Fisher Vector Coding for Covariance Matrix Descriptors Based on the Log-Euclidean and Affine Invariant Riemannian Metrics

Ioana Ilea, Lionel Bombrun, Salem Said, Yannick Berthoumieu

MDPI

Article

Fisher vector coding for covariance matrix descriptors based on the log-Euclidean and affine invariant Riemannian metrics

Ioana Ilea [1], Lionel Bombrun [2*], Salem Said [2] and Yannick Berthoumieu [2]

[1] Technical University of Cluj-Napoca, 71-73 Calea Dorobanților, 400114 Cluj-Napoca, Romania; ioana.ilea@com.utcluj.ro

[2] Laboratoire IMS, Université de Bordeaux, CNRS, UMR 5218, 351 cours de la libération, 33400 Talence, France; firstname.lastname@ims-bordeaux.fr

* Correspondence: lionel.bombrun@ims-bordeaux.fr; Tel.: +33-540-00-2473

Version June 7, 2018 submitted to J. Imaging

Abstract: This paper presents an overview of coding methods used to encode a set of covariance matrices. Starting from a Gaussian mixture model (GMM) adapted to the Log-Euclidean (LE) or affine invariant Riemannian metric, we propose a Fisher Vector (FV) descriptor adapted to each of these metrics: the Log-Euclidean Fisher Vectors (LE FV) and the Riemannian Fisher Vectors (RFV). Some experiments on texture and head pose image classification are conducted 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. And finally, some experiments are conducted in order to provide fairer comparison between the different coding strategies. This includes some comparisons between anisotropic and isotropic models, and a estimation performance analysis of the GMM dispersion parameter for covariance matrices of large dimension.

Keywords: Bag of words; vector of locally aggregated descriptors; Fisher vector; log-Euclidean metric; affine invariant Riemannian metric; covariance matrix

1. Introduction

In supervised classification, the goal is to tag an image with one class name based on its content. In the beginning of the 2000s, the leading approaches were based on feature coding. Among the most employed coding based methods, there are the bag of words model (BoW) [1], the vector of locally aggregated descriptors (VLAD) [2,3], the Fisher score (FS) [4] and the Fisher vectors (FV) [5–7]. The success of these methods is based on their main advantages. First, the information obtained by feature coding can be used in a wide variety of applications, including image classification [5,8,9], text retrieval [10], action and face recognition [11], etc. Second, combined with powerful local handcrafted features, such as SIFT, they are robust to transformations like scaling, translation, or occlusion [11].

Nevertheless, in 2012, the ImageNet Large Scale Visual Recognition Challenge has shown that Convolutional Neural Networks [12,13] (CNNs) can outperform FV descriptors. Since then, in order to take advantage of both worlds, some hybrid classification architectures have been proposed to combine FV and CNN [14]. For example, Perronnin *et al.* have proposed to train a network of fully connected layers on the FV descriptors [15]. Another hybrid architecture is the deep Fisher network composed by stacking several FV layers [16]. Some authors have proposed to extract convolutional features from different layers of the network, and then to use VLAD or FV encoding to encode features into a single vector for each image [17–19]. These latter features can also be combined with features issued from the fully connected layers in order to improve the classification accuracy [20].

At the same time, many authors have proposed to extend the formalism of encoding to features lying in a non-Euclidean space. This is the case of covariance matrices that have already demonstrated their importance as descriptors related to array processing [21], radar detection [22–25], image segmentation [26,27], face detection [28], vehicle detection [29], or classification [11,30–32], etc. As mentioned in [33], the use of covariance matrices has several advantages. First, they are able to merge the information provided by different features. Second, they are low dimensional descriptors, independent of the dataset size. Third, in the context of image and video processing, efficient methods for fast computation are available [34].

Nevertheless, since covariance matrices are positive definite matrices, conventional tools developed in the Euclidean space are not well adapted to model the underlying scatter of the data points which are covariance matrices. 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, FS 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 LE and affine invariant Riemannian metrics. This yields to the so-called Log-Euclidean bag of words (LE BoW) [33,35], bag of Riemannian words (BoRW) [36], Log-Euclidean vector of locally aggregated descriptors (LE VLAD) [11], extrinsic vector of locally aggregated descriptors (E-VLAD) [37] and intrinsic Riemannian vector of locally aggregated descriptors (RVLAD) [11]. All these approaches have been proposed by a direct analogy between the Euclidean and the Riemannian case. For that, the codebook used to encode the covariance matrix set is the standard k-means algorithm adapted to the LE and affine invariant Riemannian metrics.

Contrary to the BoW and VLAD based coding methods, a soft codebook issued from a Gaussian mixture model (GMM) should be learned for FS or FV encoding. This paper aims to present how FS and FV can be used to encode a set of covariance matrices [38]. 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 used: the LE and the affine invariant Riemannian metrics. To summarize, we provide four main contributions:

- First, based on the conventional multivariate GMM, we introduce the log-Euclidean Fisher score (LE FS). This descriptor can be interpreted as the FS computed on the log-Euclidean vector representation of the covariance matrices set.
- Second, we have recently introduced a Gaussian distribution on the space $\mathcal{P}_m$: the Riemannian Gaussian distribution [39]. This latter allows the definition of a GMM on the space of covariance matrices and an Expectation Maximization (EM) algorithm can hence be considered to learn the codebook [32]. Starting from this observation, we define the Riemannian Fisher score (RFS) [40] which can be interpreted as an extension of the RVLAD descriptor proposed in [11].
- The third main contribution is to highlight the impact of the Fisher information matrix (FIM) in the derivation of the FV. For that, the Log-Euclidean Fisher Vectors (LE FV) and the Riemannian Fisher Vectors (RFV) are introduced as an extension of the LE FS and the RFS.
- And fourth, all these coding methods will be compared on two image processing applications consisting in texture and head pose image classification. Some experiments will also be conducted in order to provide fairer comparison between the different coding strategies. It includes some comparisons between anisotropic and isotropic models. An estimation performance analysis of the dispersion parameter for covariance matrices of large dimension will also be studied.

As previously mentioned, hybrid architectures can be employed to combine FV with CNN. The adaptation of the proposed FV descriptors to these architecture is outside the scope of this paper but will remain one of the perspective of this work.

The paper is structured as follows. Section 2 introduces the workflow presenting the general idea of feature coding based classification methods. Section 3 presents the codebook generation on the manifold of SPD covariance matrices. Section 4 introduces a theoretical study of the feature encoding methods (BoW, VLAD, FS and FV) based on the LE and affine invariant Riemannian metrics. Section 5

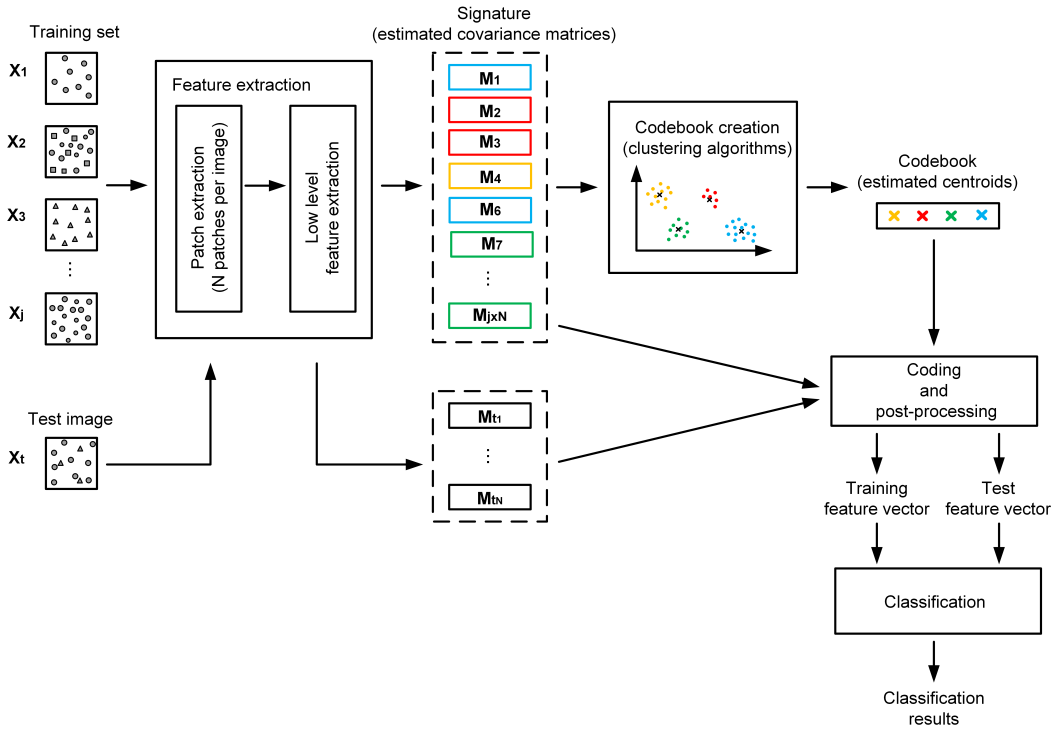

Figure 1. Workflow presenting the general idea of feature coding based classification methods.

shows two applications of these descriptors to texture and head pose image classification. And finally, Section 6 synthesizes the main conclusions and perspectives of this work.

2. General framework

The general workflow is presented in Fig. 1 and it consists in the following steps:

1. *Patch extraction* is the starting step of the classification algorithm. At the beginning, the images are divided in patches, either in a dense way, by means of fixed grids, or in a non-dense way, based on representative points such as SIFT for example.
2. *A low level feature extraction* step is then applied in order to extract some characteristics (such as spatial gradient components). These low-level handcrafted features capture the information contained in each patch.
3. *The covariance matrix* of these features are then computed. As a result, each image is represented as a set of covariance matrices which compose the signature of an image.
4. *The codebook generation* starts from the previously extracted covariance matrices. The purpose of this step is to identify the features containing the significant information. Usually, this procedure is performed by means of clustering algorithms, such as the k-means or expectation-maximization (EM) algorithm. Knowing that the features are covariance matrices, one of the following approaches can be chosen. The first one considers the LE metric. It consits in projecting the covariance matrices in the LE space [33,35] and then standard clustering algorithms for multivariate Gaussian distributions are used. The second approach considers the affine invariant Riemannian metric to measure the similarity between two covariance matrices. In this context, the conventional k-means or EM algorithm should be readapted to this metric [11,36,40]. For both approaches, the dataset is partitioned into a predefined number of clusters, each of them being described by parameters, such as the cluster's centroid, the dispersion and the associated weight. The obtained features are called codewords and they are grouped in a codebook, also called a dictionary.

5. *Feature encoding* is based on the created codebook and it consists in projecting the extracted covariance matrices onto the codebook space. For this purpose, approaches like BoW, VLAD and FV can be employed, for both the LE and affine invariant Riemannian metrics. According to [41], these are global coding strategies, that describe the entire set of features, and not the individual ones. Essentially, this is accomplished using probability density distributions to model the feature space. More precisely, they can be viewed either as voting-based methods depending on histograms, or as Fisher coding-based methods by using Gaussian mixture models adapted to the considered metric [39,42].
6. *Post-processing* is often applied after the feature encoding step, in order to minimize the influence of background information on the image signature [6] and to correct the independence assumption made on the patches [7]. Therefore, two types of normalization are used, namely the power [7] and ℓ_2 [6] normalizations.
7. *Classification* is the final step, achieved by associating the test images to the class of the most similar training observations. In practice, algorithms such as k-nearest neighbors, support vector machine or random forest can be used.

