Wang and B. C. Vemuri. An affine invariant tensor dissimilarity measure and its applications to tensor-valued image segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 223–228. IEEE, 2004. 2, 3, 5

[40] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009. 1, 2, 4, 7, 10

[41] Y. Xie, J. Ho, and B. Vemuri. On a nonlinear generalization of sparse coding and dictionary learning. In *Proc. Int. Conference on Machine Learning*, page 1480. NIH Public Access, 2013. 2, 7

[42] L. Zhang, M. Yang, and X. Feng. Sparse representation or collaborative representation: Which helps face recognition? In *Proc. Int. Conference on Computer Vision*, pages 471–478. IEEE, 2011. 2, 4

[43] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928, 2007. 9

[44] N. Zhong, W. Yang, A. Cherian, X. Yang, G.-S. Xia, and M. Liao. Unsupervised classification of polarimetric sar images via riemannian sparse coding. *IEEE Transactions on Geoscience and Remote Sensing*, 55(9):5381–5390, 2017. 2

[45] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Proc. Advances in Neural Information Processing Systems*, pages 487–495, 2014. 9

Masoud Faraki is a Postdoctoral Research Fellow at the Australian Centre for Robotic Vision (ACRV) located at Monash University. Earlier, he was a PhD candidate at the Australian National University (ANU) and at Data61 located at CSIRO, Canberra, Australia. His main research interests include machine learning and computer vision. He also holds a M.Sc. degree in artificial intelligence and a B.Sc. in computer software engineering.

Mehrtash Harandi is a senior research scientist at Machine Learning Research Group (MLRG), Data61 and an adjunct lecturer at Australian National University (ANU), Canberra, Australia. His main research interests are theoretical and computational methods in machine learning and computer vision with a focus on Riemannian geometry.

Fatih Porikli is an IEEE Fellow and a Professor at the Australian National University. He is also acting as the Chief Scientist at Huawei, Santa Clara. He has received his Ph.D. from New York University. Previously he served as the Computer Vision Research Group Leader at NICTA and Distinguished Scientist at MERL. His research interests include computer vision and machine learning with commercial applications in autonomous vehicles, video surveillance, visual inspection, robotics, and medical systems. He received the R&D100 Scientist of the Year Award in 2006, won 5 Best Paper awards at IEEE conferences, and invented 77 patents. He authored more than 200 publications and co-edited 2 books. He is serving as the Associate Editor of 6 journals for the past 10 years.