Covariance matrices encoding based on the log-Euclidean and affine invariant Riemannian metrics

Ioana Ilea, Lionel Bombrun, Salem Said, Yannick Berthoumieu

Covariance matrices encoding based on the log-Euclidean and affine invariant Riemannian metrics

Ioana Ilea
Technical University of Cluj-Napoca
ioana.ilea@com.utcluj.ro

Lionel Bombrun, Salem Said and Yannick Berthoumieu
Laboratoire IMS, Université de Bordeaux
firstname.lastname@ims-bordeaux.fr

Abstract

This paper presents coding methods used to encode a set of covariance matrices. Starting from a Gaussian mixture model adapted to the log-Euclidean or affine invariant Riemannian metric, we propose a Fisher Vector (FV) descriptor adapted to each of these metrics: the log Euclidean FV (LE FV) and the Riemannian Fisher Vector (RFV). An experiment is conducted on four conventional texture databases to compare these two metrics and to illustrate the potential of these FV based descriptors compared to state-of-the-art BoW and VLAD based descriptors. A focus is also done to illustrate the advantage of using the Fisher information matrix during the derivation of the FV.

1. Introduction

Feature coding based approaches are used in wide variety of signal, image and video processing applications including text retrieval [36], image classification [9, 37, 27], action and face recognition [11] to cite a few of them. Among the most employed coding based methods, there are the bag of words model (BoW), the vector of locally aggregated descriptors (VLAD) and the Fisher vectors (FV). These approaches have first been introduced for applications involving non-parametric features. Later, they have been extended to features defined on a Riemannian manifold. This is the case of covariance matrices that have already demonstrated their importance as descriptors related to array processing [26], radar detection [16, 5, 41, 4], image segmentation [15, 8], face detection [30], vehicle detection [24], or classification [14, 3, 32, 11], etc. Nevertheless, since covariance matrices are non-negative definite matrices, conventional tools developed in the Euclidean space cannot be applied directly. The characteristics of the Riemannian geometry of the space $\mathcal{P}_m$ of $m \times m$ symmetric and positive definite (SPD) matrices should be considered in order to obtain appropriate algorithms. The aim of this paper is to introduce a unified framework for BoW, VLAD and

FV approaches, for features being covariance matrices. In the recent literature, some authors have proposed to extend the BoW and VLAD descriptors to the log-Euclidean and affine invariant Riemannian metrics. This yields to the so-called log-Euclidean bag of words (LE BoW) [42, 13], bag of Riemannian words (BoRW) [12], log-Euclidean vector of locally aggregated descriptors (LE VLAD) [11], extrinsic vector of locally aggregated descriptors (E-VLAD) [10] and intrinsic Riemannian vector of locally aggregated descriptors (RVLAD) [11]. For the FV descriptor, based on the Riemannian Gaussian distribution [33], Ilea *et al.* have introduced in [18] the Riemannian Fisher score for texture image classification. However, they do not take into account the Fisher information which is one of the main objective of the paper.

In this paper, we provide two main contributions. The first one is to present how FV can be used to encode a set of covariance matrices. Since these elements do not lie on an Euclidean space but on a Riemannian manifold, a Riemannian metric should be considered. Here, two Riemannian metrics are considered : the log-Euclidean and the affine invariant Riemannian metrics yielding respectively to the log-Euclidean FV (LE FV) and to the Riemannian FV (RFV). The second main contribution is to highlight the impact of the Fisher information matrix (FIM) in the derivation of the FV.

The paper is structured as follows. Section 2 presents the codebook generation on the manifold of SPD covariance matrices. Section 3 introduces a theoretical study of the feature encoding methods (BoW, VLAD and FV) based on the log-Euclidean and affine invariant Riemannian metrics. Section 4 shows an application of these descriptors to texture image classification. And finally, some conclusions are drawn in Section 5.

2. Codebook generation in $\mathcal{P}_m$

This section focuses on the codebook generation. At this point, the set of extracted low-level features, *i.e.* the set of covariance matrices, is used in order to identify the ones

embedding the set's significant characteristics. In this paper, two metrics are considered to compute the codebook which are respectively the log-Euclidean (LE) and the affine invariant Riemannian metric. In the following, each method with prefix LE (resp. R) means that it is based on the log-Euclidean (resp. the affine invariant) Riemannian metric. The next two subsections describe these two strategies.

2.1. Log-Euclidean codebook

Let $\mathcal{M} = \{\mathbf{M}_n\}_{n=1:N}$, with $\mathbf{M}_n \in \mathcal{P}_m$, be a sample of N training SPD matrices of size $m \times m$. The log-Euclidean codebook is obtained by considering the log-Euclidean metric as similarity measure between two covariance matrices. For such a purpose, each training covariance matrix $\mathbf{M}_n$ is first mapped on the LE space by applying the matrix logarithm $\mathbf{M}_n^{LE} = \log \mathbf{M}_n$ [2, 13, 31]. Next, a vectorization operator is applied to obtain the LE vector representation. To sum up, for a given SPD matrix $\mathbf{M}$, its LE vector representation, $\mathbf{m} \in \mathbb{R}^{\frac{m(m+1)}{2}}$, is defined as $\mathbf{m} = \mathrm{Vec}(\log(\mathbf{M}))$ where Vec is the vectorization operator defined as:

$$\mathrm{Vec}(\mathbf{X}) = \left[X_{11}, \sqrt{2}X_{12}, \ldots, \sqrt{2}X_{1m}, X_{22}, \sqrt{2}X_{23}, \ldots, X_{mm} \right], \quad (1)$$

with X_{ij} the elements of $\mathbf{X}$.

Once the SPD matrices are mapped on the LE metric space, all the conventional algorithms developed on the Euclidean space can be considered. In particular, the LE vector representation of $\mathcal{M}$, i.e. $\{\mathbf{m}_n\}_{n=1:N}$, can be assumed to be independent and identically distributed samples from a mixture of K multivariate Gaussian distribution, whose probability density function is

$$p(\mathbf{m}_n|\theta) = \sum_{k=1}^{K} \varpi_k p(\mathbf{m}_n|\bar{\mathbf{m}}_k, \Sigma_k), \quad (2)$$

where $\theta = \{(\varpi_k, \bar{\mathbf{m}}_k, \Sigma_k)_{1 \le k \le K}\}$ is the parameter vector. For each cluster k, ϖ_k represent the mixture weight, $\bar{\mathbf{m}}_k$ the mean vector and Σ_k the covariance matrices. It yields:

$$p(\mathbf{m}|\theta_k) = \frac{1}{(2\pi)^{\frac{m}{2}} |\Sigma_k|^{\frac{1}{2}}} \exp \left\{ -\tfrac{1}{2}(\mathbf{m} - \bar{\mathbf{m}}_k)^T \Sigma_k^{-1}(\mathbf{m} - \bar{\mathbf{m}}_k) \right\}. \quad (3)$$

where $(\cdot)^T$ is the transpose operator, $\bar{\mathbf{m}}_k \in \mathbb{R}^{\frac{m(m+1)}{2}}$, $\Sigma_k \in \mathcal{P}_m$ and $\varpi_k \in \mathbb{R}$. In addition, the covariance matrix is assumed to be diagonal, i.e. $\sigma_k^2 = \mathrm{diag}(\Sigma_k)$ is the variance vector. For such a model, the classical k-means or expectation-maximization (EM) algorithm can be applied to estimate the mixture parameters. The estimated parameters of each mixture component ($\bar{\mathbf{m}}_k$, σ_k^2 and ϖ_k) represent the codewords and the set composed by the K codewords gives the log-Euclidean codebook.