As shown in Fig. 1, the codebook generation along with the feature encoding are the two central steps in this framework. The next two sections present a detailed analysis of how these steps are adapted to covariance matrix features.

3. 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 LE and the affine invariant Riemannian metric. The next two subsections describe these two strategies.

3.1. Log-Euclidean codebook

Let $\boldsymbol{\mathcal{M}} = \{\mathbf{M}_n\}_{n=1:N}$, with $\mathbf{M}_n \in \mathcal{P}_m$, be a sample of N training SPD matrices of size $m \times m$. The LE codebook is obtained by considering the LE 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 $\mathbf{M}_n^{LE} = \log \mathbf{M}_n$ [33,43,44]. 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:

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

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 $\boldsymbol{\mathcal{M}}$, *i.e.* $\{\mathbf{m}_n\}_{n=1:N}$, can be assumed to be independent and identically distributed (i.i.d.) samples from a mixture of K multivariate Gaussian distributions, whose probability density function is

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

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

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

where $(\cdot)^T$ is the transpose operator, $|\cdot|$ is the determinant, $\bar{\mathbf{m}}_k \in \mathbb{R}^{\frac{m(m+1)}{2}}$, $\Sigma_k \in \mathcal{P}_{m(m+1)/2}$ and $\varpi_k \in \mathbb{R}$. In addition, the covariance matrix is assumed to be diagonal, *i.e.* $\sigma_k^2 = \text{diag}(\Sigma_k) \in \mathbb{R}^{\frac{m(m+1)}{2}}$ is the variance vector. For such a model, the classical k-means or 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 LE codebook.

3.2. New Riemannian codebook

In this section, we present the construction of the Riemannian codebook which is based on the affine invariant Riemannian metric. We recall some properties of the manifold of SPD matrices and introduce the Riemannian Gaussian mixture model.

3.2.1. Riemannian geometry of the space of SPD matrices

The space $\mathcal{P}_m$ of $m \times m$ real SPD matrices $\mathbf{M}$ satisfies the following conditions:

$$\mathbf{M} - \mathbf{M}^T = 0 \tag{4}$$

and

$$\mathbf{x}^T \mathbf{M} \mathbf{x} > 0, \tag{5}$$

$\forall \mathbf{x} \in \mathbb{R}^m$ and $\mathbf{x} \neq 0$.

In this space, the Rao-Fisher metric defines a distance, called the Rao's geodesic distance [45,46], given by the length of the shortest curve connecting two points in $\mathcal{P}_m$. Mathematically, this definition can be stated as follows [32]. Let $\mathbf{M}_1, \mathbf{M}_2$ be two points in $\mathcal{P}_m$ and $c : [0,1] \longrightarrow \mathcal{P}_m$ a differentiable curve, with $c(0) = \mathbf{M}_1$ and $c(1) = \mathbf{M}_2$. Thus, the length of curve c, denoted by $L(c)$ is defined as:

$$L(c) = \int_0^1 \left\| \frac{dc}{dt} \right\| dt. \tag{6}$$

The geodesic distance $d : \mathcal{P}_m \times \mathcal{P}_m \longrightarrow \mathbb{R}_+$ between $\mathbf{M}_1$ and $\mathbf{M}_2$ is the infimum of $L(c)$ with respect to all differentiable curves c. Based on the properties of Rao-Fisher metric, it has been shown that the unique curve γ fulfilling this condition is [45,46]:

$$\gamma(t) = \mathbf{M}_1^{\frac{1}{2}} \left(\mathbf{M}_1^{-\frac{1}{2}} \mathbf{M}_2 \mathbf{M}_1^{-\frac{1}{2}} \right)^t \mathbf{M}_1^{\frac{1}{2}}, \tag{7}$$

called the geodesic connecting $\mathbf{M}_1$ and $\mathbf{M}_2$. Moreover, the distance between two points in $\mathcal{P}_m$ can be expressed as [47]:

$$d^2(\mathbf{M}_1, \mathbf{M}_2) = \text{tr}\left(\left[\log\left(\mathbf{M}_1^{-\frac{1}{2}} \mathbf{M}_2 \mathbf{M}_1^{-\frac{1}{2}} \right) \right]^2 \right) = \sum_{i=1}^{m} (\ln \lambda_i)^2, \tag{8}$$

with λ_i, $i = 1, \ldots, m$ being the eigenvalues of $\mathbf{M}_1^{-1} \mathbf{M}_2$.

The affine invariant Riemannian (Rao-Fisher) metric can be also used to define the Riemannian volume element [45]:

$$dv(\mathbf{M}) = |\mathbf{M}|^{-\frac{m+1}{2}} \prod_{i \leq j} d\mathbf{M}_{ij}. \tag{9}$$

For each point on the manifold $\mathbf{M}_1 \in \mathcal{P}_m$, the tangent space at $\mathbf{M}_1$, denoted by $T_{\mathbf{M}_1}$ can be defined. This space contains the vectors $\mathbf{V}_T$ that are tangent to all possible curves passing through $\mathbf{M}_1$. The correspondence between a point on the manifold and its tangent space can be achieved by using two operators: the Riemannian exponential mapping and the Riemannian logarithm mapping [48,49].

More precisely, the Riemannian exponential mapping for a point $\mathbf{M}_1 \in \mathcal{P}_m$ and the tangent vector $\mathbf{V}_T$ is given by [48,49]:

$$\mathbf{M}_2 = \mathrm{Exp}_{\mathbf{M}_1}(\mathbf{V}_T) = \mathbf{M}_1^{\frac{1}{2}} \exp\left(\mathbf{M}_1^{-\frac{1}{2}} \mathbf{V}_T \mathbf{M}_1^{-\frac{1}{2}}\right) \mathbf{M}_1^{\frac{1}{2}}, \tag{10}$$

where $\exp(\cdot)$ is the matrix exponential. By this transformation, the tangent vector $\mathbf{V}_T$ can be mapped on the manifold.

Further on, the inverse of the Riemannian exponential mapping is the Riemannian logarithm mapping. For two points $\mathbf{M}_1, \mathbf{M}_2 \in \mathcal{P}_m$, this operator is given by [48,49]:

$$\mathbf{V}_T = \mathrm{Log}_{\mathbf{M}_1}(\mathbf{M}_2) = \mathbf{M}_1^{\frac{1}{2}} \log\left(\mathbf{M}_1^{-\frac{1}{2}} \mathbf{M}_2 \mathbf{M}_1^{-\frac{1}{2}}\right) \mathbf{M}_1^{\frac{1}{2}}, \tag{11}$$

where $\log(\cdot)$ is the matrix logarithm. In practice, this operation gives the tangent vector $\mathbf{V}_T$, by transforming the geodesic γ in a straight line in the tangent space. In addition, the geodesic's length between $\mathbf{M}_1$ and $\mathbf{M}_2$ is equal to the norm of the tangent vector $\mathbf{V}_T$.

3.2.2. Mixture of Riemannian Gaussian distribution

Riemannian Gaussian model

In order to model the space $\mathcal{P}_m$ of SPD covariance matrices, a generative model has been introduced in [39,42]: the Riemannian Gaussian distribution (RGD). For this model, the probability density function with respect to the Riemannian volume element given in (9) is defined as follow [39,42]:

$$p(\mathbf{M}_n | \bar{\mathbf{M}}, \sigma) = \frac{1}{Z(\sigma)} \exp\left\{ -\frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}})}{2\sigma^2} \right\}, \tag{12}$$

where $\bar{\mathbf{M}}$ and σ are the distribution parameters, representing respectively the central value (centroid) and the dispersion. $d(\cdot)$ is the Riemannian distance given in (8) and $Z(\sigma)$ is a normalization factor independent of $\bar{\mathbf{M}}$ [39,50].

$$Z(\sigma) = \frac{8^{\frac{m(m-1)}{4}} \pi^{m^2/2}}{m!\, \Gamma_m(m/2)} \int_{\mathbb{R}^m} e^{-\frac{\|\mathbf{r}\|^2}{2\sigma^2}} \prod_{i<j} \sinh\left(\frac{|r_i - r_j|}{2}\right) \prod_{i=1}^{m} \mathrm{d}r_i \tag{13}$$

with Γ_m the multivariate Gamma function [51]. 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 $\mathcal{N}(0, \sigma^2 I_m)$ [39]. Afterwards, a cubic spline interpolation can be used to smooth this function [52].

Mixture model for RGDs

As for the LE 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 $\boldsymbol{\mathcal{M}} = \{\mathbf{M}_n\}_{n=1:N}$, are i.i.d. samples from a mixture of K RGDs. In this case, the likelihood of $\boldsymbol{\mathcal{M}}$ is given by:

$$p(\boldsymbol{\mathcal{M}} | \theta) = \prod_{n=1}^{N} p(\mathbf{M}_n | \theta) = \prod_{n=1}^{N} \sum_{k=1}^{K} \varpi_k p(\mathbf{M}_n | \bar{\mathbf{M}}_k, \sigma_k), \tag{14}$$

where $p(\mathbf{M}_n | \bar{\mathbf{M}}_k, \sigma_k)$ is the RGD defined in (12) and $\theta = \{(\varpi_k, \bar{\mathbf{M}}_k, \sigma_k)_{1 \leq k \leq K}\}$ 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 [36,53]. Nevertheless, it implies 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].

Maximum likelihood estimation

First, let's consider the following two quantities that are defined for each mixture component k, $k = 1, \ldots, K$:

$$\gamma_k(\mathbf{M}_n, \theta) = \frac{\omega_k \times p(\mathbf{M}_n | \bar{\mathbf{M}}_k, \sigma_k)}{\sum_{j=1}^{K} \omega_j \times p(\mathbf{M}_n | \bar{\mathbf{M}}_j, \sigma_j)} \tag{15}$$

and

$$n_k(\theta) = \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n, \theta). \tag{16}$$

Then, the estimated parameters $\hat{\theta} = \{(\hat{\omega}_k, \widehat{\mathbf{M}}_k, \hat{\sigma}_k)_{1 \leq k \leq K}\}$ are iteratively updated based on the current value of $\hat{\theta}$:

- The estimated mixture weight $\hat{\omega}_k$ is given by:

$$\hat{\omega}_k = \frac{n_k(\hat{\theta})}{\sum_{k=1}^{K} n_k(\hat{\theta})}; \tag{17}$$

- The estimated central value $\widehat{\mathbf{M}}_k$ is computed as:

$$\widehat{\mathbf{M}}_k = \arg\min_{\mathbf{M}} \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n, \hat{\theta}) d^2(\mathbf{M}, \mathbf{M}_n); \tag{18}$$

In practice, (18) is solved by means of a gradient descent algorithm [54].

- The estimated dispersion $\hat{\sigma}_k$ is obtained as:

$$\hat{\sigma}_k = \Phi\left(n_k^{-1}(\theta) \times \sum_{n=1}^{N} \omega_k(\mathbf{M}_n, \hat{\theta}) d^2(\widehat{\mathbf{M}}_k, \mathbf{M}_n) \right), \tag{19}$$

where Φ is the inverse function of $\sigma \mapsto \sigma^3 \times \frac{d}{d\sigma} \log Z(\sigma)$.

Practically, the estimation procedure is repeated for a fixed number of iterations, or until convergence, that is until the estimated parameters remain almost stable for successive iterations. Moreover, as the estimation with the EM algorithm depends on the initial parameter setting, the EM algorithm is run several times (10 in practice) and the best result is kept (*i.e.* the one maximizing the log-likelihood criterion).

