Pattern Recognition 45 (2012) 3997–4017

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A nonparametric Riemannian framework on tensor field with application to foreground segmentation

Rui Caseiro*, Pedro Martins, João F. Henriques, Jorge Batista

Institute for Systems and Robotics, Faculty of Science and Technology, University of Coimbra, 3030 Coimbra, Portugal

ARTICLE INFO

Article history:
Received 20 July 2011
Received in revised form
3 April 2012
Accepted 7 April 2012
Available online 8 May 2012

Keywords:
Nonparametric density estimation
Kernel density estimation
Riemannian geometry
Tensor manifold
Riemannian metrics
Foreground segmentation on tensor field

ABSTRACT

Background modeling on tensor field has recently been proposed for foreground detection tasks. Taking into account the Riemannian structure of the tensor manifold, recent research has focused on developing parametric methods on the tensor domain, e.g. mixture of Gaussians (GMM). However, in some scenarios, simple parametric models do not accurately explain the physical processes. Kernel density estimators (KDEs) have been successful to model, on Euclidean sample spaces, the nonparametric nature of complex, time varying, and non-static backgrounds. Founded on a mathematically rigorous KDE paradigm on general Riemannian manifolds recently proposed in the literature, we define a KDE specifically to operate on the tensor manifold in order to nonparametrically reformulate the existing tensor-based algorithms. We present a mathematically sound framework for nonparametric modeling on tensor field to foreground detection. We endow the tensor manifold with two well-founded Riemannian metrics, i.e. Affine-Invariant and Log-Euclidean. Theoretical aspects are presented and the metrics are compared experimentally. By inducing a space with a null curvature, the Log-Euclidean metric considerably simplifies the scheme, from a practical point of view, while maintaining the mathematical soundness and the excellent segmentation performance. Theoretic analysis and experimental results demonstrate the promise and effectiveness of this framework.

1. Introduction

Foreground detection is a crucial aspect in the understanding and analysis of video sequences. It is often described as the process that subdivides an image into regions of interest and background. This task usually relies on the extraction of suitable features from the image that are highly discriminative.

Statistical modeling in the color/intensity space is a widely used approach for background modeling to foreground detection. However, there are situations where these features may not be distinct enough, i.e. sometimes statistical modeling directly on image values is not enough to achieve a good discrimination (e.g. dynamic scenes, illumination variation, etc.). Thus, the image may be converted into a more information rich form, such as a *structure tensor field* [1–3] to yield latent discriminating features (e.g. in which can be encoded color, gradients, filters responses, etc.). Texture is one of the most important features in images, and therefore its consideration can greatly improve image analysis.

The structure tensor [3–6] has been introduced for such texture analysis as a fast local computation method providing a measure of the presence of edges and their orientation. In other cases, the image is already in tensor form. For example, tensor MRI may be represented in this manner from the direction of water diffusion at each pixel. In this case, brain structures such as nerve bundles comprise regions of similarly oriented tensors as water diffuses along the fibers [7,8].

Simple attempts at tensors statistical analysis are based on statistical models of the linear tensor coefficients. However, the tensor space do not conform to Euclidean geometry, because is not a vector space (e.g. the space is not closed under multiplication with negative scalars), thus standard linear statistical techniques do not apply [9]. Although the classical Euclidean operations are well adapted to general square matrices ($d \times d$), they are practically and theoretically unsatisfactory for tensors, which are very specific matrices, i.e. symmetric positive-definite ($\mathbf{S}_d^+$). Tensors form a convex half-cone in the vector space of matrices, i.e. $\mathbf{S}_d^+$ lies on a Riemannian manifold (differentiable manifold equipped with a Riemannian metric) [10].

Background modeling on tensor field has only recently been proposed for foreground detection tasks. In order to exploit the information present in all the structure tensor components and taking into account the Riemannian structure of the tensor manifold, previous work has focused on developing parametric methods on the tensor domain, e.g. mixture of Gaussians models (GMM) [11,2]. This way, the nice structure tensor properties for texture discrimination are fully exploited.