2.2. Riemannian codebook

In this section, we present the construction of the Riemannian codebook which is based on the affine invariant

Riemannian metric. For that, we first introduce the Riemannian Gaussian model.

Riemannian Gaussian model In order to model the space $\mathcal{P}_m$ of SPD covariance matrices, a generative model has been introduced in [33, 34]: the Riemannian Gaussian distribution (RGD). For this model, the probability density function with respect to the Riemannian volume element is defined as follow [33, 34]:

$$p(\mathbf{M}_n|\bar{\mathbf{M}}, \sigma) = \frac{1}{Z(\sigma)} \exp \left\{ -\frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}})}{2\sigma^2} \right\}, \quad (4)$$

where $\bar{\mathbf{M}}$ and σ are the distribution parameters, representing respectively the central value and the dispersion. $d(\cdot)$ is the Riemannian distance given by $d(\mathbf{M}_1, \mathbf{M}_2) = \left[\sum_i (\ln \lambda_i)^2 \right]^{\frac{1}{2}}$, with λ_i, $i = 1 \ldots m$ being the eigenvalues of $\mathbf{M}_1^{-1} \mathbf{M}_2$. $Z(\sigma)$ is a normalization factor independent of $\bar{\mathbf{M}}$ [6]:

$$Z(\sigma) = \frac{8^{\frac{m(m-1)}{4}} \pi^{m^2/2}}{m! \, \Gamma_m(m/2)} \int_{\mathbb{R}^m} e^{-\frac{\|\mathbf{r}\|^2}{2\sigma^2}} \prod_{i<j} \sinh \left(\frac{|r_i - r_j|}{2} \right) \prod_{i=1}^m \mathrm{d}r_i \quad (5)$$

with Γ_m the multivariate Gamma function [25]. In practice, for $m = 2$, the normalization factor admits a closed-form expression [32], while for $m > 2$ the normalization factor can be computed numerically as the expectation of the product of $\sinh$ functions with respect to the multivariate normal distribution $\mathcal{N}(0, \sigma^2 I_m)$ [33]. Afterwards, a cubic spline interpolation can be used to smooth this function [43].

Mixture model for RGDs As for the log Euclidean codebook, a generative model is considered for the construction of the Riemannian codebook. For the former, a mixture of multivariate Gaussian distribution was considered since the SPD matrices were projected on the LE space. For the construction of the Riemannian codebook, we follow a similar approach by considering that $\mathcal{M} = \{\mathbf{M}_n\}_{n=1:N}$, are independent and identically distributed samples of a mixture of K RGDs. In this case, the likelihood of $\mathcal{M}$ is given by:

$$p(\mathcal{M}|\theta) = \prod_{n=1}^N p(\mathbf{M}_n|\theta) = \prod_{n=1}^N \sum_{k=1}^K \varpi_k p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k), \quad (6)$$

where $p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k)$ is the RGD defined in (4) and $\theta = \{(\varpi_k, \bar{\mathbf{M}}_k, \sigma_k)_{1 \le k \le K}\}$ is the parameter vector containing the mixture weight ϖ_k, the central value $\bar{\mathbf{M}}_k$ and the dispersion parameter σ_k.

Once estimated, the parameters of each mixture component represent the codewords, and the set of all K codewords gives the Riemannian codebook. Regarding the estimation, the conventional intrinsic k-means clustering algorithm can be considered [39, 12]. Nevertheless, it as-

sumes the homoscedasticity assumption, for which the clusters have the same dispersion. To relax this assumption, we consider in the following the maximum likelihood estimation with the expectation maximization algorithm defined in [32].

Based on the extracted (log-Euclidean or Riemannian) codebook, the next section presents various strategies to encode a set of SPD matrices. These approaches are based whether on the LE metric or on the affine invariant Riemannian metric. In the next section, three kinds of coding approaches are reviewed, namely the bag of words (BoW) model, the vector of locally agregated descriptors (VLAD) [21, 1] and the Fisher vectors (FV) [27, 29, 28]. Here, the main contribution is the proposition of coding approaches based on the FV model: the log-Euclidean Fisher vectors (LE FV) and the Riemannian Fisher vectors (RFV) [18].

3. Feature encoding methods

Given the extracted codebook, the purpose of this part is to project the feature set of SPD matrices onto the codebook elements. In other words, the initial feature set is expressed using the codewords contained in the codebook. Fig. 1 draws an overview of the relation between the different approaches based on the BoW, VLAD and FV models. The log-Euclidean (resp. the affine invariant) Riemannian-based metric approaches appear in red (resp. in blue). The E-VLAD descriptor is displayed in purple since it considers the Riemannian codebook combined with LE representation of the features.

3.1. Bag of words descriptor

One of the most common encoding methods is represented by the BoW model. With this model, a set of features is encoded in an histogram descriptor obtained by counting the number of features which are closest to each codeword of the codebook. In the beginning, this descriptor has been employed for text retrieval and categorization [36, 22], by modeling a text with an histogram containing the number of occurrences of each word. Later on, the BoW model has been extended to visual categorization [7], where images are described by a set of descriptors, such as SIFT features. In such case, the "words" of the codebook are obtained by considering a clustering algorithm with the standard Euclidean metric. Recently, the BoW model has been extended to features lying in a non-Euclidean space, such as SPD matrices. In this context, two approaches have been proposed based respectively on the log-Euclidean and affine invariant Riemannian metrics:

- the log-Euclidean bag of words (LE BoW) [42, 13].