Based on the extracted (LE 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) [2,3] and the Fisher vectors (FV) [5–7]. 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) [40].

4. 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. 2 draws an overview of the relation between the different approaches

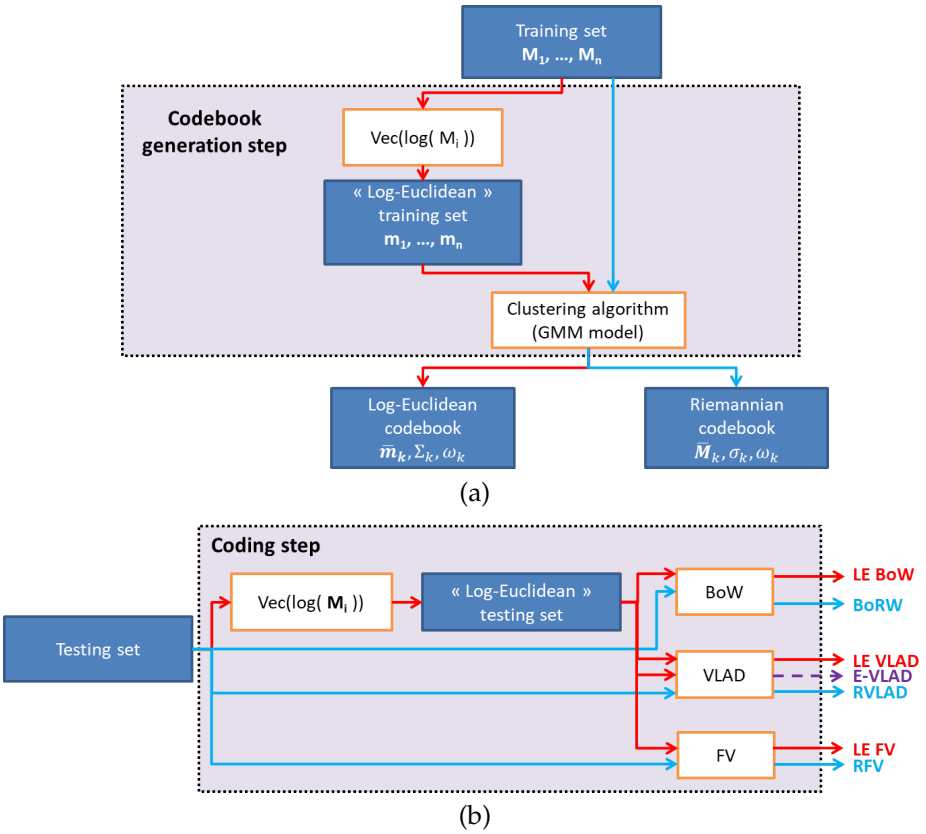

Figure 2. Workflow explaining (a) the codebook creation step and (b) the coding step. The LE based approaches appear in red while the Riemannian based ones are displayed in blue. The E-VLAD descriptor is displayed in purple since it considers simultaneously a Riemannian codebook and LE vector representation of the covariance matrices.

based on the BoW, VLAD and FV models. The LE-based metric approaches appear in red while the affine invariant ones are displayed in blue. The E-VLAD descriptor is displayed in purple since it considers the Riemannian codebook combined with LE representation of the features.

4.1. 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 [10,55], 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 [56], 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 LE and affine invariant Riemannian metrics:

- the log-Euclidean bag of words (LE BoW) [33,35].
- the bag of Riemannian words (BoRW) [36].

These two descriptors have been employed successfully for different applications, including texture and human epithelial type 2 cells classification [36], action recognition [33,35].

4.1.1. Log-Euclidean bag of words (LE BoW)

The LE BoW model has been considered in [33,35]. 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 3.1. 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:

$$\arg\max_k \varpi_k \; p(\mathbf{m}_n | \bar{\mathbf{m}}_k, \Sigma_k), \tag{20}$$

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:
$$\Sigma_k = \Sigma. \tag{21}$$
- the same weight is given to all mixture components:
$$\varpi_k = \frac{1}{K}. \tag{22}$$

4.1.2. Bag of Riemannian words (BoRW)

This descriptor has been introduced in [36]. 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 3.2. 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 (8).

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 defined in (12) 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:

$$\arg\max_k \varpi_k \; p(\mathbf{M}_n | \bar{\mathbf{M}}_k, \sigma_k). \tag{23}$$

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

It has been shown in the literature that the performance of BoW descriptors depends on the codebook size, best results being generally obtained for large dictionaries [5]. Moreover, 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.

4.2. Vectors of locally aggregated descriptors

VLAD descriptors have been introduced in [2] and represent a method of encoding the difference between the codewords and the features. For features lying in a Euclidean space, the 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 samples assigned to it are computed for each cluster:

$$\mathbf{v}_k = \sum_{\mathbf{x}_n \in c_k} \mathbf{x}_n - \bar{\mathbf{x}}_k. \tag{24}$$

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

$$\mathbf{VLAD} = [\mathbf{v}_1^T, \ldots, \mathbf{v}_K^T]. \tag{25}$$

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) [37].
- the intrinsic Riemannian vector of locally aggregated descriptors (RVLAD) [11].

4.2.1. 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, (24) is rewritten as:

$$\mathbf{v}_k = \sum_{\mathbf{m}_n \in c_k} \mathbf{m}_n - \bar{\mathbf{m}}_k, \tag{26}$$

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 LE codebook. The proximity is measured here according to the Euclidean distance between the LE vectors.

4.2.2. 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. It yields that:

$$\mathbf{v}_k = \sum_{\mathbf{M}_n \in c_k} \mathbf{m}_n - \bar{\mathbf{m}}_k, \tag{27}$$

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 [37] to replace the affine invariant Riemannian metric by the Stein metric [57]. 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 since a recursive computation of the Stein center from a set of covariance matrices has been proposed in [58].

Since this approach exploits two metrics, one for the codebook creation (with the affine invariant Riemannian or Stein metric) and another for the coding step (with the LE metric), we referred it as an extrinsic method.

4.2.3. 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 [47] recalled in (8) is considered to measure similarity between SPD matrices. The affine invariant Riemannian metric is used to define the Voronoï

regions) and the Riemannian logarithm mapping [48] is used to perform the subtraction on the manifold. It yields that for the RVLAD model, the vectors $\mathbf{v}_k$ are obtained as:

$$\mathbf{v}_k = \text{Vec}\left(\sum_{\mathbf{M}_n \in c_k} \text{Log}_{\bar{\mathbf{M}}_k}(\mathbf{M}_n)\right), \tag{28}$$

where $\text{Log}_{\bar{\mathbf{M}}_k}(\cdot)$ is the Riemannian logarithm mapping defined in (11). Note that the vectorization operator $\text{Vec}(\cdot)$ is used to represent $\mathbf{v}_k$ as a vector.

As explained in [2], 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.

4.3. Fisher vector descriptor

Fisher vectors (FV) are descriptors based on Fisher kernels [59]. 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 Fisher information matrix (FIM).

First, the gradient of the log-likelihood with respect to the model parameter vector θ, also known as the Fisher score (FS) $U_{\boldsymbol{\mathcal{X}}}$ [59], should be computed:

$$U_{\boldsymbol{\mathcal{X}}} = \nabla_\theta \log p(\boldsymbol{\mathcal{X}}|\theta) = \nabla_\theta \sum_{n=1}^{N} \log p(\mathbf{X}_n|\theta). \tag{29}$$

As mentioned in [5], 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 [59]. 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 [59]:

$$I_\theta = E_{\boldsymbol{\mathcal{X}}}[U_{\boldsymbol{\mathcal{X}}} U_{\boldsymbol{\mathcal{X}}}^T], \tag{30}$$

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 [5]:

$$\mathcal{G}_\theta^{\boldsymbol{\mathcal{X}}} = I_\theta^{-1/2} \nabla_\theta \log p(\boldsymbol{\mathcal{X}}|\theta). \tag{31}$$

As reported in previous works, exploiting the FIM I_θ in the derivation of FV yields to excellent results with linear classifiers [6,7,9]. However, the computation of the 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 [5,9].

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

- the Log-Euclidean Fisher vectors (LE FV).
- the Riemannian Fisher vectors (RFV) [40].

4.3.1. Log-Euclidean Fisher vectors (LE FV)

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 descriptor 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 k^{th} Gaussian component

$$\gamma_k(\mathbf{m}_n) = \frac{\varpi_k \; p(\mathbf{m}_n|\theta_k)}{\sum_{j=1}^{K} \varpi_j \; p(\mathbf{m}_n|\theta_j)}. \tag{32}$$

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

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}_{LE}|\theta)}{\partial \bar{\mathbf{m}}_k^d} = \sum_{n=1}^{N} \gamma_k(\mathbf{m}_n) \left(\frac{\mathbf{m}_n^d - \bar{\mathbf{m}}_k^d}{\left(\sigma_k^d\right)^2} \right), \tag{33}$$

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}_{LE}|\theta)}{\partial \sigma_k^d} = \sum_{n=1}^{N} \gamma_k(\mathbf{m}_n) \left(\frac{\left[\mathbf{m}_n^d - \bar{\mathbf{m}}_k^d\right]^2}{\left(\sigma_k^d\right)^3} - \frac{1}{\sigma_k^d} \right), \tag{34}$$

$$\frac{\partial \log p(\boldsymbol{\mathcal{M}}_{LE}|\theta)}{\partial \alpha_k} = \sum_{n=1}^{N} \Big(\gamma_k(\mathbf{m}_n) - \varpi_k \Big), \tag{35}$$

where $\bar{\mathbf{m}}_k^d$ (resp. σ_k^d) is the d^{th} 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 [9,60]:

$$\varpi_k = \frac{\exp(\alpha_k)}{\sum_{j=1}^{K} \exp(\alpha_j)}. \tag{36}$$

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 [9]. It yields that the LE FV of $\mathcal{M}$ is obtained as:

$$\mathscr{G}_{\bar{\mathbf{m}}_k^d}^{\boldsymbol{\mathcal{M}}_{LE}} = \frac{1}{\sqrt{\varpi_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{m}_n) \left(\frac{\mathbf{m}_n^d - \bar{\mathbf{m}}_k^d}{\sigma_k^d} \right), \tag{37}$$

$$\mathscr{G}_{\sigma_k^d}^{\boldsymbol{\mathcal{M}}_{LE}} = \frac{1}{\sqrt{2\varpi_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{m}_n) \left(\frac{\left[\mathbf{m}_n^d - \bar{\mathbf{m}}_k^d\right]^2}{\left(\sigma_k^d\right)^2} - 1 \right), \tag{38}$$

$$\mathscr{G}_{\alpha_k}^{\boldsymbol{\mathcal{M}}_{LE}} = \frac{1}{\sqrt{\varpi_k}} \sum_{n=1}^{N} \Big(\gamma_k(\mathbf{m}_n) - \varpi_k \Big). \tag{39}$$

4.3.2. Riemannian Fisher vectors (RFV)

Ilea *et al.* have proposed in [40] 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 [39] as presented in Section 3.2.2. 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 $\theta = \{(\omega_k, \bar{\mathbf{M}}_k, \sigma_k)_{1 \leq k \leq K}\}$. It yields that [40]:

$$\frac{\partial \log p(\mathcal{M}|\theta)}{\partial \bar{\mathbf{M}}_k} = \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \, \frac{\mathrm{Log}_{\bar{\mathbf{M}}_k}(\mathbf{M}_n)}{\sigma_k^2}, \tag{40}$$

$$\frac{\partial \log p(\mathcal{M}|\theta)}{\partial \sigma_k} = \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \left\{ \frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{\sigma_k^3} - \frac{Z'(\sigma_k)}{Z(\sigma_k)} \right\}, \tag{41}$$

$$\frac{\partial \log p(\mathcal{M}|\theta)}{\partial \alpha_k} = \sum_{n=1}^{N} \left[\gamma_k(\mathbf{M}_n) - \omega_k \right], \tag{42}$$

where $\mathrm{Log}_{\bar{\mathbf{M}}_k}(\cdot)$ is the Riemannian logarithm mapping in (11) 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 3.2.2).