* Corresponding author. Tel.: +351 963886632.
E-mail addresses: ruicaseiro@gmail.com, ruicaseiro@isr.uc.pt (R. Caseiro).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2012.04.011

In [11], Caseiro et al. proposed a foreground detection method for tensor-valued images based on the definition of GMM on the tensor domain. They reviewed the geometrical properties of the tensor space and focused on the characterization of the mean, covariance and Normal law on that manifold. They proposed an online K-means approximation of the Expectation Maximization (EM) algorithm to estimate the mixture parameters based on the Affine-Invariant Riemannian metric [12,13]. This Riemannian metric has excellent theoretical properties and provides powerful processing tools, but essentially due to the curvature induced on the tensor space the computational burden can be high. To overcome this limitation, based on a novel vector space structure for tensors, a new metric called Log-Euclidean was presented in [14]. A space with a null curvature is obtained, while the excellent theoretical properties are preserved. This novel view-point on the tensor space provides a particularly powerful and simple-to-use framework to process tensors. Hence, classical statistical tools usually reserved to vectors are simply and efficiently generalized to tensors in the Log-Euclidean framework. This metric also has excellent theoretical properties and yields similar results in practice, but with much simpler and faster computations.

In order to speed up the foreground segmentation process, Caseiro et al. [2] presented a novel and faster online K-means algorithm based on the Log-Euclidean metric, while conserving the theoretical properties. They presented the theoretical aspects and the Affine-Invariant and Log-Euclidean frameworks are compared. From a practical point of view, results are similar, but the Log-Euclidean is much faster (at least two times faster).

However, in some scenarios, the density function that describes the data is more complex and simple parametric models do not accurately explain the physical processes, i.e. the parametric approach cannot model the nonparametric nature of complex, time varying and non-static backgrounds. As shown by Elgammal et al. [15,16], kernel density estimators (KDEs) have been successful to model, on the Euclidean sample spaces, the nonparametric nature of complex physical processes associated with the foreground segmentation problem. Seeing that recently, in the mathematics community, was proposed and rigorously defined the KDE on general Riemannian manifolds [17], it would be interesting to nonparametrically reformulate the existing tensor-based GMM algorithms [2,11]. The idea is to leave the data to show the structure lying beyond them, instead of imposing one.

This journal paper extends a previous conference publication where the nonparametric Riemannian framework on tensor field was presented for the first time [1].

1.1. Paper contributions

In this paper, we present a novel nonparametric Riemannian framework on the tensor manifold, and evaluate its usefulness to foreground segmentation. The main contributions of our work are as follows:

- It is well known that the differential geometry is not a trivial subject and even nowadays, in general, an easy introduction to differential geometry is hard to find. Therefore, throughout the paper, our goal is not only to present the proposed nonparametric tensor-based framework but is also our intention, whenever possible, to provide the necessary knowledge about differential geometry in order to enable the average reader to understand and implement the derived approaches.
- Founded on the mathematically rigorous KDE on general Riemannian manifolds proposed by Pelletier [17], we define a KDE specifically to operate on the tensor manifold. To accomplish this, the tensor manifold is endowed with two Riemannian metrics (Affine-Invariant and Log-Euclidean) and with a Euclidean metric to prove the benefits of take into account the Riemannian structure. By inducing a space with a null curvature, the Log-Euclidean metric considerably simplifies the scheme.
- We present a mathematically sound framework for nonparametric modeling on tensor field to foreground detection. In the literature, Caseiro et al. [2,11] were the only one to use the paradigm of background modeling on tensor field to foreground detection. Taking into account that their work is based on a parametric approach (GMM) on the tensor domain, to the best of our knowledge, this is the first time that a nonparametric modeling technique on the tensor domain is applied to the foreground detection problem. We generalized herein the nonparametric background model proposed by Elgammal [15,16], one of the most widely used per-pixel models, from pixel domain (vector space features) to tensor domain. We nonparametrically reformulated the tensor-based GMM proposed by Caseiro et al. [2] in a similar way to what Elgammal [15,16] did in relation to Stauffer's work [18] (GMM on the vectorial domain).