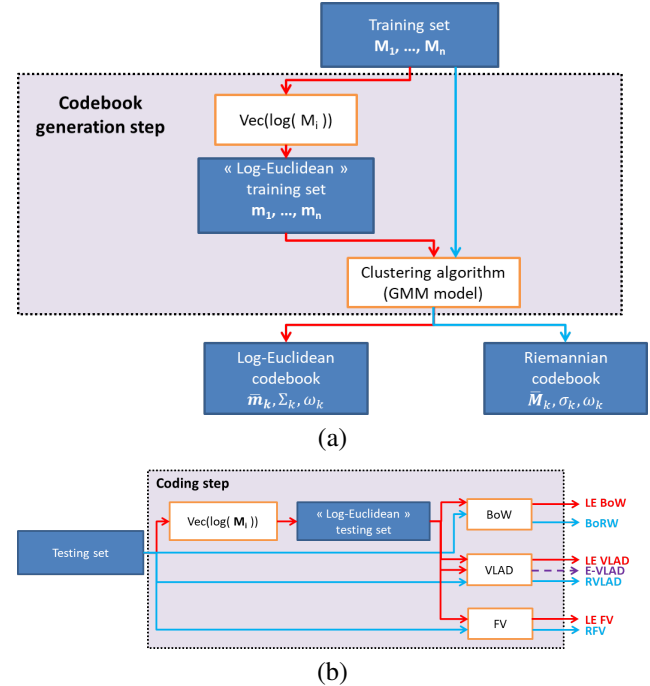- the bag of Riemannian words (BoRW) [12].

(a)

(b)

Figure 1. Workflow explaining (a) the codebook creation step and (b) the coding step. The log-Euclidean based approaches appear in red while the Riemannian based ones are displayed in blue. The E-VLAD descriptor is displayed in purple since it considers simultaneously a Riemannian codebook and LE vector representation of the covariance matrices.

These two descriptors have been employed successfully for different applications, including texture and human epithelial type 2 cells classification [12], action recognition [42, 13].

3.1.1 Log-Euclidean bag of words (LE BoW)

The LE BoW model has been considered in [42, 13]. First, the space of covariance matrices is embedded into a vector space by considering the LE vector representation $\mathbf{m}$ given in (1). With this embedding, the LE BoW model can be interpreted as the BoW model in the LE space. This means that codewords are elements of the log-Euclidean codebook detailed in Section 2.1. Next, each observed SPD matrix $\mathbf{M}_n$ is assigned to cluster k of closest codeword $\bar{\mathbf{m}}_k$ to compute the histogram descriptor. The vicinity is evaluated here as the Euclidean distance between the LE vector representation $\mathbf{m}_n$ and the codeword $\bar{\mathbf{m}}_k$.

The LE BoW descriptor can also be interpreted by considering the Gaussian mixture model recalled in (2). In such case, each feature $\mathbf{m}_n$ is assigned to the cluster k, for $k = 1, \ldots, K$ according to:

$$\arg\max_k \varpi_k \, p(\mathbf{m}_n | \bar{\mathbf{m}}_k, \sigma_k), \qquad (7)$$

where $p(\mathbf{m}_n|\bar{\mathbf{m}}_k, \sigma_k)$ is the multivariate Gaussian distribution given in (3). In addition, two constraints are assumed $\forall k = 1, \ldots, K$: the homoscedasticity assumption, *i.e.* $\sigma_k = \sigma$ and the same weight is given to all mixture components, *i.e.* $\varpi_k = \frac{1}{K}$.

3.1.2 Bag of Riemannian words (BoRW)

This descriptor has been introduced in [12]. Contrary to the LE BoW model, the BoRW model exploits the affine invariant Riemannian metric. For that, it considers the Riemannian codebook detailed in Section 2.2. Then, the histogram descriptor is computed by assigning each SPD matrix to the cluster k of the closest codebook element $\bar{\mathbf{M}}_k$, the proximity being measured with the geodesic distance recalled in Section 2.2.

As for the LE BoW descriptor, the definition of the BoRW descriptor can be obtained by the Gaussian mixture model, except that the RGD model (4) is considered instead of the multivariate Gaussian distribution. Each feature $\mathbf{M}_n$ is assigned to the cluster k, for $k = 1, \ldots, K$ according to:

$$\arg\max_k \varpi_k \, p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k). \qquad (8)$$

In addition, the two previously cited assumptions are done, the same dispersion and weight are given to all mixture components.

BoW descriptors are based only on the number of occurrences of each codeword from the dataset. In order to increase the classification performances, second order statistics can be considered. This is the case of VLAD and FV that are presented next.

3.2. Vectors of locally aggregated descriptors

VLAD descriptors have been introduced in [21] and represent a method of encoding the difference between the codewords and the features. For features lying in Euclidean space, codebook is composed by cluster centroids $\{(\bar{\mathbf{x}}_k)_{1 \leq k \leq K}\}$ obtained by clustering algorithm on the training set. Next, to encode a feature set $\{(\mathbf{x}_n)_{1 \leq n \leq N}\}$, vectors $\mathbf{v}_k$ containing the sum of differences between codeword and feature sample assigned to it are computed for each cluster:

$$\mathbf{v}_k = \sum_{\mathbf{x}_n \in c_k} \mathbf{x}_n - \bar{\mathbf{x}}_k. \qquad (9)$$

The final VLAD descriptor is obtained as the concatenation of all vectors $\mathbf{v}_k$:

$$\mathbf{VLAD} = [\mathbf{v}_1^T, \ldots, \mathbf{v}_K^T]. \qquad (10)$$

In order to generalize this formalism to features lying in a Riemannian manifold, two theoretical aspects should be addressed carefully, which are the definition of a metric to describe how features are assigned to the codewords, and the definition of subtraction operator for these kind of features. By addressing these aspects, three approaches have been proposed in the literature:

- the log-Euclidean vector of locally aggregated descriptors (LE VLAD) [11].

- the extrinsic vector of locally aggregated descriptors (E-VLAD) [10].

- the intrinsic Riemannian vector of locally aggregated descriptors (RVLAD) [11].

3.2.1 Log-Euclidean vector of locally aggregated descriptors (LE VLAD)

This descriptor has been introduced in [11] to encode a set of SPD matrices with VLAD descriptors. In this approach, VLAD descriptors are computed in the LE space. For this purpose, (9) is rewritten as:

$$\mathbf{v}_k = \sum_{\mathbf{m}_n \in c_k} \mathbf{m}_n - \bar{\mathbf{m}}_k, \qquad (11)$$

where the LE representation $\mathbf{m}_n$ of $\mathbf{M}_n$ belongs to the cluster c_k if it is closer to $\bar{\mathbf{m}}_k$ than any other element of the log Euclidean codebook. The proximity is measured here according to the Euclidean distance between the LE vectors.

3.2.2 Extrinsic vector of locally aggregated descriptors (E-VLAD)

The E-VLAD descriptor is based on the LE vector representation of SPD matrices. However, contrary to the LE VLAD model, this descriptor uses the Riemannian codebook to define the Voronoï regions.

$$\mathbf{v}_k = \sum_{\mathbf{M}_n \in c_k} \mathbf{m}_n - \bar{\mathbf{m}}_k, \qquad (12)$$

where $\mathbf{M}_n$ belongs to the cluster c_k if it is closer to $\bar{\mathbf{M}}_k$ according to the affine invariant Riemannian metric. Note also that here $\bar{\mathbf{m}}_k$ is the LE vector representation of the Riemannian codebook element $\bar{\mathbf{M}}_k$.