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

$$\gamma_k(\mathbf{M}_n) = \frac{\omega_k \; p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k)}{\sum_{j=1}^{K} \omega_j \; p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}. \tag{43}$$

By comparing (33), (34), (35) with (40), (41), (42), one can directly notice the similarity between the LE FS and the RFS. In these equations, vector difference in the LE FS is replaced by log map function in the RFS. Similarly, Euclidean distance in the LE FS is replaced by geodesic distance in the RFS.

In [40], 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 4.3.1 should be made, *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 [61]. In this paper, Zanini *et al.* have used the 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.

First, let's recall some elements regarding the derivation of the FIM. This block diagonal matrix is composed of three terms, one for the weight, one for the centroid and one for the dispersion.

- For the weight term, the same procedure as the one used in the conventional Euclidean framework can employed [9]. In [61], they proposed another way to derive this term by using the notation $\mathbf{s} = [\sqrt{\omega_1}, \; \ldots, \sqrt{\omega_K}]$ and observing that $\mathbf{s}$ belongs to a Riemannian manifold (more precisely the $(K-1)$-sphere $\mathbb{S}^{K-1}$). These two approaches yield exactly to the same final result.
- For the centroid term, it should be noted that each centroid $\bar{\mathbf{M}}_k$ is a covariance matrix which lives in the manifold $\mathcal{P}_m$ of $m \times m$ symmetric positive definite matrices. To derive the FIM associated to this term, the space $\mathcal{P}_m$ should be decomposed as the product of two irreducible manifolds, *i.e.* $\mathcal{P}_m = \mathbb{R} \times \mathcal{SP}_m$ where $\mathcal{SP}_m$ is the manifold of symmetric positive definite matrices with unitary determinant. Hence, each observed covariance matrix $\mathbf{M}$ can be decomposed as $\phi(\mathbf{M}) = \{(\mathbf{M})_1, (\mathbf{M})_2\}$ where
  - $(\mathbf{M})_1 = \log \det \mathbf{M}$ is a scalar element lying in $\mathbb{R}$.
  - $(\mathbf{M})_2 = e^{-\frac{(\mathbf{M})_1}{m}} \mathbf{M}$ is a covariance matrix of unit determinant.
- For the dispersion parameter, the notation $\eta = -\frac{1}{2\sigma^2}$ is considered to ease the mathematical derivation. Since this parameter is real, the conventional Euclidean framework is employed to derive the FIM. The only difference is that the Euclidean distance is replaced by the geodesic one.

For more information on the derivation of the FIM for the Riemannian Gaussian mixture model, the interested reader is referred to [61]. To summarize, the elements of the block-diagonal FIM for the Riemannian Gaussian mixture model are defined by:

$$I_s = 4\mathbf{I}_K, \tag{44}$$

$$I_{(\bar{\mathbf{M}}_k)_1} = \frac{\omega_k}{\sigma_k^3}, \tag{45}$$

$$I_{(\bar{\mathbf{M}}_k)_2} = \frac{\omega_k \ \psi_2^{'}(\eta_k)}{\sigma_k^4 \left(\frac{m(m+1)}{2} - 1\right)} \mathbf{I}_{\frac{m(m+1)}{2}-1}, \tag{46}$$

$$I_{\eta_k} = \omega_k \ \psi^{''}(\eta_k), \tag{47}$$

where $\mathbf{I}_K$ is the $K \times K$ identity matrix, $\psi(\eta) = \log\left(Z(\sigma)\right)$ and $\psi^{'}(\cdot)$ (resp. $\psi^{''}(\cdot)$) are the first (resp. the second) order derivatives of the $\psi(\cdot)$ function with respect to η. $\psi_2^{'}(\eta) = \psi^{'}(\eta) + \frac{1}{2\eta}$.

Now that the FIM and the FS score are obtained for the Riemannian Gaussian mixture model, we can define the RFV by combining (40) to (42) and (44) to (47) in (31). It yields that:

$$\mathscr{G}^{\mathcal{M}}_{(\bar{\mathbf{M}}_k)_1} = \frac{1}{\sqrt{\omega_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \left(\frac{(\bar{\mathbf{M}}_k)_1 - (\mathbf{M}_n)_1}{\sigma_k} \right), \tag{48}$$

$$\mathscr{G}^{\mathcal{M}}_{(\bar{\mathbf{M}}_k)_2} = \frac{1}{\sqrt{\omega_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \ \frac{\sqrt{\frac{m(m+1)}{2} - 1}}{\psi_2^{'}(\eta_k)} \ \mathrm{Log}_{(\bar{\mathbf{M}}_k)_2}\Big((\mathbf{M}_n)_2\Big), \tag{49}$$

$$\mathscr{G}^{\mathcal{M}}_{\sigma_k} = \frac{1}{\sqrt{\omega_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \left(\frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k) - \psi^{'}(\eta_k)}{\sqrt{\psi^{''}(\eta_k)}} \right), \tag{50}$$

$$\mathscr{G}^{\mathcal{M}}_{\omega_k} = \frac{1}{\sqrt{\omega_k}} \sum_{n=1}^{N} \Big(\gamma_k(\mathbf{M}_n) - \omega_k \Big). \tag{51}$$

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 (37), (38) and (39) are retrieved when the normalization factor $Z(\sigma)$ is set to $\sigma\sqrt{2\pi}$ in (48), (50) and (51).

In the end, the RFVs are obtained by concatenating some, or all of the derivatives in (48), (49), (50) and (51). Note also that since (49) is a matrix, the vectorization operator $\mathrm{Vec}(\cdot)$ is used to represent it as a vector.

4.3.3. 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:

- the hard assignment scheme, that is:

$$\gamma_k(\mathbf{M}) = \begin{cases} 1, & \text{if } \mathbf{M} \in c_k \\ 0, & \text{otherwise,} \end{cases} \tag{52}$$

  where $\mathbf{M} \in c_k$ are the elements assigned to cluster c_k and $k = 1, \ldots, K$,
- the homoscedasticity assumption, that is $\sigma_k = \sigma \ , \forall k = 1, \ldots, K$.

By taking into account these hypotheses, it can be noticed that (33) reduces to (26), 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).

4.4. Post-processing

Once the set of SPD matrices is encoded by one of the previously exposed coding methods (BoW, VLAD, FS 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. These operations are detailed next.

4.4.1. Power normalization

The purpose of this normalization method is to correct the independence assumption that is usually made on the image patches [7]. For the same vector $\mathbf{v}$, its power-normalized version $\mathbf{v}_{power}$ is obtained as:

$$\mathbf{v}_{power} = \text{sign}(\mathbf{v})|\mathbf{v}|^{\rho}, \tag{53}$$

where $0 < \rho \leq 1$, and $\text{sign}(\cdot)$ is the signum function and $|\cdot|$ is the absolute value. In practice, ρ is set to $\frac{1}{2}$, as suggested in [9].

4.4.2. ℓ_2 normalization

This normalization method has been proposed in [6] to minimize the influence of the background information on the image signature. For a vector $\mathbf{v}$, its normalized version $\mathbf{v}_{L_2}$ is computed as:

$$\mathbf{v}_{L_2} = \frac{\mathbf{v}}{\|\mathbf{v}\|_2}, \tag{54}$$

where $\|\cdot\|_2$ is the L_2 norm.

Depending on the considered coding method, one or both normalization steps are applied. For instance, for VLAD, FS and FV based methods, both normalizations are used [36,40], while for BoW based methods only the ℓ_2 normalization is considered [33].

4.5. Synthesis

Table 1 draws an overview of the different coding methods. As seen before, two metrics can be considered, namely the LE and the affine invariant Riemannian metrics. This yields to two Gaussian mixture models: a mixture of multivariate Gaussian distributions and a mixture of Riemannian Gaussian distributions. These mixture models are the central point in the computation of the codebook which are further used to encode the features. In this table and in the following ones, the proposed coding methods are displayed in gray.

As observed, a direct parallel can be drawn between the different coding methods (BoW, VLAD, FS and FV). More precisely, it is interesting to note how the conventional coding methods used for descriptors lying in $\mathbb{R}^{\frac{m(m+1)}{2}}$ are adapted to covariance matrix descriptors.

5. Application to image classification

This section introduces some applications to image classification. Two experiments are conducted, one for texture image classification and one for head pose image classification. The aim of these experiments 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.

Table 1. Overview of the coding descriptors.