The remainder of the paper is organized as follows: Section 2 reviews the related work in the field of foreground segmentation. In Section 3, we describe the tensor descriptors used in this paper, namely the structure tensor (ST) [3] and the region covariance matrix (RCM) [19]. Section 4 provides a brief introduction to the differential geometry and the main notions of the geometric properties of the general Riemannian manifolds. In Section 5, we focus on the space of symmetric positive definite matrices describing the main geometric properties of this manifold endowed with the standard Euclidean metric and with two Riemannian metrics (Affine-Invariant and Log-Euclidean). In Section 6, we present a proper derivation of KDE on general Riemannian manifolds and we extend this concept to the tensor manifold endowed with all the three metrics previously referred. Section 7 demonstrates the experimental results conducted on several challenging video sequences presented in previous literature. Section 8 summarizes the paper.

2. Related work

For the sake of brevity, the related work description will be neither rigorous nor complete, but we want to at least outline some of the key ideas. Please refer to [20–25] for a set of excellent surveys.

Over the years, a considerable number of background models for foreground detection have been proposed. These models can be broadly classified into *pixel-wise* (*temporal*) and *block-wise* (*spatio-temporal*) models.

Pixel-wise models: They rely on the separation of a statistical model for each pixel and the pixel models are learned entirely from each pixel history. The background model can be parametrically estimated using a single Gaussian distribution, a mixture of Gaussians (GMM) or through Bayesian approaches. Once the per-pixel background model was derived, the likelihood of each incident pixel can be calculated and labeled as belonging, or not, to the background. In [26], Wren et al. modeled the statistical distribution of each color pixel with a single three-dimensional Gaussian, whose parameters are regularly updated by a simple adaptive filter. This model works for static or slowly changing backgrounds but fails in the case of dynamic backgrounds. To handle with possible data multi-modalities, Friedman et al. [27] extended the concept of Gaussian distribution by using a mixture of Gaussian distributions (GMM). In their work, the intensity is

modeled by a mixture of three Gaussians (background, moving object and shadow) and the pixel model is learned by an incremental EM algorithm. Stauffer et al. [18] proposed to represent each color pixel as a mixture of (3–5) Gaussians distributions to capture the multi-modal nature of the background and the mixture parameters are updated using an online K-means approximation of the EM algorithm to meet real time requirements. Based on the persistence and the variance of each of the Gaussians distributions, a mixture background is determined. In [28], Porikli et al. model each color pixel as a set of layered Normal distributions, and used a recursive Bayesian learning approach not only to estimate the mean and variance of each layer but also to obtain probability distributions of the mean and variance. Their Bayesian approach can also estimate the ideal number of necessary layers for representing each pixel.

In some scenarios, the density function that describes the pixel data is more complex and simple parametric models do not accurately explain the physical processes, i.e. the parametric approach cannot model the nonparametric nature of complex, time varying and non-static backgrounds. Therefore, one needs to employ nonparametric estimation techniques that do not make any assumptions about the *pdf*, except the mild assumption that *pdf* are smooth functions, and can represent arbitrary *pdfs* given sufficient data. Elgammal et al. [15,16] proposed the use of Gaussian kernels (KDE) to estimate the density function of each pixel from its past samples. Foreground detection is performed by thresholding the probability of the observed samples. The pixel-based methods mentioned above do not consider the correlation between pixels. In general, they will fail when the scenes to be modeled are dynamic natural scenes, which include repetitive motions like swaying vegetation, waving trees, rippling water, etc.