In order to speed-up the processing time, Faraki *et al.* have proposed in [10] to replace the Riemannian metric by the Stein metric [38]. For this latter, computational cost to estimate the centroid of a set of covariance matrices is less demanding than with the affine invariant Riemannian metric. A recursive computation of the Stein center from a set of covariance matrices has been proposed in [35]. Since this approach exploits two metrics, one for the codebook creation (with the Riemannian or Stein metric) and another for the coding step (with the log-Euclidean metric), we referred it as an extrinsic method.

3.2.3 Riemannian vector of locally aggregated descriptors (RVLAD)

This descriptor has been introduced in [11] to propose a solution for the affine invariant Riemannian metric. More precisely, the geodesic distance [20] is considered to measure similarity between SPD matrices and the Riemannian logarithm mapping [17] is used to perform the subtraction on the manifold. It yields that for the RVLAD model, the vectors $\mathbf{v}_k$ are obtained as:

$$\mathbf{v}_k = \text{Vec}\left(\sum_{\mathbf{M}_n \in c_k} \text{Log}_{\bar{\mathbf{M}}_k}(\mathbf{M}_n) \right), \qquad (13)$$

where $\text{Log}_{\bar{\mathbf{M}}_k}(\cdot)$ is the Riemannian logarithm mapping. Note that the vectorization operator $\text{Vec}(\cdot)$ is used to represent $\mathbf{v}_k$ as a vector.

As explained in [21], the VLAD descriptor can be interpreted as a simplified non probabilistic version of the FV. In the next section, we give an explicit relationship between these two descriptors which is one of the main contribution of the paper.

3.3. Fisher vector descriptor

Fisher vectors (FV) are descriptors based on Fisher kernels [19]. FV measures how samples are correctly fitted by a given generative model $p(\mathbf{X}|\theta)$. Let $\boldsymbol{\mathcal{X}} = \{\mathbf{x}_n\}_{n=1:N}$, be a sample of N observations. The FV descriptor associated to $\boldsymbol{\mathcal{X}}$ is the gradient of the sample log-likelihood with respect to the parameters θ of the generative model distribution, scaled by the inverse square root of the FIM.

First, the gradient of the log-likelihood with respect to the model parameter vector θ, also known as the Fisher score (FS) $U_{\boldsymbol{\mathcal{X}}}$ [19], should be computed:

$$U_{\boldsymbol{\mathcal{X}}} = \nabla_\theta \log p(\boldsymbol{\mathcal{X}}|\theta) = \nabla_\theta \sum_{n=1}^{N} \log p(\mathbf{X}_n|\theta). \qquad (14)$$

As mentioned in [27], the gradient describes the direction in which parameters should be modified to best fit the data. In other words, the gradient of the log-likelihood with respect to a parameter describes the contribution of that parameter to the generation of a particular feature [19]. A large value of this derivative is equivalent to a large deviation from the model, suggesting that the model does not correctly fit the data.

Second, the gradient of the log-likelihood can be normalized by using the FIM I_θ [19]:

$$I_\theta = E_{\boldsymbol{\mathcal{X}}}[U_{\boldsymbol{\mathcal{X}}} U_{\boldsymbol{\mathcal{X}}}^T], \qquad (15)$$

where $E_{\boldsymbol{\mathcal{X}}}[\cdot]$ denotes the expectation over $p(\boldsymbol{\mathcal{X}}|\theta)$. It yields that the FV representation of $\boldsymbol{\mathcal{X}}$ is given by the normalized gradient vector [27]:

$$\mathscr{G}_\theta^{\boldsymbol{\mathcal{X}}} = I_\theta^{-1/2} \nabla_\theta \log p(\boldsymbol{\mathcal{X}}|\theta). \qquad (16)$$

As reported in previous work, exploiting the FIM I_θ in the derivation of FV yields to excellent results with linear classifiers [28, 29, 37]. However, the computation of FIM might be quite difficult. It does not admit a close-form expression for many generative models. In such case, it can be approximated empirically by carrying out a Monte Carlo integration, but this latter can be costly especially for high dimensional data. To solve this issue, some analytical approximations can be considered [27, 37].

The next part explain how the FV model can be used to encode a set of SPD matrices. Once again, two approaches are considered by using respectively the log-Euclidean and the affine invariant Riemannian metrics:

- the log-Euclidean Fisher vectors (LE FV).

- the Riemannian Fisher vectors (RFV) [18].

3.3.1 Log-Euclidean Fisher vectors

The LE FV model consists in an approach where the FV descriptors are computed in the LE space. In such case, the multivariate Gaussian mixture model recalled in (2) is considered.

Let $\boldsymbol{\mathcal{M}}_{LE} = \{\mathbf{m}_n\}_{n=1:N}$ be the LE representation of the set $\boldsymbol{\mathcal{M}}$. To compute the LE FV representation of $\boldsymbol{\mathcal{M}}$, the derivatives of the log-likelihood function with respect to θ should first be computed. Let $\gamma_k(\mathbf{m}_n)$ be the soft assignment of $\mathbf{m}_n$ to the kth Gaussian component

$$\gamma_k(\mathbf{m}_n) = \frac{\varpi_k \, p(\mathbf{m}_n|\theta_k)}{\sum_{j=1}^{K} \varpi_j \, p(\mathbf{m}_n|\theta_j)}. \qquad (17)$$

It yields that, the elements of the log-Euclidean Fisher score (LE FS) are obtained as:

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}_{LE}|\theta)}{\partial \bar{\mathbf{m}}_k^d} = \sum_{n=1}^{N} \gamma_k(\mathbf{m}_n) \left(\frac{\mathbf{m}_n^d - \bar{\mathbf{m}}_k^d}{\left(\sigma_k^d\right)^2} \right), \qquad (18)$$

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}_{LE}|\theta)}{\partial \sigma_k^d} = \sum_{n=1}^{N} \gamma_k(\mathbf{m}_n) \left(\frac{\left[\mathbf{m}_n^d - \bar{\mathbf{m}}_k^d\right]^2}{\left(\sigma_k^d\right)^3} - \frac{1}{\sigma_k^d} \right), \qquad (19)$$