	Log-Euclidean metric	**Affine invariant Riemannian metric**
Mixture model		
Gaussian mixture model	Mixture of multivariate Gaussian distributions $p(\mathbf{m}_n\|\theta) = \sum_{k=1}^{K} \varpi_k p(\mathbf{m}_n\|\bar{\mathbf{m}}_k, \Sigma_k)$ with $\bar{\mathbf{m}}_k \in \mathbb{R}^{\frac{m(m+1)}{2}}$, $\sigma_k^2 = \text{diag}(\Sigma_k) \in \mathbb{R}^{\frac{m(m+1)}{2}}$ and $\varpi_k \in \mathbb{R}$.	Mixture of Riemannian Gaussian distributions [39,42] $p(\mathbf{M}_n\|\theta) = \sum_{k=1}^{K} \varpi_k p(\mathbf{M}_n\|\bar{\mathbf{M}}_k, \sigma_k)$ with $\bar{\mathbf{M}}_k \in \mathcal{P}_m$, $\sigma_k \in \mathbb{R}$ and $\varpi_k \in \mathbb{R}$.
Coding method		
Bag of Words (BoW)	Log-Euclidean BoW (LE BoW) [33,35] Histogram based on the decision rule $\arg\max_k \varpi_k \; p(\mathbf{m}_n\|\bar{\mathbf{m}}_k, \Sigma_k)$	Bag of Riemannian Words (BoRW) [36] Histogram based on the decision rule $\arg\max_k \varpi_k \; p(\mathbf{M}_n\|\bar{\mathbf{M}}_k, \sigma_k)$
Vector of Locally Aggregated Descriptors (VLAD)	Log-Euclidean VLAD (LE VLAD) [11] $\mathbf{v}_k = \sum_{\mathbf{m}_n \in c_k} \mathbf{m}_n - \bar{\mathbf{m}}_k$	Riemannian VLAD (RVLAD) [11] $\mathbf{v}_k = \text{Vec}\left(\sum_{\mathbf{M}_n \in c_k} \text{Log}_{\bar{\mathbf{M}}_k}(\mathbf{M}_n)\right)$
	Extrinsic VLAD (E-VLAD) [37] $\mathbf{v}_k = \sum_{\mathbf{M}_n \in c_k} \mathbf{m}_n - \bar{\mathbf{m}}_k$	
Fisher Score (FS)	Log-Euclidean Fisher Score (LE FS) $\frac{\partial \log p(\boldsymbol{\mathcal{M}}_{LE}\|\theta)}{\partial \bar{\mathbf{m}}_k^d} = \sum_{n=1}^{N} \gamma_k(\mathbf{m}_n) \left(\frac{\mathbf{m}_n^d - \bar{\mathbf{m}}_k^d}{(\sigma_k^d)^2} \right)$ $\frac{\partial \log p(\boldsymbol{\mathcal{M}}_{LE}\|\theta)}{\partial \sigma_k^d} = \sum_{n=1}^{N} \gamma_k(\mathbf{m}_n) \left(\frac{[\mathbf{m}_n^d - \bar{\mathbf{m}}_k^d]^2}{(\sigma_k^d)^3} - \frac{1}{\sigma_k^d} \right)$ $\frac{\partial \log p(\boldsymbol{\mathcal{M}}_{LE}\|\theta)}{\partial \varpi_k} = \sum_{n=1}^{N} \left(\gamma_k(\mathbf{m}_n) - \varpi_k \right)$	Riemannian Fisher Score (RFS) [40] $\frac{\partial \log p(\boldsymbol{\mathcal{M}}\|\theta)}{\partial \bar{\mathbf{M}}_k} = \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \; \frac{\text{Log}_{\bar{\mathbf{M}}_k}(\mathbf{M}_n)}{\sigma_k^2}$ $\frac{\partial \log p(\boldsymbol{\mathcal{M}}\|\theta)}{\partial \sigma_k} = \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \left\{ \frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k)}{\sigma_k^3} - \frac{Z'(\sigma_k)}{Z(\sigma_k)} \right\}$ $\frac{\partial \log p(\boldsymbol{\mathcal{M}}\|\theta)}{\partial \varpi_k} = \sum_{n=1}^{N} \left[\gamma_k(\mathbf{M}_n) - \varpi_k \right]$
Fisher Vector (FV)	Log-Euclidean Fisher Vectors (LE FV) $\mathscr{G}^{\boldsymbol{\mathcal{M}}_{LE}}_{\bar{\mathbf{m}}_k^d} = \frac{1}{\sqrt{\varpi_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{m}_n) \left(\frac{\mathbf{m}_n^d - \bar{\mathbf{m}}_k^d}{\sigma_k^d} \right)$ $\mathscr{G}^{\boldsymbol{\mathcal{M}}_{LE}}_{\sigma_k^d} = \frac{1}{\sqrt{2\varpi_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{m}_n) \left(\frac{[\mathbf{m}_n^d - \bar{\mathbf{m}}_k^d]^2}{(\sigma_k^d)^2} - 1 \right)$ $\mathscr{G}^{\boldsymbol{\mathcal{M}}_{LE}}_{\alpha_k} = \frac{1}{\sqrt{\varpi_k}} \sum_{n=1}^{N} \left(\gamma_k(\mathbf{m}_n) - \varpi_k \right)$	Riemannian Fisher Vectors (RFV) $\mathscr{G}^{\boldsymbol{\mathcal{M}}}_{(\bar{\mathbf{M}}_k)_1} = \frac{1}{\sqrt{\varpi_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \left(\frac{(\bar{\mathbf{M}}_k)_1 - (\mathbf{M}_n)_1}{\sigma_k} \right)$ $\mathscr{G}^{\boldsymbol{\mathcal{M}}}_{(\bar{\mathbf{M}}_k)_2} = \frac{1}{\sqrt{\varpi_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \; \frac{\sqrt{\frac{m(m+1)}{2} - 1}}{\psi'(\eta_k)} \; \text{Log}_{(\bar{\mathbf{M}}_k)_2}\left((\mathbf{M}_n)_2\right)$ $\mathscr{G}^{\boldsymbol{\mathcal{M}}}_{\sigma_k} = \frac{1}{\sqrt{\varpi_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \left(\frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k) - \psi'(\eta_k)}{\sqrt{\psi''(\eta_k)}} \right)$ $\mathscr{G}^{\boldsymbol{\mathcal{M}}}_{\varpi_k} = \frac{1}{\sqrt{\varpi_k}} \sum_{n=1}^{N} \left(\gamma_k(\mathbf{M}_n) - \varpi_k \right)$

5.1. Texture image classification

5.1.1. Image databases

To answer these questions, a first experiment is conducted on four conventional texture databases, namely the VisTex [62], Brodatz [63], Outex-TC-00013 [64] and USPtex [65] databases. Some examples of texture images issued from these four texture databases are displayed in Fig. 3

The VisTex database is composed of 40 texture images of size 512×512 pixels. In the following, each texture image is divided into 64 non-overlapping images of size 64×64 pixels, yielding to a database of 2560 images. The grayscale Brodatz database contains 112 textures images of size 640×640 pixels which represent a large variety of natural textures. Each one is divided into 25 non-overlapping images of size 128×128 pixels, thus creating 2800 images in total (i.e., 112 classes with 25 images/class).

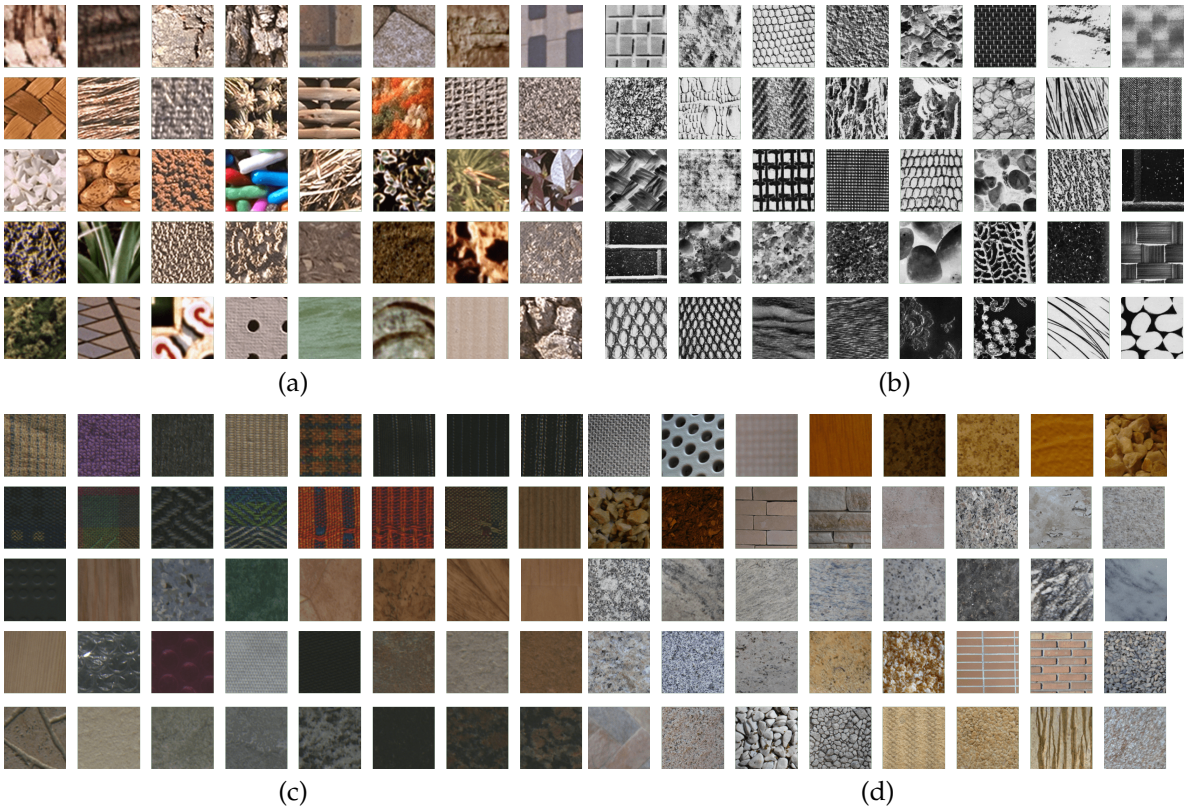

Figure 3. Examples of texture images used in the experimental study issued from the (a) VisTex, (b) Brodatz, (c) Outex and (d) USPtex texture databases.

Table 2. Description of the texture databases used in this experiment.

Database	Number of classes	Number of images per class	Total number of images	Dimension
VisTex	40	64	2560	64× 64 pixels
Brodatz	112	25	2800	128× 128 pixels
Outex	68	20	1380	128× 128 pixels
USPtex	191	12	2292	128× 128 pixels

The Outex database consists of a dataset of 68 texture classes (canvas, stone, wood, ...) with 20 image samples per class of size 128 × 128 pixels. And finally, The USPtex database is composed of 191 texture classes with 12 image samples of size 128 × 128 pixels. Table. 2 summarizes the main characteristics of each of these four databases.

5.1.2. Context

As shown in Fig. 1, 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 [34]. The extracted RCovD are the estimated covariance matrices of vectors $\mathbf{v}(x,y)$ computed on sliding patches of size 15 × 15 pixels where:

$$\mathbf{v}(x,y) = \left[I(x,y), \left|\frac{\partial I(x,y)}{\partial x}\right|, \left|\frac{\partial I(x,y)}{\partial y}\right|, \left|\frac{\partial^2 I(x,y)}{\partial x^2}\right|, \left|\frac{\partial^2 I(x,y)}{\partial y^2}\right| \right]^T. \tag{55}$$

In this experiment, the patches are overlapped by 50%. The fast covariance matrix computation algorithm based on integral images presented in [34] is adopted to speed-up the computation time of

Table 3. Classification results on the VisTex database (40 classes).

Coding method	Log-Euclidean metric	Affine invariant Riemannian metric
LE BoW [35] / BoRW [36]	86.4 ± 0.01	85.9 ± 0.01
LE VLAD [11] / RVLAD [11]	91.3 ± 0.1	82.8 ± 0.02
E-VLAD [37]	91.6 ± 0.01	
LE FS / RFS [40]: $\bar{\mathbf{M}}$	95.3 ± 0.01	88.9 ± 0.01
LE FS / RFS [40]: $\bar{\mathbf{M}}, \omega$	95.1 ± 0.01	90.0 ± 0.01
LE FS / RFS [40]: $\bar{\mathbf{M}}, \sigma$	95.2 ± 0.01	91.2 ± 0.01
LE FS / RFS [40]: $\bar{\mathbf{M}}, \sigma, \omega$	95.1 ± 0.01	91.2 ± 0.01
LE FV / RFV: $\bar{\mathbf{M}}$	95.5 ± 0.01	91.3 ± 0.01
LE FV / RFV: $\bar{\mathbf{M}}, \omega$	$\mathbf{95.7 \pm 0.01}$	92.6 ± 0.01
LE FV / RFV: $\bar{\mathbf{M}}, \sigma$	95.6 ± 0.01	92.7 ± 0.01
LE FV / RFV: $\bar{\mathbf{M}}, \sigma, \omega$	95.4 ± 0.01	93.2 ± 0.01

this feature extraction step. It yields that each texture class is composed by a set $\{\mathbf{M}_1, \ldots, \mathbf{M}_N\}$ 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). Note that the same number of modes K has been considered for each class and has been set experimentally to 3 which represents a good trade-off between the model complexity and the within-class diversity. This parameter has been fixed for all these experiments since the aim is to fairly compare the different coding strategies for the same codebook.

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 LE 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 3 to 6 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, different versions of the FS and FV descriptors can be considered to analyze separately the contribution of each term or by combining these different terms. For example, the row "LE FS / RFS: $\bar{\mathbf{M}}$" indicates the classification results when only the derivatives with respect to the centroid are considered to derive the FS (see (33) and (40)). In the following, only the results employing the mean are presented since the state-of-the-art have already proved that the mean provides the most significant information [6,7].

Note 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 "LE FV / RFV: $\bar{\mathbf{M}}$" with "LE FS / RFS: $\bar{\mathbf{M}}$".