Block-wise models: In the case of block-based models, the background model of a pixel depends not only on that pixel but also on the nearby pixels (e.g. [29,30]). These models consider spatial information an essential element to understand the structure of the scene. In [31], Oliver et al. considered the whole image as a single block and used the best M eigenvectors generated by applying PCA to a set of training images to represent the background. In [32], Monnet et al. divided each frame into blocks and then mapped each block into a lower dimensional feature space whose basis vectors were incrementally updated. A prediction mechanism was used in the lower dimensional feature space for background–foreground differentiation. In [33], Seki et al. proposed a background subtraction method in which the frames were divided into blocks and co-occurrences of image variations at neighboring blocks were used for dynamically narrowing the permissible range of background image variations. One major disadvantage of these block-based methods is that the boundary of the foreground objects cannot be delineated exactly. In recent years, researchers have been concentrating more on incorporating spatial aspect into background modeling to take advantage of the correlation that exists between neighboring pixels [34]. Thus, the background model of a pixel also depends on its neighbors. Jabri et al. [35] were one of the first to use image gradient information as a feature. They presented an approach to detect people by an adaptive fusion of color and edge information using confidence maps. In [36], Javed et al. used gradient magnitude and orientation, as well as color information, to create a five-dimensional mixture of Gaussian algorithm to achieve a more accurate background subtraction. In [37], Pless used a mixture-of-Gaussians distribution for each pixel in the feature space defined by intensity and the spatio-temporal derivatives of intensity at that pixel. Sheikh et al. [38] proposed to model the full background with a single distribution, instead of one distribution per pixel, and included image pixel position into the model, unifying the temporal and spatial consistencies into a single model. They used a MAP-MRF framework to stress spatial context to detect moving objects. In [39], Babacan et al. used a spatio-temporal hybrid model. Gibbs–Markov random field was used to model spatial interactions and Gaussian mixture model was used to model temporal interactions.

Some researchers also used texture based methods to incorporate spatial aspect into background models (e.g. [40–42]). Spatial variation information, such as gradient (or edge) feature, helps to improve the reliability of structure change detection. A textured-based method was used in [43], where each pixel is non-parametrically modeled as a group of adaptive local binary pattern (LBP) histograms that are calculated over a circular region around the pixel, which means that no assumptions about the underlying distributions are needed. Odobez et al. [44] proposed a robust multi-layer background subtraction technique, using local texture features represented by local binary patterns (LBPs) and photometric invariant color measurements in RGB color space. They intend to overcome the problems of LBP single-layer approach in uniform regions, in situations of light variation. Recently, the concept of local binary patterns (LBPs) proposed in [43] was extended, from spatial to spatio-temporal domain. Zhang et al. [45] modeled each pixel as a group of STLBP (spatio-temporal local binary pattern) histograms. Several variants of background models, based on the LBP features, have been proposed in the literature, namely in [46–48].

Finally, there are several other important works in the literature to address some specific problems in the foreground segmentation task that we want to remark, e.g. highly dynamic/complex scenes [49,50], highly dynamic scenes and real time requirements [51], sudden illumination changes [52] and freely moving cameras [53,54].

To the best of our knowledge, Caseiro et al. [2,11] were the only ones to use the paradigm of background modeling on tensor field to foreground detection. They proposed a tensor-based parametric approach (GMM).

3. Tensor descriptors

Positive definite symmetric matrices (tensors) are widely used in image processing. As previously referred, two typical applications capture structural information of an image by means of a structure tensor (ST) [4,3,5,6] and characterize the diffusion of water molecules in DT-MRI [7,8,55]. Region covariance matrices (RCMs), which are also tensors, have recently been a popular choice for versatile tasks like texture classification [19], object detection [56] and tracking [57] in video sequences, due to their powerful properties as local descriptor and their low computational demands. Taking into account that the RCM has some special properties that can help in more difficult scene conditions (e.g. image noise), we also present herein the RCM as a descriptor for foreground detection. This section outlines the similarities of RCM to the ST.