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}_{LE}|\theta)}{\partial \alpha_k} = \sum_{n=1}^{N} \left(\gamma_k(\mathbf{m}_n) - \varpi_k \right), \qquad (20)$$

where $\bar{\mathbf{m}}_k^d$ (resp. σ_k^d) is the dth element of vector $\bar{\mathbf{m}}_k$ (resp. σ_k). Note that, to ensure the constraints of positivity and sum-to-one for the weights ϖ_k, the derivative of the log-likelihood with respect to this parameter is computed by taking into consideration the soft-max parametrization as proposed in [23, 37]:

$$\varpi_k = \frac{\exp(\alpha_k)}{\sum_{j=1}^{K} \exp(\alpha_j)}. \qquad (21)$$

Under the assumption of nearly hard assignment, that is the soft assignment distribution $\gamma_k(\mathbf{m}_n)$ is sharply peaked on a single value of k for any observation $\mathbf{m}_n$, the FIM I_θ is diagonal and admits a close-form expression [37]. It yields that the LE FV of $\mathcal{M}$ is obtained as:

$$\mathcal{G}^{\mathcal{M}_{LE}}_{\bar{\mathbf{m}}_k^d} = \frac{1}{\sqrt{\varpi_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{m}_n) \left(\frac{\mathbf{m}_n^d - \bar{\mathbf{m}}_k^d}{\sigma_k^d} \right), \qquad (22)$$

$$\mathcal{G}^{\mathcal{M}_{LE}}_{\sigma_k^d} = \frac{1}{\sqrt{2\varpi_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{m}_n) \left(\frac{\left[\mathbf{m}_n^d - \bar{\mathbf{m}}_k^d\right]^2}{\left(\sigma_k^d\right)^2} - 1 \right), \tag{23}$$

$$\mathcal{G}^{\mathcal{M}_{LE}}_{\alpha_k} = \frac{1}{\sqrt{\varpi_k}} \sum_{n=1}^{N} \left(\gamma_k(\mathbf{m}_n) - \varpi_k \right), \qquad (24)$$

3.3.2 Riemannian Fisher vectors

Ilea *et al.* have proposed in [18] an approach to encode a set of SPD matrices with FS based on the affine invariant Riemannian metric: the Riemannian Fisher score (RFS). In this method, the generative model is a mixture of RGDs [33] as presented in Section 2.2. By following the same procedure as before, the RFS is obtained by computing the derivatives of the log-likelihood function with respect to the distribution parameters $\theta = \{(\varpi_k, \bar{\mathbf{M}}_k, \sigma_k)_{1 \leq k \leq K}\}$. It yields that [18]:

$$\frac{\partial \log p(\mathcal{M}|\theta)}{\partial \bar{\mathbf{M}}_k} = \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \frac{\mathrm{Log}_{\bar{\mathbf{M}}_k}(\mathbf{M}_n)}{\sigma_k^2}, \qquad (25)$$

$$\frac{\partial \log p(\mathcal{M}|\theta)}{\partial \sigma_k} = \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \left\{ \frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{\sigma_k^3} - \frac{Z'(\sigma_k)}{Z(\sigma_k)} \right\}, \tag{26}$$

$$\frac{\partial \log p(\mathcal{M}|\theta)}{\partial \alpha_k} = \sum_{n=1}^{N} \left[\gamma_k(\mathbf{M}_n) - \varpi_k \right], \qquad (27)$$

where $\mathrm{Log}_{\bar{\mathbf{M}}_k}(\cdot)$ is the Riemannian logarithm mapping and $Z'(\sigma_k)$ is the derivative of $Z(\sigma_k)$ with respect to σ_k. The function $Z'(\sigma)$ can be computed numerically by a Monte Carlo integration, in a similar way to the one for the normalization factor $Z(\sigma)$ (see Section 2.2).

In these expressions, $\gamma_k(\mathbf{M}_n)$ represents the probability that the feature $\mathbf{M}_n$ is generated by the kth mixture component, computed as:

$$\gamma_k(\mathbf{M}_n) = \frac{\varpi_k \, p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k)}{\sum_{j=1}^{K} \varpi_j \, p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}. \qquad (28)$$

By comparing (18), (19), (20) with (25), (26), (27), one can directly notice the similarity between the LE FS and the RFS. The vector difference (resp. the Euclidean distance) in the LE FS is replaced by the log map function (resp. the geodesic distance) in the RFS.

In [18], Ilea *et al.* have not exploited the FIM. In this paper, we propose to add this term in order to define the Riemannian Fisher vectors (RFV). To derive the FIM, the same assumption as the one given in Section 3.3.1 should be done, *i.e.* the assumption of nearly hard assignment, that is the soft assignment distribution $\gamma_k(\mathbf{M}_n)$ is sharply peaked on a single value of k for any observation $\mathbf{M}_n$. In that case, the FIM is block diagonal and admits a close-form expression detailed in [44]. In this paper, Zanini *et al.* have used this FIM to propose an online algorithm for estimating the parameters of a Riemannian Gaussian mixture model. Here, we propose to add this matrix in another context which is the derivation of a descriptor : the Riemannian FV. For more information on the derivation of the FIM for the Riemannian Gaussian mixture model, the interested reader is referred to [44]. Now that the FIM and the FS are properly defined for the Riemannian Gaussian mixture model, we can introduce the RFV by combining (25) to (27) and the definition of the FIM elements defined in [44] in (16). It yields that:

$$\mathcal{G}^{\mathcal{M}}_{(\bar{\mathbf{M}}_k)_1} = \frac{1}{\sqrt{\varpi_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \left(\frac{(\bar{\mathbf{M}}_k)_1 - (\mathbf{M}_n)_1}{\sigma_k} \right), \tag{29}$$

$$\mathcal{G}^{\mathcal{M}}_{(\bar{\mathbf{M}}_k)_2} = \frac{1}{\sqrt{\varpi_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \frac{\sqrt{\frac{m(m+1)}{2} - 1}}{\psi'_2(\eta_k)} \mathrm{Log}_{(\bar{\mathbf{M}}_k)_2}\left((\mathbf{M}_n)_2 \right), \tag{30}$$

$$\mathcal{G}^{\mathcal{M}}_{\sigma_k} = \frac{1}{\sqrt{\varpi_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \left(\frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k) - \psi'(\eta_k)}{\sqrt{\psi''(\eta_k)}} \right), \tag{31}$$

$$\mathcal{G}^{\mathcal{M}}_{\varpi_k} = \frac{1}{\sqrt{\varpi_k}} \sum_{n=1}^{N} \left(\gamma_k(\mathbf{M}_n) - \varpi_k \right). \qquad (32)$$

where $\eta = -\frac{1}{2\sigma^2}$, $\psi(\eta) = \log(Z(\sigma))$ and $\psi'(\cdot)$ (resp. $\psi''(\cdot)$) are the first (resp. the second) order derivative of the $\psi(\cdot)$ function with respect to η. $\psi'_2(\eta) = \psi'(\eta) + \frac{1}{2\eta}$. $(\mathbf{M})_1 = \log \det \mathbf{M}$ is a scalar element lying in $\mathbb{R}$ and $(\mathbf{M})_2 = e^{-\frac{(\mathbf{M})_1}{m}} \mathbf{M}$ is a covariance matrix of unit determinant.

Unsurprisingly, this definition of the RFV can be interpreted as a direct extension of the FV computed in the Euclidean case to the Riemannian case. In particular (22), (23) and (24) are retrieved when the normalization factor $Z(\sigma)$ is set to $\sigma\sqrt{2\pi}$ in (29), (31) and (32).