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.

5.1.3. Comparison between anisotropic and isotropic models

As observed in Tables 3 to 6, the performance for the LE metric are generally better than that with the affine invariant Riemannian metric. But, both approaches are not directly comparable since

Table 4. Classification results on the Brodatz database (112 classes).

Coding method	Log-Euclidean metric	Affine invariant Riemannian metric
LE BoW [35] / BoRW [36]	92.0 ± 0.01	92.1 ± 0.01
LE VLAD [11] / RVLAD [11]	92.5 ± 0.01	88.3 ± 0.01
E-VLAD [37]	92.4 ± 0.01	
LE FS / RFS [40]: $\bar{\mathbf{M}}$	92.5 ± 0.01	90.1 ± 0.01
LE FS / RFS [40]: $\bar{\mathbf{M}}, \omega$	92.7 ± 0.01	91.1 ± 0.01
LE FS / RFS [40]: $\bar{\mathbf{M}}, \sigma$	90.3 ± 0.01	91.7 ± 0.01
LE FS / RFS [40]: $\bar{\mathbf{M}}, \sigma, \omega$	90.8 ± 0.03	91.6 ± 0.01
LE FV / RFV: $\bar{\mathbf{M}}$	93.5 ± 0.01	92.9 ± 0.01
LE FV / RFV: $\bar{\mathbf{M}}, \omega$	$\mathbf{93.7 \pm 0.01}$	93.2 ± 0.01
LE FV / RFV: $\bar{\mathbf{M}}, \sigma$	93.1 ± 0.01	93.1 ± 0.01
LE FV / RFV: $\bar{\mathbf{M}}, \sigma, \omega$	92.9 ± 0.01	93.2 ± 0.01

Table 5. Classification results on the Outex database (68 classes).

Coding method	Log-Euclidean metric	Affine invariant Riemannian metric
LE BoW [35] / BoRW [36]	83.5 ± 0.01	83.7 ± 0.01
LE VLAD [11] / RVLAD [11]	85.9 ± 0.01	82.0 ± 0.01
E-VLAD [37]	85.1 ± 0.01	
LE FS / RFS [40]: $\bar{\mathbf{M}}$	87.2 ± 0.01	83.8 ± 0.01
LE FS / RFS [40]: $\bar{\mathbf{M}}, \omega$	$\mathbf{88.0 \pm 0.01}$	84.2 ± 0.01
LE FS / RFS [40]: $\bar{\mathbf{M}}, \sigma$	86.7 ± 0.01	84.9 ± 0.01
LE FS / RFS [40]: $\bar{\mathbf{M}}, \sigma, \omega$	87.6 ± 0.01	85.2 ± 0.01
LE FV / RFV: $\bar{\mathbf{M}}$	87.3 ± 0.01	85.4 ± 0.01
LE FV / RFV: $\bar{\mathbf{M}}, \omega$	87.9 ± 0.01	86.0 ± 0.01
LE FV / RFV: $\bar{\mathbf{M}}, \sigma$	87.1 ± 0.01	86.0 ± 0.01
LE FV / RFV: $\bar{\mathbf{M}}, \sigma, \omega$	87.2 ± 0.01	86.3 ± 0.01

Table 6. Classification results on the USPtex database (191 classes).

Coding method	Log-Euclidean metric	Affine invariant Riemannian metric
LE BoW [35] / BoRW [36]	79.9 ± 0.01	80.2 ± 0.01
LE VLAD [11] / RVLAD [11]	86.5 ± 0.01	78.9 ± 0.01
E-VLAD [37]	86.7 ± 0.01	
LE FS / RFS [40]: $\bar{\mathbf{M}}$	84.8 ± 0.03	84.7 ± 0.01
LE FS / RFS [40]: $\bar{\mathbf{M}}, \omega$	85.1 ± 0.02	85.2 ± 0.01
LE FS / RFS [40]: $\bar{\mathbf{M}}, \sigma$	76.8 ± 0.03	84.0 ± 0.01
LE FS / RFS [40]: $\bar{\mathbf{M}}, \sigma, \omega$	77.9 ± 0.03	84.0 ± 0.01
LE FV / RFV: $\bar{\mathbf{M}}$	88.3 ± 0.01	87.0 ± 0.01
LE FV / RFV: $\bar{\mathbf{M}}, \omega$	88.0 ± 0.01	87.0 ± 0.01
LE FV / RFV: $\bar{\mathbf{M}}, \sigma$	87.7 ± 0.01	87.3 ± 0.01
LE FV / RFV: $\bar{\mathbf{M}}, \sigma, \omega$	$\mathbf{88.4 \pm 0.01}$	87.2 ± 0.01

an anisotropic model is considered for the LE metric while an isotropic model is used for the affine invariant Riemannian metric. Indeed, for the former the dispersion for the Gaussian mixture model is a diagonal matrix Σ_k while for the latter the dispersion σ_k is a scalar. In order to provide a fairer comparison between these two approaches, an experiment is conducted to illustrate if the observed gain with the LE metric comes from the metric or from the fact that the Gaussian model is anisotropic.

For the LE metric, an isotropic model can be built by considering that $\Sigma_k = \sigma_k^2 \, \mathbf{I}_{\frac{m(m+1)}{2}}$. For the affine invariant Riemannian metric, the Riemannian Gaussian distribution recalled in Section 3.2.2 is isotropic. Pennec has introduced in [66] an anisotropic Gaussian model, but for this latter the

Table 7. Comparison between anisotropic and isotropic models, classification results based on FV : $\bar{\mathbf{M}}$.

Database	Anisotropic model, Log-Euclidean metric	Isotropic model, Log-Euclidean metric	Isotropic model, Affine invariant Riemannian metric
VisTex	95.5 ± 0.01	88.7 ± 0.01	91.3 ± 0.01
Brodatz	93.5 ± 0.01	87.1 ± 0.01	92.9 ± 0.01
Outex	87.3 ± 0.01	83.2 ± 0.01	85.4 ± 0.01
USPtex	88.3 ± 0.01	81.5 ± 0.01	87.0 ± 0.01

normalization factor depends on both the centroid $\bar{\mathbf{M}}_k$ and the concentration matrix. It yields that the computation of the FS score and the derivation of the FIM for this model are still an open problem. This model will not be considered in the following.

Table 7 shows the classification results obtained on the four considered texture databases. Here, the performances are displayed for the FV descriptor computed by using the derivative with respect to the centroid (*i.e.* LE FV / RFV: $\bar{\mathbf{M}}$). It can be noticed that for the LE metric, an anisotropic model yields to a significant gain of about 4 to 7% compared to an isotropic model. More interestingly, for an istropic model, descriptors based on the affine invariant Riemannian metric yield to better performances than that obtained with the LE metric. A gain of about 2 to 6% is observed. These experiments clearly illustrate that the gain observed in Tables 3 to 6 for the LE metric comes better from the anistropicty of the Gaussian mixture model than from the metric definition. According to these observations, it is expected that classifying with FV issued from anistropic Riemannian Gaussian mixture model will improve the performance. This point will be subject of future research works including the derivation of normalization factor of the anistropic Riemannian Gaussian model and the computation of the FIM.

5.2. Head pose classification

5.2.1. Context

The aim of this second experiment is to illustrate how the proposed framework can be used for classifying a set of covariance matrices of larger dimension. Here, the head pose classification problem is investigated on the HOCoffee dataset [67]. This dataset contains 18 117 head images of size 50×50 pixels with six head pose classes (front left, front, front right, left, rear and right). Some examples of images of each class (one class per row) are displayed in Fig. 4. It has a predefined experiment protocol where 9 522 images are used for training and the remaining 8 595 images are used for testing. We follow the same experiment protocol as in [11]. The extracted RCovD are the estimated covariance matrices of vectors $\mathbf{v}(x,y)$ computed on sliding patches of size 15×15 pixels where:

$$\mathbf{v}(x,y) = \left[I_L(x,y), I_a(x,y), I_b(x,y), \sqrt{I_x^2(x,y) + I_y^2(x,y)}, \arctan\left(\frac{I_x(x,y)}{I_y(x,y)}\right), G_1(x,y), \ \ldots, G_8(x,y) \right]^T \tag{56}$$

with $I_c(x,y)$, $c \in \{L,a,b\}$ are the CIELab color information for the pixel at coordinate (x,y), $I_x(x,y)$ and $I_y(x,y)$ are the first order luminance derivatives, and $G_i(x,y)$ denotes the response of the i-th Difference Of Offset Gaussian (DOOG) filter-bank centered at position (x,y) of I_L. An overlap of 50% is considered to compute the covariance matrices. Hence, each image in the database is represented by a set of 25 covariance matrices of size 13×13. As for the previous experiment, 3 atoms per class are considered to compute the codebook.

Table 8 shows the classification accuracy on the HOCoffee dataset. Similar conclusions can be drawn with the previous experiment on texture image classification. The use of the FIM in the derivation of the FV still allows to improve the classification accuracy. The best performances are obtained for the LE metric compared to the affine invariant Riemannian metric. Nevertheless, for this

Figure 4. Examples of images from the HOCoffee dataset. It contains six head pose classes, from the first row to the last one (front left, front, front right, left, rear and right).

Table 8. Classification results on the HOCoffee database (6 classes).

Coding method	Log-Euclidean metric	Affine invariant Riemannian metric
LE BoW [35] / BoRW [36]	53.5	56.2
LE VLAD [11] / RVLAD [11]	79.1	70.6
E-VLAD [37]	79.3	
LE FS / RFS [40]: $\bar{\mathbf{M}}$	79.8	64.6
LE FS / RFS [40]: $\bar{\mathbf{M}}, \varpi$	79.8	65.0
LE FS / RFS [40]: $\bar{\mathbf{M}}, \sigma$	79.5	64.9
LE FS / RFS [40]: $\bar{\mathbf{M}}, \sigma, \varpi$	79.7	64.6
LE FV / RFV: $\bar{\mathbf{M}}$	**80.0**	67.7
LE FV / RFV: $\bar{\mathbf{M}}, \varpi$	79.9	67.5
LE FV / RFV: $\bar{\mathbf{M}}, \sigma$	79.7	67.9
LE FV / RFV: $\bar{\mathbf{M}}, \sigma, \varpi$	79.8	67.8

latter, the performance are quite low, especially for the FV obtained by deriving with respect to the dispersion parameter. Note that for this experiment the RVLAD descriptor allows to obtain better classification accuracy than the best RFV (70.6% vs. 67.9%).

In order to understand why the performance with RFV are relatively low for the HOCoffee dataset, an experiment is conducted to see if the dispersion parameter can be considered with confidence.