3.1. Structure tensor (ST)

The structure tensor analyzes dependencies between different low-level features and it gained great success in corner detection [58], optical flow estimation [59], etc. Consider that for each image pixel we have a window of size $w \times w$ centered at that pixel and let R be the set of $S = w^2$ samples inside the window (pixel neighborhood). Each pixel p in the region R is represented by a d-dimensional feature vector v_p. The classical structure tensor $\mathbf{T}$, with only gradient information, is a 2×2 matrix defined as the

product of the image derivatives and formed as follows:

$$\mathbf{T} = \iint_{p \in R} K_\rho * (v_p \cdot (v_p)^T) = \iint_{p \in R} K_\rho * \begin{pmatrix} I_{p,x}^2 & I_{p,x} I_{p,y} \\ I_{p,y} I_{p,x} & I_{p,y}^2 \end{pmatrix} \quad (1)$$

$$\mathbf{T}_c = \iint_{p \in R} K_\rho * \sum_{i=1}^{C} (v_p^i \cdot (v_p^i)^T) \quad (2)$$

with $v_p = [I_{p,x} \;\; I_{p,y}]$, where I is a gray image, K_ρ is a Gaussian kernel with standard deviation ρ, and $(I_{p,x}, I_{p,y})$ are the first order derivatives. Therefore, it analyzes the dependency of the image derivatives without normalization (as opposed to covariance matrices). The structure tensor represents the local orientation by its eigenvectors and eigenvalues.

For vector-valued images, e.g. color images, the structure tensor $\mathbf{T}_c$ may be formed by summing along the color channels (see Eq. (2)), where C is the number of color channels and $v_p^i = [I_{p,x}^i \;\; I_{p,y}^i]$ are the first order derivatives for each color channel i [60,5]. In general, augmenting the feature vector improves segmentation by increasing the information available. For example, in [5] was included the intensity information with the image derivatives. Since for foreground segmentation purposes we want to model the dependencies of multiple low-level features, including for example color/intensity, texture, first and second order derivatives, filter responses, etc., we use the generalized form of the structure tensor [4,61] as descriptor. The generalized structure tensor $\mathbf{T}_G$ is a powerful analytical tool that can model and estimate the position and orientation of feature patterns and is defined by

$$\mathbf{T}_G = \iint_{p \in R} K_\rho * \begin{pmatrix} v_{p,1}^2 & v_{p,1} v_{p,2} & \cdots & v_{p,1} v_{p,d} \\ \vdots & \vdots & \ddots & \vdots \\ v_{p,d} v_{p,1} & v_{p,d} v_{p,2} & \cdots & v_{p,d}^2 \end{pmatrix} \quad (3)$$

where v_p is the d-dimensional feature vector containing the low-level features to be considered. Therefore, d features yield a symmetric $d \times d$ generalized structure tensor $\mathbf{T}_G$, which is used to describe the unnormalized feature dependencies within a local image patch.

The structure tensor descriptor owns good properties. First of all, it provides an effective way to fuse different features. The structure tensor descriptor integrates two distinct levels: pixel level and region level. At the pixel level, appearance properties, i.e. intensity, gradient, etc., are used to describe each pixel. In the region level, the correlation of features extracted at the pixel level is represented by the structure tensor that is calculated over a square region around the pixel. Computing the structure tensor descriptor from multiple information sources yields a straightforward technique for a low-dimensional feature representation, since a structure tensor contains in its diagonal elements the variance of each source channel and off diagonal the correlation values between the involved modalities, which is very important for dynamic background modeling. The second advantage is scale invariant since the order of structure tensor descriptor does not depend on the window size, but on the dimension of the feature vector. This property enables comparing two windows without being restricted to the same window size. Thirdly, it provides some invariance to illumination since the structure tensor descriptor contains the partial derivatives which can compensate small illumination changes.

3.2. Region covariance matrix (RCM)

Here, we present a brief overview of the region covariance matrix [19]. Let I be a one-dimensional intensity or a three-dimensional color image, and F be the d-dimensional feature image extracted from I

$$F(x,y) = \Phi(I,x,y) \quad (4)$$

$$\Sigma_R = \frac{1}{S-1} \sum_{p=1}^{S} (v_p - \mu_R)(v_p - \mu_R)^T \quad (5)$$

where the function Φ can be any mapping such as intensity, color, texture, gradients, edge magnitude/orientation, and filter responses. This list can be extended by including higher order derivatives, texture scores, and temporal frame differences. For a given region ($w \times w$ window) $R \subset F$, let $\{v_p\}_{p=1,\ldots,S}$ ($S = w^2$) be the d-dimensional feature vectors inside R. The region R is represented with the $d \times d$ covariance matrix Σ_R of the feature points given by Eq. (5), where μ_R is the vector of the means of the corresponding features for the points within the region R. In practice, the difference between the structure tensor (ST) and the region covariance matrix (RCM) is basically the zero-mean normalization performed in the covariance calculus.