In the end, the RFVs are obtained by concatenating some, or all of the derivatives in (29), (30), (31), (32). Note also that since (30) is a matrix, the vectorization operator $\mathrm{Vec}(\cdot)$ is used to represent it as a vector.

3.3.3 Relation with VLAD

As stated before, the VLAD descriptor can be retrieved from the FV model. In this case, only the derivatives with respect to the central element ($\bar{\mathbf{m}}_k^d$ or $\bar{\mathbf{M}}_k$) are considered. Two assumptions are also made, that are the homoscedasticity assumption, *i.e.* $\sigma_k = \sigma$, $\forall k = 1, \dots, K$ and the hard assignment scheme

$$\gamma_k(\mathbf{M}) = \begin{cases} 1, & \text{if } \mathbf{M} \in c_k \\ 0, & \text{otherwise,} \end{cases} \tag{33}$$

where $\mathbf{M} \in c_k$ are the elements assigned to cluster c_k and $k = 1, \dots, K$.

By taking into account these hypotheses, it can be noticed that (18) reduces to (11), confirming that LE FV are a generalization of LE VLAD descriptors. The same remark can be done for the approach exploiting the affine invariant Riemannian metric where the RFV model can be viewed as an extension of the RVLAD model. The proposed RFV gives a mathematical explanation of the RVLAD descriptor which has been introduced in [11] by an analogy between the Euclidean space (for the VLAD descriptor) and the Riemannian manifold (for the RVLAD descriptor).

3.4. Post-processing

Once the set of SPD is encoded by one of the previously exposed coding methods (BoW, VLAD or FV), a post-processing step is classically employed. In the framework of feature coding, the post-processing step consists in two possible normalization steps: the power and ℓ_2 normalization. Depending on the considered coding method, one or both normalization steps are applied. For instance, for FV and VLAD based methods, both normalizations are used [12, 18], while for BoW based methods only the ℓ_2 normalization is considered [13].

4. Application to texture image classification

This section introduces an application to texture image classification. The aim of this experiment is three-fold. The first objective is to compare two Riemannian metrics : the log-Euclidean and the affine invariant Riemmannian metrics. The second objective is to analyze the potential of the proposed FV based methods compared to the recently proposed BoW and VLAD based models. And finally, the third objective is to evaluate the advantage of including the FIM in the derivation of the FVs, *i.e.* comparing the performance between FS and FV.

To answer these questions, an experiment is conducted on four conventional texture databases, namely the VisTex, Brodatz, Outex and USPtex database. The first stage is the feature extraction step which consists in representing each texture image by a set of covariance matrices. Since the experiment purpose is not to find the best classification accuracies on these databases, but rather to compare the different strategies (choice of the metric, influence of the coding model) on the same features, we have adopted the simple but effective region covariance descriptors (RcovD) used in [40]. The extracted RCovD are the estimated covariance matrices of vectors $\mathbf{v}(x, y)$ computed on sliding patches of size 15×15 pixels where:

$$\mathbf{v}(x,y) = \left[I(x,y), \left| \frac{\partial I(x,y)}{\partial x} \right|, \left| \frac{\partial I(x,y)}{\partial y} \right|, \left| \frac{\partial^2 I(x,y)}{\partial x^2} \right|, \left| \frac{\partial^2 I(x,y)}{\partial y^2} \right| \right]^T.$$
$$\tag{34}$$

In this experiment, the patches are overlapped by 50%. The fast covariance matrix computation algorithm based on integral images presented in [40] is adopted to speed-up the computation time of this feature extraction step. It yields that each texture class is composed by a set $\mathbf{M}_1, \dots, \mathbf{M}_N$ of N covariance matrices, that are elements in $\mathcal{P}_5$.

For each class, codewords are represented by the estimated parameters of the mixture of K Gaussian distributions. For this experiment, the number of modes K is set to 3. In the end, the codebook is obtained by concatenating the previously extracted codewords (for each texture class).

Once the codebook is created, the covariance matrices of each image are encoded by one of the previously described method (namely BoW, VLAD, FS or FV) adapted to the log-Euclidean or affine invariant Riemannian metric. Then after some post-processing (power and/or ℓ_2 normalization), the obtained feature vectors are classified. Here, the SVM classifier with Gaussian kernel is used. The parameter of the Gaussian kernel is optimized by using a cross validation procedure on the training set.

The whole procedure is repeated 10 times for different training and testing sets. Each time, half of the database is used for training while the remaining half is used for testing. Tables 1 to 4 show the classification performance in term of overall accuracy (mean $\pm$ standard deviation) on the VisTex, Brodatz, Outex and USPtex databases.

As the FS and FV descriptors are obtained by deriving the log-likelihood function with respect to the weight, dispersion and centroid parameters, the contribution of each term to the classification accuracy can be analyzed. Therefore, seven versions of the FS and FV descriptors are considered by analyzing separately the contribution of each term or by combining these different terms. For example, the row "FS : ϖ" indicates the classification results when only the derivatives with respect to the weights are considered to derive the FS (see (20) and (27)).

One can observe that the best performance are consistently obtained for the LE FV. As stated in many computer vision applications [27, 23, 9], the most informative FV descriptor is obtained when the derivation is with respect to the mean $\bar{\mathbf{m}}_k^d$ (or $\bar{\mathbf{M}}$ when the affine invariant Riemannian

Table 1. Classification results on the VisTex database (40 classes).

Coding method	Log-Euclidean metric	Affine invariant Riemannian metric
BoW	86.4 ± 0.01	85.9 ± 0.01
VLAD	91.3 ± 0.1	82.8 ± 0.02
E-VLAD	91.6 ± 0.01	
FS : ϖ	82.4 ± 0.01	88.4 ± 0.01
FS : σ	91.6 ± 0.01	87.8 ± 0.01
FS : $\bar{\mathbf{M}}$	95.3 ± 0.01	88.9 ± 0.01
FS : σ, ϖ	91.3 ± 0.01	87.9 ± 0.01
FS : $\bar{\mathbf{M}}, \varpi$	95.1 ± 0.01	90.0 ± 0.01
FS : $\bar{\mathbf{M}}, \sigma$	95.2 ± 0.01	91.2 ± 0.01
FS : $\bar{\mathbf{M}}, \sigma, \varpi$	95.1 ± 0.01	91.2 ± 0.01
FV : ϖ	81.5 ± 0.01	88.4 ± 0.01
FV : σ	92.9 ± 0.01	80.5 ± 0.01
FV : $\bar{\mathbf{M}}$	95.5 ± 0.01	91.3 ± 0.01
FV : σ, ϖ	92.9 ± 0.01	89.0 ± 0.01
FV : $\bar{\mathbf{M}}, \varpi$	$\mathbf{95.7 \pm 0.01}$	92.6 ± 0.01
FV : $\bar{\mathbf{M}}, \sigma$	95.6 ± 0.01	92.7 ± 0.01
FV : $\bar{\mathbf{M}}, \sigma, \varpi$	95.4 ± 0.01	93.2 ± 0.01

Table 2. Classification results on the Brodatz database (112 classes).