5.2.2. Estimation performance

This section presents simulation results to evaluate the performance of the estimator of the dispersion parameter for Gaussian models based on the LE and affine invariant Riemannian metrics. For all these experiments,

$$\bar{\mathbf{M}}_{ij} = \rho^{|i-j|} \text{ for } i, j \in [\![0, m-1]\!]. \tag{57}$$

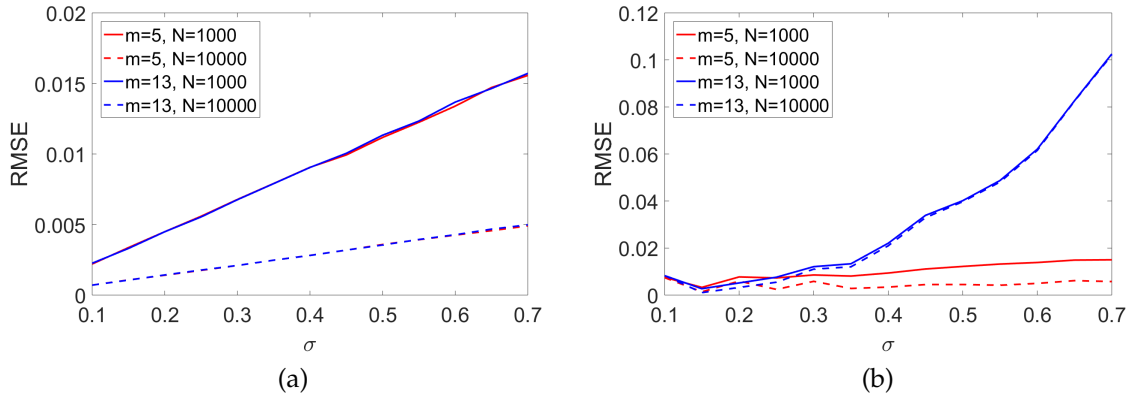

Figure 5. Root mean square error of the dispersion parameter for Gaussian models based on (a) the LE and (b) affine invariant Riemannian metrics.

ρ is set to 0.7 in the following. For the LE metric, N i.i.d. vector samples $(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ are generated according to a multivariate Gaussian distribution $\mathcal{N}(\bar{\mathbf{m}}, \Sigma)$, with $\Sigma = \sigma^2 \mathbf{I}_{\frac{m(m+1)}{2}}$. For the affine invariant Riemannian metric, N i.i.d. covariance matrix samples are generated according the Riemannian Gaussian distribution defined in Section 3.2.2. In the following, 1 000 Monte Carlo runs have been used to evaluate the performance of the estimation algorithm.

Fig. 5 draws the evolution of the root mean square error (RMSE) of the dispersion parameter σ for Gaussian models based on the LE and affine invariant Riemannian metrics as a function of σ. The red curve corresponds to an experiment with covariance matrices of dimension 5×5, while the blue one is for 13×13 covariance matrices. In this figure, 1 000 (resp. 10 000) covariance matrices samples issued from the Gaussian model are generated to plot the solid (resp. the dashed) curve. This yields that the texture classification experiment of Section 5 is mimicked with the solid red curve while the head pose classification experiment is mimicked with the dashed blue one. As observed in Fig. 5.(a) for the LE m etric, the RMSE of the dispersion parameter is mainly influenced by the number of generated samples N. For this LE metric, the dimension of the covariance matrices has less importance, since the red and blue curves are superposed. Nevertheless, for the affine invariant Riemannian metric in Fig. 5(b), the RMSE of the dispersion parameter is greatly influenced by the dimension of the covariance matrices, especially for large values of σ.

For the five databases, Fig. 6 shows the boxplots of the dispersion parameter for the LE (Fig. 6.(a)) and Riemannian (Fig. 6.(b)) codebooks. Note that since two different metrics are considered, the amplitude value of the dispersion parameter are not directly comparable between Fig. 6.(a) and Fig. 6.(b). But for a given metric, it is possible to analyze the variability of the dispersion parameter for the five experiments. As observed in Fig. 6.(b), the estimated dispersion parameter σ_k for the Riemannian codebook takes larger values for the HOCoffee dataset than that for the four texture datasets. For the former, the estimated dispersion parameters of the Riemannian codebook are larger than 0.4 which corresponds to the area in Fig. 5.(b) where the RMSE of σ increases greatly. This explains why the performance with the RFV (especially when the dispersion is considered) are relatively low compared to the LE FV. Indeed, as observed in Fig. 5.(a) for the LE codebook, the dispersion parameters are much more comparable for the five datasets and the dimension m of the observed covariance matrix has less impact on the RMSE of σ for the LE metric.

5.3. Computation time

The computation time can be separated in two parts:

- The first one concerns the time used in learning stage to generate the codebook.
- The second one concerns the time used to encode an image.

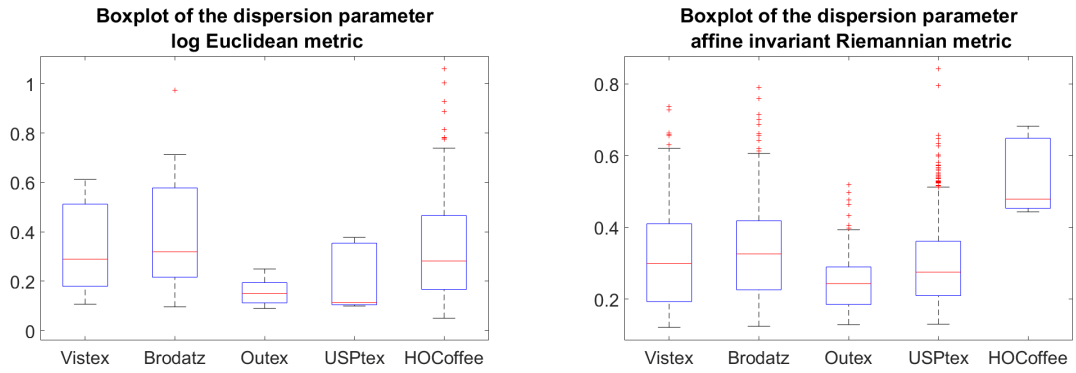

Figure 6. Boxplots of the dispersion parameter for the codebook computed with (a) the LE and (b) the affine invariant Riemannian metrics.

Obviously the codebook generation step requires much more time than the coding step. But this codebook generation step can be done offline. This is similar to a deep learning approach where the estimation of the model takes much more time than the classification itself. Table 9 summarizes these computation times for the experiment on the VisTex database. For the coding method, the LE FV and RFV descriptors with only the derivative with respect to the centroid $\bar{\mathbf{m}}$ or $\bar{\mathbf{M}}$ are considered. All the implementations are carried out using MATLAB 2017 on a PC machine Core i7-4790 3.6GHz, 16GB RAM.

Table 9. Computation time in seconds on the VisTex database.

Descriptor	Codebook creation	Coding time (per image)
LE FV	9s	0.077s
RFV	270s	0.476s

As expected, the LE metric allows to faster the computation time compared to the affine invariant Riemannian metric. A gain of a factor of 6 is observed for the coding time with the log-Euclidean metric for 5×5 covariance matrices.

6. Conclusion

Starting from the Gaussian mixture model (for the LE metric) and the Riemannian Gaussian mixture model (for the affine invariant Riemannian metric), we have proposed a unified view of coding methods. The proposed LE FV and RFV can be interpreted as a generalization of the BoW and VLAD based approaches. The experimental results have shown that: (i) the use of the FIM in the derivation of the FV allows to improve the classification accuracy, (ii) the proposed FV descriptors outperform the state-of-the-art BoW and VLAD based descriptors, and (iii) the descriptors based on the LE metric lead to better classification results than those based on the affine invariant Riemannian metric. For this latter observation, the gain observed with the LE metric comes better from the anistropicty of the Gaussian mixture model than on the metric itself. For isotropic models, FV described issued from the affine invariant Riemannian metric leads to better results than those obtained with the LE metric. It is hence expected that the definition of a FV issued from an anistropic Riemannian Gaussian mixture model will improve the performance. This point represents one of the main perspective of this research work.

For larger covariance matrices, the last experiment on head pose classification has illustrated the limits of the RFV issued from the Riemannian Gaussian mixture model. It has been shown that the

root mean square error of the dispersion parameter σ can be large for high value of σ ($\sigma > 0.4$). In that case, the LE FV are a good alternative to the RFV.

Future works will include the use of the proposed FV coding for covariance matrices descriptors in a hybrid classification architecture which will combine them with convolutional neural networks [17–19].

Author Contributions: All the authors contributed equally for the mathematical development and the specification of the algorithms. Ioana Ilea and Lionel Bombrun conducted the experiments and wrote the paper. Yannick Berthoumieu gave the central idea of the paper and managed the main tasks and experiments. All the authors read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

1. Sivic, J.; Russell, B.C.; Efros, A.A.; Zisserman, A.; Freeman, W.T. Discovering objects and their location in images. Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, 2005, Vol. 1, pp. 370–377 Vol. 1. doi:10.1109/ICCV.2005.77.
2. Jégou, H.; Douze, M.; Schmid, C.; Pérez, P. Aggregating local descriptors into a compact image representation. IEEE Conference on Computer Vision and Pattern Recognition, 2010.
3. Arandjelović, R.; Zisserman, A. All about VLAD. IEEE Conference on Computer Vision and Pattern Recognition, 2013.
4. Tsuda, K.; Kawanabe, M.; Müller, K.R. Clustering with the Fisher Score. Proceedings of the 15th International Conference on Neural Information Processing Systems; MIT Press: Cambridge, MA, USA, 2002; NIPS'02, pp. 745–752.
5. Perronnin, F.; Dance, C. Fisher kernels on visual vocabularies for image categorization. IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
6. Perronnin, F.; Sánchez, J.; Mensink, T., Improving the Fisher kernel for large-scale image classification. In *Computer Vision – ECCV 2010*; Daniilidis, K.; Maragos, P.; Paragios, N., Eds.; Springer Berlin Heidelberg, 2010; Vol. 6314, *Lecture Notes in Computer Science*, pp. 143–156. doi:10.1007/978-3-642-15561-1_11.
7. Perronnin, F.; Liu, Y.; Sánchez, J.; Poirier, H. Large-scale image retrieval with compressed Fisher vectors. The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010, 2010, pp. 3384–3391. doi:10.1109/CVPR.2010.5540009.
8. Douze, M.; Ramisa, A.; Schmid, C. Combining attributes and Fisher vectors for efficient image retrieval. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 745–752. doi:10.1109/CVPR.2011.5995595.
9. Sánchez, J.; Perronnin, F.; Mensink, T.; Verbeek, J. Image classification with the Fisher vector: Theory and practice. *International Journal of Computer Vision* **2013**, *105*, 222–245.
10. Salton, G.; Buckley, C. Term-weighting approaches in automatic text retrieval. *Information Processing and Management* **1988**, *24*, 513–523. doi:10.1016/0306-4573(88)90021-0.
11. Faraki, M.; Harandi, M.T.; Porikli, F. More about VLAD: A leap from Euclidean to Riemannian manifolds. IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4951–4960. doi:10.1109/CVPR.2015.7299129.
12. Le Cun, Y.; Boser, B.E.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.E.; Jackel, L.D. Handwritten Digit Recognition with a Back-Propagation Network. In *Advances in Neural Information Processing Systems 2*; Touretzky, D.S., Ed.; Morgan-Kaufmann, 1990; pp. 396–404.
13. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1; Curran Associates Inc.: USA, 2012; NIPS'12, pp. 1097–1105.
14. Chandrasekhar, V.; Lin, J.; Morère, O.; Goh, H.; Veillard, A. A Practical Guide to CNNs and Fisher Vectors for Image Instance Retrieval. *Signal Process.* **2016**, *128*, 426–439. doi:10.1016/j.sigpro.2016.05.021.
15. Perronnin, F.; Larlus, D. Fisher vectors meet Neural Networks: A hybrid classification architecture. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3743–3752. doi:10.1109/CVPR.2015.7298998.

16. Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep Fisher Networks for Large-scale Image Classification. Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1; Curran Associates Inc.: USA, 2013; NIPS'13, pp. 163–171.
17. Ng, J.Y.; Yang, F.; Davis, L.S. Exploiting Local Features from Deep Networks for Image Retrieval. *CoRR* **2015**, *abs/1504.05133*, [1504.05133].
18. Cimpoi, M.; Maji, S.; Kokkinos, I.; Vedaldi, A. Deep Filter Banks for Texture Recognition, Description, and Segmentation. *International Journal of Computer Vision* **2016**, *118*, 65–94. doi:10.1007/s11263-015-0872-3.
19. Diba, A.; Pazandeh, A.M.; Gool, L.V. Deep visual words: Improved fisher vector for image classification. 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA), 2017, pp. 186–189. doi:10.23919/MVA.2017.7986832.
20. Li, E.; Xia, J.; Du, P.; Lin, C.; Samat, A. Integrating Multilayer Features of Convolutional Neural Networks for Remote Sensing Scene Classification. *IEEE Transactions on Geoscience and Remote Sensing* **2017**, *55*, 5653–5665. doi:10.1109/TGRS.2017.2711275.
21. Ollila, E.; Koivunen, V. Robust antenna array processing using M-estimators of pseudo-covariance. 14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, 2003, Vol. 3, pp. 2659–2663. doi:10.1109/PIMRC.2003.1259213.
22. Greco, M.; Fortunati, S.; Gini, F. Maximum likelihood covariance matrix estimation for complex elliptically symmetric distributions under mismatched conditions. *Signal Processing* **2014**, *104*, 381–386. doi:10.1016/j.sigpro.2014.04.002.
23. Chen, Y.; Wiesel, A.; Hero, A.O. Robust shrinkage estimation of high-dimensional covariance matrices. *IEEE Transactions on Signal Processing* **2011**, *59*, 4097–4107. doi:10.1109/TSP.2011.2138698.
24. Yang, L.; Arnaudon, M.; Barbaresco, F. Riemannian median, geometry of covariance matrices and radar target detection. *European Radar Conference* **2010**, pp. 415–418.
25. Barbaresco, F.; Arnaudon, M.; Yang, L. Riemannian medians and means with applications to Radar signal processing. *IEEE Journal of Selected Topics in Signal Processing* **2013**, *7*, 595–604.
26. Garcia, G.; Oller, J.M. What does intrinsic mean in statistical estimation? *Statistics and Operations Research Transactions* **2006**, *30*, 125–170.
27. de Luis-García, R.; Westin, C.F.; Alberola-López, C. Gaussian mixtures on tensor fields for segmentation: applications to medical imaging. *Computerized Medical Imaging and Graphics* **2011**, *35*, 16–30.
28. Robinson, J. Covariance matrix estimation for appearance-based face image processing. *Proceedings of the British Machine Vision Conference 2005* **2005**, pp. 389–398.
29. Mader, K.; Reese, G. Using covariance matrices as feature descriptors for vehicle detection from a fixed camera. *ArXiv e-prints* **2012**, [arXiv:cs.CV/1202.2528].
30. Formont, P.; Pascal, F.; Vasile, G.; Ovarlez, J.; Ferro-Famil, L. Statistical classification for heterogeneous polarimetric SAR images. *IEEE Journal of Selected Topics in Signal Processing* **2011**, *5*, 567–576. doi:10.1109/JSTSP.2010.2101579.
31. Barachant, A.; Bonnet, S.; Congedo, M.; Jutten, C. Classification of covariance matrices using a Riemannian-based kernel for BCI applications. *NeuroComputing* **2013**, *112*, 172–178.
32. Said, S.; Bombrun, L.; Berthoumieu, Y. Texture classification using Rao's distance on the space of covariance matrices. Geometric Science of Information, 2015.
33. Faraki, M.; Palhang, M.; Sanderson, C. Log-Euclidean bag of words for human action recognition. *IET Computer Vision* **2015**, *9*, 331–339. doi:10.1049/iet-cvi.2014.0018.
34. Tuzel, O.; Porikli, F.; Meer, P., Region covariance: a fast descriptor for detection and classification. In *Computer Vision – ECCV 2006*; Leonardis, A.; Bischof, H.; Pinz, A., Eds.; Springer Berlin Heidelberg, 2006; Vol. 3952, *Lecture Notes in Computer Science*, pp. 589–600. doi:10.1007/11744047_45.
35. Yuan, C.; Hu, W.; Li, X.; Maybank, S.; Luo, G., Human action recognition under log-Euclidean Riemannian metric. In *Computer Vision – ACCV 2009: 9th Asian Conference on Computer Vision, Xi'an, September 23-27, 2009, Revised Selected Papers, Part I*; Zha, H.; Taniguchi, R.i.; Maybank, S., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2010; pp. 343–353. doi:10.1007/978-3-642-12307-8_32.
36. Faraki, M.; Harandi, M.T.; Wiliem, A.; Lovell, B.C. Fisher tensors for classifying human epithelial cells. *Pattern Recognition* **2014**, *47*, 2348 – 2359.
37. Faraki, M.; Harandi, M.T.; Porikli, F. Material classification on symmetric positive definite manifolds. IEEE Winter Conference on Applications of Computer Vision. IEEE, 2015, pp. 749–756.

38. Ilea, I.; Bombrun, L.; Said, S.; Berthoumieu, Y. Covariance matrices encoding based on the log-Euclidean and affine invariant Riemannian metrics. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2018, CVPRW'18.
39. Said, S.; Bombrun, L.; Berthoumieu, Y.; Manton, J.H. Riemannian Gaussian Distributions on the Space of Symmetric Positive Definite Matrices. *IEEE Transactions on Information Theory* **2017**, *63*, 2153–2170. doi:10.1109/TIT.2017.2653803.
40. Ilea, I.; Bombrun, L.; Germain, C.; Terebes, R.; Borda, M.; Berthoumieu, Y. Texture image classification with Riemannian Fisher vectors. IEEE International Conference on Image Processing, 2016, pp. 3543 – 3547.
41. Huang, Y.; Wu, Z.; Wang, L.; Tan, T. Feature Coding in Image Classification: A Comprehensive Study. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2014**, *36*, 493–506. doi:10.1109/TPAMI.2013.113.
42. Said, S.; Hajri, H.; Bombrun, L.; Vemuri, B.C. Gaussian Distributions on Riemannian Symmetric Spaces: Statistical Learning With Structured Covariance Matrices. *IEEE Transactions on Information Theory* **2018**, *64*, 752–772. doi:10.1109/TIT.2017.2713829.
43. Arsigny, V.; Fillard, P.; Pennec, X.; Ayache, N. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. Magnetic Resonance in Medicine, 2006, Vol. 56, pp. 411–421.
44. Rosu, R.; Donias, M.; Bombrun, L.; Said, S.; Regniers, O.; Da Costa, J.P. Structure tensor Riemannian statistical models for CBIR and classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing* **2017**, *55*, 248–260. doi:10.1109/TGRS.2016.2604680.
45. Terras, A. *Harmonic analysis on symmetric spaces and applications*; Number vol. 1 in Harmonic Analysis on Symmetric Spaces and Applications, Springer-Verlag, 1988.
46. Helgason, S. *Differential geometry, Lie groups, and symmetric spaces*; Crm Proceedings & Lecture Notes, American Mathematical Society, 2001.
47. James, A.T., The variance information manifold and the functions on it. In *Multivariate Analysis–III*; Krishnaiah, P.R., Ed.; Academic Press, 1973; pp. 157 – 169.
48. Higham, N.J. *Functions of matrices: theory and computation*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2008; pp. xx+425.
49. Fletcher, P.T.; Venkatasubramanian, S.; Joshi, S. The geometric median on Riemannian manifolds with application to robust atlas estimation. *Neuroimage* **2009**, *45*, S143–S152.
50. Cheng, G.; Vemuri, B.C. A Novel Dynamic System in the Space of SPD Matrices with Applications to Appearance Tracking. *SIAM Journal on Imaging Sciences* **2013**, *6*, 592–615. doi:10.1137/110853376.
51. Muirhead, R.J. *Aspects of multivariate statistical theory*; Wiley Series in Probability and Statistics, Wiley, 1982.
52. Zanini, P.; Congedo, M.; Jutten, C.; Said, S.; Berthoumieu, Y. Parameters estimate of Riemannian Gaussian distribution in the manifold of covariance matrices. IEEE Sensor Array and Multichannel Signal Processing Workshop; , 2016.
53. Turaga, P.; Veeraraghavan, A.; Srivastava, A.; Chellappa, R. Statistical Computations on Grassmann and Stiefel Manifolds for Image and Video-Based Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2011**, *33*, 2273–2286. doi:10.1109/TPAMI.2011.52.
54. Karcher, H. Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics* **1977**, *30*, 509–541. doi:10.1002/cpa.3160300502.
55. Joachims, T. Text categorization with suport vector machines: learning with many relevant features. Proceedings of the 10th European Conference on Machine Learning. Springer-Verlag, 1998, pp. 137–142.
56. Csurka, G.; Dance, C.R.; Fan, L.; Willamowski, J.; Bray, C. Visual categorization with bags of keypoints. Workshop on Statistical Learning in Computer Vision, European Conference on Computer Vision, 2004, pp. 1–22.
57. Sra, S. A new metric on the manifold of kernel matrices with application to matrix geometric means. In *Advances in Neural Information Processing Systems 25*; Pereira, F.; Burges, C.J.C.; Bottou, L.; Weinberger, K.Q., Eds.; Curran Associates, Inc., 2012; pp. 144–152.
58. Salehian, H.; Cheng, G.; Vemuri, B.C.; Ho, J. Recursive Estimation of the Stein Center of SPD Matrices and Its Applications. IEEE International Conference on Computer Vision. IEEE Computer Society, 2013, pp. 1793–1800.
59. Jaakkola, T.; Haussler, D. Exploiting generative models in discriminative classifiers. In Advances in Neural Information Processing Systems 11. MIT Press, 1998, pp. 487–493.

60. Krapac, J.; Verbeek, J.; Jurie, F. Modeling spatial layout with Fisher vectors for image categorization. 2011 International Conference on Computer Vision, 2011, pp. 1487–1494. doi:10.1109/ICCV.2011.6126406.
61. Zanini, P.; Said, S.; Berthoumieu, Y.; Congedo, M.; Jutten, C., Riemannian online algorithms for estimating mixture model parameters. In *Geometric Science of Information: Third International Conference, GSI 2017, Paris, France, November 7-9, 2017, Proceedings*; Nielsen, F.; Barbaresco, F., Eds.; Springer International Publishing: Cham, 2017; pp. 675–683.
62. Vision Texture Database. MIT Vision and Modeling Group. Available: http://vismod.media.mit.edu/pub/VisTex.
63. Brodatz, P. *Textures: A Photographic Album for Artists and Designers*; Dover photography collections, Dover Publications, 1999.
64. Ojala, T.; Maenpaa, T.; Pietikainen, M.; Viertola, J.; Kyllonen, J.; Huovinen, S. Outex - new framework for empirical evaluation of texture analysis algorithms. Object recognition supported by user interaction for service robots, 2002, Vol. 1, pp. 701–706 vol.1. doi:10.1109/ICPR.2002.1044854.
65. Backes, A.R.; Casanova, D.; Bruno, O.M. Color texture analysis based on fractal descriptors. *Pattern Recognition* **2012**, *45*, 1984–1992. doi:10.1016/j.patcog.2011.11.009.
66. Pennec, X. Intrinsic statistics on Riemannian manifolds: basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision* **2006**, *25*, 127–154. doi:10.1007/s10851-006-6228-4.
67. Tosato, D.; Spera, M.; Cristani, M.; Murino, V. Characterizing Humans on Riemannian Manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2013**, *35*, 1972–1984. doi:10.1109/TPAMI.2012.263.