There are several advantages of using covariance matrices as region descriptors over the structure tensor descriptor that are important for foreground segmentation. Firstly, the noise corrupting individual samples are largely filtered out with the average filter during covariance computation. Secondly, the covariance is invariant to mean changes such as identical shifting of color values. This is very valuable when scenes are under some varying illumination conditions, i.e. due to the zero-mean normalization by subtraction the sample mean, the descriptor achieves some invariance in the case of photometric and illumination changes.

4. Differential geometry

In this section, we will try to briefly review some basic theory of differential geometry and the main notions from Riemannian geometry that will be required in the sequel. For the sake of brevity, our treatment will not be complete, but we want to at least outline some of the key concepts that we consider crucial to understand the proposed framework. We try to make the paper self contained and at the same time keep the notions of differential geometry that we use to a minimum. A thorough introduction to differential geometry can be found in [10,62–64]. We recommend Barret's book [10] for a more comprehensive treatment.

Manifold: let $\mathcal{M}$ be a n-manifold. A manifold $\mathcal{M}$ is a topological space that is locally similar to an Euclidean space $\Re^n$. This is formally achieved by building mappings which make each small patch of a manifold similar to an open set in the Euclidean space and this similarity is defined by the *coordinate charts* at each point. It is generally not possible to define global coordinates which make the whole manifold look like an Euclidean space.

Nevertheless, coordinate charts are an essential tool for addressing fundamental notions such as the differentiability of a function on a manifold. To do this, we need to answer the following questions: How do we find these patches which look similar to Euclidean space? How are the different patches related to each other?

Formally, a manifold, $\mathcal{M}$ is a Hausdorff topological space that is locally *homeomorphic* to an Euclidean space. Points can be separated by neighborhoods such that for each point $\mathbf{P} \in \mathcal{M}$ there exists a neighborhood $\mathcal{U} \subset \mathcal{M}$ containing $\mathbf{P}$ and an associated *homeomorphism* $\varphi : \mathcal{U} \to \tilde{\mathcal{U}}$ (*one-to-one*, *onto* and *continuous* mapping in both directions) from $\mathcal{U}$ to some Euclidean space $\Re^n$, such that $\varphi(\mathcal{U})$ is an open set in $\Re^n$, i.e. $\varphi(\mathcal{U}) = \tilde{\mathcal{U}} \subset \Re^n$.

The neighborhood $\mathcal{U}$ and its associated mapping φ form together a *coordinate chart* $(\mathcal{U},\varphi)$. Given a coordinate chart $(\mathcal{U},\varphi)$ and $\mathbf{P} \in \mathcal{U}$, the set $\mathcal{U}$ is called a *coordinate domain* or a *coordinate*

neighborhood. The map φ is denominated as *local coordinate map*, and the component functions of φ are called *local coordinates* on $\mathcal{U}$, i.e. the chart defines a *local coordinate system* $x=(x^1,\ldots,x^n)^T$. The elements of $\varphi(\mathbf{P})\in\Re^n$ are called the local coordinates of $\mathbf{P}$ in the chart $(\mathcal{U},\varphi)$. The interest of the notion of chart $(\mathcal{U},\varphi)$ is that it makes it possible to study objects associated with $\mathcal{U}$ by bringing them to the subset $\varphi(\mathcal{U})$ of $\Re^n$.

Riemannian manifold: It is a differentiable manifold $\mathcal{M}$ endowed with a Riemannian metric g. It is possible to define on the same manifold different metrics and obtain different Riemannian manifolds. The metric is chosen to have geometrical significance such as being invariant to a set of geometric transformations.