Coding method	Log-Euclidean metric	Affine invariant Riemannian metric
BoW	92.0 ± 0.01	92.1 ± 0.01
VLAD	92.5 ± 0.01	88.3 ± 0.01
E-VLAD	92.4 ± 0.01	
FS : ϖ	79.4 ± 0.01	91.5 ± 0.01
FS : σ	88.4 ± 0.03	91.4 ± 0.01
FS : $\bar{\mathbf{M}}$	92.5 ± 0.01	90.1 ± 0.01
FS : σ, ϖ	89.4 ± 0.03	91.3 ± 0.01
FS : $\bar{\mathbf{M}}, \varpi$	92.7 ± 0.01	91.1 ± 0.01
FS : $\bar{\mathbf{M}}, \sigma$	90.3 ± 0.01	91.7 ± 0.01
FS : $\bar{\mathbf{M}}, \sigma, \varpi$	90.8 ± 0.03	91.6 ± 0.01
FV : ϖ	79.6 ± 0.01	91.4 ± 0.01
FV : σ	92.0 ± 0.01	87.5 ± 0.01
FV : $\bar{\mathbf{M}}$	93.5 ± 0.01	92.9 ± 0.01
FV : σ, ϖ	92.5 ± 0.01	90.7 ± 0.01
FV : $\bar{\mathbf{M}}, \varpi$	$\mathbf{93.7 \pm 0.01}$	93.2 ± 0.01
FV : $\bar{\mathbf{M}}, \sigma$	93.1 ± 0.01	93.1 ± 0.01
FV : $\bar{\mathbf{M}}, \sigma, \varpi$	92.9 ± 0.01	93.2 ± 0.01

Table 3. Classification results on the Outex database (68 classes).

Coding method	Log-Euclidean metric	Affine invariant Riemannian metric
BoW	83.5 ± 0.01	83.7 ± 0.01
VLAD	85.9 ± 0.01	82.0 ± 0.01
E-VLAD	85.1 ± 0.01	
FS : ϖ	72.8 ± 0.01	84.7 ± 0.01
FS : σ	83.3 ± 0.01	84.7 ± 0.01
FS : $\bar{\mathbf{M}}$	87.2 ± 0.01	83.8 ± 0.01
FS : σ, ϖ	84.0 ± 0.01	84.6 ± 0.01
FS : $\bar{\mathbf{M}}, \varpi$	$\mathbf{88.0 \pm 0.01}$	84.2 ± 0.01
FS : $\bar{\mathbf{M}}, \sigma$	86.7 ± 0.01	84.9 ± 0.01
FS : $\bar{\mathbf{M}}, \sigma, \varpi$	87.6 ± 0.01	85.2 ± 0.01
FV : ϖ	72.6 ± 0.01	84.5 ± 0.01
FV : σ	84.0 ± 0.02	79.2 ± 0.01
FV : $\bar{\mathbf{M}}$	87.3 ± 0.01	85.4 ± 0.01
FV : σ, ϖ	84.3 ± 0.01	83.9 ± 0.01
FV : $\bar{\mathbf{M}}, \varpi$	87.9 ± 0.01	86.0 ± 0.01
FV : $\bar{\mathbf{M}}, \sigma$	87.1 ± 0.01	86.0 ± 0.01
FV : $\bar{\mathbf{M}}, \sigma, \varpi$	87.2 ± 0.01	86.3 ± 0.01

Table 4. Classification results on the USPtex database (191 classes).

Coding method	Log-Euclidean metric	Affine invariant Riemannian metric
BoW	79.9 ± 0.01	80.2 ± 0.01
VLAD	86.5 ± 0.01	78.9 ± 0.01
E-VLAD	86.7 ± 0.01	
FS : ϖ	56.8 ± 0.01	80.7 ± 0.01
FS : σ	71.9 ± 0.01	81.7 ± 0.01
FS : $\bar{\mathbf{M}}$	84.8 ± 0.03	84.7 ± 0.01
FS : σ, ϖ	71.5 ± 0.02	81.7 ± 0.01
FS : $\bar{\mathbf{M}}, \varpi$	85.1 ± 0.02	85.2 ± 0.01
FS : $\bar{\mathbf{M}}, \sigma$	76.8 ± 0.03	84.0 ± 0.01
FS : $\bar{\mathbf{M}}, \sigma, \varpi$	77.9 ± 0.03	84.0 ± 0.01
FV : ϖ	56.3 ± 0.01	80.7 ± 0.02
FV : σ	84.4 ± 0.01	77.6 ± 0.01
FV : $\bar{\mathbf{M}}$	88.3 ± 0.01	87.0 ± 0.01
FV : σ, ϖ	84.0 ± 0.01	80.8 ± 0.01
FV : $\bar{\mathbf{M}}, \varpi$	88.0 ± 0.01	87.0 ± 0.01
FV : $\bar{\mathbf{M}}, \sigma$	87.7 ± 0.01	87.3 ± 0.01
FV : $\bar{\mathbf{M}}, \sigma, \varpi$	$\mathbf{88.4 \pm 0.01}$	87.2 ± 0.01

metric is considered). A significant gain of about 1 to 4% is observed compared to the FV descriptor obtained by deriving with respect to the standard deviation. This gain is observed for both the log-Euclidean and the affine invariant Riemannian metric. Note also that the use of the FIM in the derivation of the FV allows to improve the classification accuracy. As observed for the four considered databases, a gain of about 1 to 3% is obtained when comparing FV : $\bar{\mathbf{M}}$ with FS : $\bar{\mathbf{M}}$.

Finally, for these four experiments on texture image classification, the proposed FV descriptors outperform the state-of-the-art BoW and VLAD based descriptors. Classifying with the best FV descriptor yields to a gain of about 1 to 4% compared to the best BoW and VLAD based descriptors.

5. Conclusion

Starting from the Gaussian mixture model (for the log-Euclidean metric) and the Riemannian Gaussian mixture model (for the affine invariant Riemannian metric), we have shown how Fisher vectors can be used to encode a set of covariance matrices. The proposed LE FV and RFV can be interpreted as a generalization of the BoW and VLAD based approach. The experimental results on the four considered texture databases have shown that: (i) the use of the FIM in the derivation of the FV allows to improve the classification accuracy, (ii) the descriptors based on the log-Euclidean metric lead to better classification results than those based on the affine invariant Riemannian metric and (iii) the proposed FV based descriptors outperforms the state-of-the-art BoW and VLAD based descriptors.

References

[1] R. Arandjelović and A. Zisserman. All about VLAD. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[2] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. In *Magnetic Resonance in Medicine*, volume 56, pages 411–421, Aug 2006.