Tangent space: for differentiable manifolds, it is possible to define the derivatives of curves on the manifold. The derivatives at a point $\mathbf{P}\in\mathcal{M}$ lie on a vector space $T_{\mathbf{P}}\mathcal{M}$, which is the *tangent space* at that point. The tangent space can be thought of as the set of allowed velocities for a point constrained to move on the manifold, i.e. the tangent space $T_{\mathbf{P}}\mathcal{M}$, defined $\forall\mathbf{P}\in\mathcal{M}$, is simply a vector space, attached to $\mathbf{P}$, which contains the *tangent vectors* to all curves on $\mathcal{M}$ passing through $\mathbf{P}$ (set of all tangent vectors at $\mathbf{P}$).

Riemannian metric: a *Riemannian metric* is defined by a continuous collection of inner products $\langle\cdot,\cdot\rangle_{\mathbf{P}}$, defined $\forall\mathbf{P}\in\mathcal{M}$ on the tangent space $T_{\mathbf{P}}\mathcal{M}$. For continuity, the inner product vary smoothly with $\mathbf{P}$. We denote this inner product by g and for two tangent vectors $u,v\in T_{\mathbf{P}}\mathcal{M}$ the inner product is written as $g_{\mathbf{P}}(u,v)$. The inner product induces a norm for u given by $\|u\|=\sqrt{g_{\mathbf{P}}(u,u)}$. Given a chart $(\mathcal{U},\varphi)$ at $\mathbf{P}$ with a local coordinate system $x=(x^1,\ldots,x^n)$, it is possible to determine a *basis* $\partial/\partial x=(\partial_1,\ldots,\partial_n)$ of the tangent space $T_{\mathbf{P}}\mathcal{M}$ (∂_i = shorter notation for $\partial/\partial x^i$). Any element of the $T_{\mathbf{P}}\mathcal{M}$ can be expressed in the form $\sum_{i=1}^n x^i\partial_i$ (Fig. 1). We can express the metric in this basis by a $(n\times n)$ symmetric, bilinear and positive-definite form $G_{\mathbf{P}}(x)=[g_{ij}(x)]_{\mathbf{P}}$ given by the inner products $g_{ij}(x)=\langle\partial_i,\partial_j\rangle_{\mathbf{P}}$.

The form $G_{\mathbf{P}}(x)$ is called the *local representation of the Riemannian metric*. Tangents can now be represented as vectors in this basis and relative to this basis the inner product can be written as a symmetric positive definite matrix. The Riemannian metric is an inherently geometric notion. It does not require the definition of a coordinate chart or a basis for $T_{\mathbf{P}}\mathcal{M}$. Different charts lead to different coordinates for tangent vectors and different Riemannian metric matrices, but for a given pair of tangents the inner product is independent of the basis.

Geodesic: for Riemannian manifolds, tangents vectors (on the tangent space) and geodesics (on the manifold) are closely related (see Fig. 1). Distances on manifolds are defined in terms of minimum length curves between points on the manifold.

The *geodesic* between two points $\gamma(0)$ and $\gamma(1)$ on a Riemannian manifold is locally defined as the minimum length curve $\gamma(t):I=[0,1]\subset\Re\rightarrow M$ over all possible smooth curves on the manifold connecting these points. This minimum length is called *geodesic* or *intrinsic distance* ($D(\gamma(0),\gamma(1))=\langle\dot\gamma(0),\dot\gamma(0)\rangle_{\gamma(0)}^{1/2}$).