[3] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten. Classification of covariance matrices using a Riemannian-based kernel for BCI applications. *NeuroComputing*, 112:172–178, 2013.

[4] F. Barbaresco, M. Arnaudon, and L. Yang. Riemannian medians and means with applications to radar signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 7(4):595–604, 2013.

[5] Y. Chen, A. Wiesel, and A. O. Hero. Robust shrinkage estimation of high-dimensional covariance matrices. *IEEE Transactions on Signal Processing*, 59(9):4097–4107, Sept 2011.

[6] G. Cheng and B. C. Vemuri. A novel dynamic system in the space of SPD matrices with applications to appearance tracking. *SIAM Journal on Imaging Sciences*, 6(1):592–615, 2013.

[7] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, European Conference on Computer Vision*, pages 1–22, 2004.

[8] R. de Luis-García, C.-F. Westin, and C. Alberola-López. Gaussian mixtures on tensor fields for segmentation: applications to medical imaging. *Computerized Medical Imaging and Graphics*, 35(1):16–30, 2011.

[9] M. Douze, A. Ramisa, and C. Schmid. Combining attributes and Fisher vectors for efficient image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 745–752, 2011.

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

[11] M. Faraki, M. T. Harandi, and F. Porikli. More about VLAD: A leap from Euclidean to Riemannian manifolds. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4951–4960, 2015.

[12] M. Faraki, M. T. Harandi, A. Wiliem, and B. C. Lovell. Fisher tensors for classifying human epithelial cells. *Pattern Recognition*, 47(7):2348 – 2359, 2014.

[13] M. Faraki, M. Palhang, and C. Sanderson. Log-euclidean bag of words for human action recognition. *IET Computer Vision*, 9(3):331–339, 2015.

[14] P. Formont, F. Pascal, G. Vasile, J. Ovarlez, and L. Ferro-Famil. Statistical classification for heterogeneous polarimetric SAR images. *IEEE Journal of Selected Topics in Signal Processing*, 5(3):567–576, 2011.

[15] G. Garcia and J. M. Oller. What does intrinsic mean in statistical estimation? *Statistics and Operations Research Transactions*, 30(2):125–170, 2006.

[16] M. Greco, S. Fortunati, and F. Gini. Maximum likelihood covariance matrix estimation for complex elliptically symmetric distributions under mismatched conditions. *Signal Processing*, 104:381–386, 2014.

[17] N. J. Higham. *Functions of matrices: theory and computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.

[18] I. Ilea, L. Bombrun, C. Germain, R. Terebes, M. Borda, and Y. Berthoumieu. Texture image classification with Riemannian Fisher vectors. In *IEEE International Conference on Image Processing*, pages 3543 – 3547, 2016.

[19] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *In Advances in Neural Information Processing Systems 11*, pages 487–493. MIT Press, 1998.

[20] A. T. James. *The variance information manifold and the functions on it*, pages 157 – 169. Academic Press, 1973.

[21] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[22] T. Joachims. Text categorization with suport vector machines: learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, pages 137–142. Springer-Verlag, 1998.

[23] J. Krapac, J. Verbeek, and F. Jurie. Modeling spatial layout with Fisher vectors for image categorization. In *2011 International Conference on Computer Vision*, pages 1487–1494, Nov 2011.

[24] K. Mader and G. Reese. Using covariance matrices as feature descriptors for vehicle detection from a fixed camera. *ArXiv e-prints*, Feb. 2012.

[25] R. J. Muirhead. *Aspects of multivariate statistical theory*. Wiley Series in Probability and Statistics. Wiley, 1982.

[26] E. Ollila and V. Koivunen. Robust antenna array processing using M-estimators of pseudo-covariance. In *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications*, volume 3, pages 2659–2663, 2003.

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

[28] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed Fisher vectors. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010*, pages 3384–3391, 2010.

[29] F. Perronnin, J. Sánchez, and T. Mensink. *Improving the Fisher kernel for large-scale image classification*, volume 6314 of *Lecture Notes in Computer Science*, pages 143–156. Springer Berlin Heidelberg, 2010.

[30] J. Robinson. Covariance matrix estimation for appearance-based face image processing. *Proceedings of the British Machine Vision Conference 2005*, pages 389–398, 2005.

[31] R. Rosu, M. Donias, L. Bombrun, S. Said, O. Regniers, and J. P. Da Costa. Structure tensor Riemannian statistical models for CBIR and classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 55(1):248–260, Jan 2017.

[32] S. Said, L. Bombrun, and Y. Berthoumieu. Texture classification using Rao's distance on the space of covariance matrices. In *Geometric Science of Information*, 2015.

[33] S. Said, L. Bombrun, Y. Berthoumieu, and J. H. Manton. Riemannian Gaussian distributions on the space of symmetric positive definite matrices. *IEEE Transactions on Information Theory*, 63(4):2153–2170, April 2017.

[34] S. Said, H. Hajri, L. Bombrun, and B. C. Vemuri. Gaussian distributions on Riemannian symmetric spaces: Statistical learning with structured covariance matrices. *IEEE Transactions on Information Theory*, 64(2):752–772, Feb 2018.

[35] H. Salehian, G. Cheng, B. C. Vemuri, and J. Ho. Recursive estimation of the stein center of SPD matrices and its applications. In *IEEE International Conference on Computer Vision*, pages 1793–1800. IEEE Computer Society, 2013.

[36] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[37] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the Fisher vector: Theory and practice. *International Journal of Computer Vision*, 105(3):222–245, 2013.

[38] S. Sra. A new metric on the manifold of kernel matrices with application to matrix geometric means. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 144–152. Curran Associates, Inc., 2012.

[39] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2273–2286, Nov 2011.

[40] O. Tuzel, F. Porikli, and P. Meer. *Region covariance: a fast descriptor for detection and classification*, volume 3952 of *Lecture Notes in Computer Science*, pages 589–600. Springer Berlin Heidelberg, 2006.

[41] L. Yang, M. Arnaudon, and F. Barbaresco. Riemannian median, geometry of covariance matrices and radar target detection. *European Radar Conference*, pages 415–418, 2010.

[42] C. Yuan, W. Hu, X. Li, S. Maybank, and G. Luo. *Human Action Recognition under Log-Euclidean Riemannian Metric*, pages 343–353. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[43] P. Zanini, M. Congedo, C. Jutten, S. Said, and Y. Berthoumieu. Parameters estimate of Riemannian Gaussian distribution in the manifold of covariance matrices. In *IEEE Sensor Array and Multichannel Signal Processing Workshop*, Rio de Janeiro, Brazil, 2016.

[44] P. Zanini, S. Said, Y. Berthoumieu, M. Congedo, and C. Jutten. *Riemannian Online Algorithms for Estimating Mixture Model Parameters*, pages 675–683. Springer International Publishing, Cham, 2017.