The geodesic concept is the equivalent of straight line in Euclidean spaces, defined as the locally length-minimizing piecewise smooth curve and characterized by the fact that it is *autoparallel*, e.g. the field of tangent vectors $\dot\gamma(t)$ stays parallel along $\gamma(t)$ (the velocity is constant along the geodesic). This property of having zero acceleration is sometimes used to define a geodesic. It is equivalent to say that, in local coordinates notation, a curve is a geodesic if and only if it is the solution of the n second order Euler–Lagrange equations (where Γ_{ij}^k is the *Christoffel symbols* of the second kind [10])

$$\frac{d^2x^k}{dt^2}+\sum_{i,j=1}^{n}\Gamma_{ij}^k\frac{dx^i}{dt}\frac{dx^j}{dt}=0\quad\forall\,k=1,\ldots,n \tag{6}$$

Let $\gamma(0)=\mathbf{P}$, given a tangent vector $u\in T_{\mathbf{P}}\mathcal{M}$ there exists a unique geodesic $\gamma(t)$ starting at $\mathbf{P}$ with initial velocity $\dot\gamma(0)=u$. Therefore, the geodesic $\gamma(t)$ is uniquely defined by its starting point $\mathbf{P}$ and its initial velocity $\dot\gamma(0)$. The endpoint $\gamma(1)$ of the geodesic curve can be computed by applying the *exponential map* at $\mathbf{P}$, such that $\gamma(1)=\exp_{\mathbf{P}}(\dot\gamma(0))$. Two maps are defined for mapping points between the manifold and a tangent space (*exponential map* and *logarithm map*).

Exponential map: the *exponential map*, $\exp_{\mathbf{P}}:T_{\mathbf{P}}\mathcal{M}\rightarrow\mathcal{M}$ is a mapping between the tangent space $T_{\mathbf{P}}M$ and the corresponding manifold $\mathcal{M}$. It maps the tangent vector $\dot\gamma(0)=u$ at point $\mathbf{P}=\gamma(0)$ to the point reached by the geodesic at time step one, $\gamma(1)=\exp_{\mathbf{P}}(\dot\gamma(0))$. The origin of the $T_{\mathbf{P}}\mathcal{M}$ is mapped to the point itself, $\exp_{\mathbf{P}}(0)=\mathbf{P}$. For each point $\mathbf{P}\in\mathcal{M}$, there exists a neighborhood $\tilde{\mathcal{U}}$ of the origin in $T_{\mathbf{P}}\mathcal{M}$, such that $\exp_{\mathbf{P}}$ is a *diffeomorphism* from $\tilde{\mathcal{U}}$ onto a neighborhood $\mathcal{U}$ of $\mathbf{P}$.

Logarithm map: in general, the exponential map is *onto* but only *one-to-one* in a neighborhood of $\mathbf{P}$. Therefore, the inverse mapping, given by the *logarithm map* $\log_{\mathbf{P}}:\mathcal{M}\rightarrow T_{\mathbf{P}}\mathcal{M}$ is uniquely defined only around the neighborhood of the point $\mathbf{P}$. Over this neighborhood $\mathcal{U}$, we can define the inverse of the exponential map, i.e. the mapping from $\mathcal{U}$ to $\tilde{\mathcal{U}}$ is the *logarithm map* $\log_{\mathbf{P}}=\exp_{\mathbf{P}}^{-1}:\mathcal{U}\rightarrow\tilde{\mathcal{U}}$. It maps any point $\mathbf{Q}\in\mathcal{U}$ to the unique tangent vector $u\in T_{\mathbf{P}}\mathcal{M}$ that is the initial velocity $\dot\gamma(0)$ of the unique geodesic $\gamma(t)$ between $\gamma(0)=\mathbf{P}$ and $\gamma(1)=\mathbf{Q}$. In other words, for two points $\mathbf{P}$ and $\mathbf{Q}$ on the manifold $\mathcal{M}$ the tangent vector to the geodesic curve from $\mathbf{P}$ to $\mathbf{Q}$ is defined as $\dot\gamma(0)=\log_{\mathbf{P}}(\gamma(1))$.

Normal neighborhood: the neighborhood $\tilde{\mathcal{U}}$ is not necessarily convex. However, $\tilde{\mathcal{U}}$ is *star-shaped*, i.e. for any point $\in\tilde{\mathcal{U}}$, the line joining the point to the origin is contained in $\tilde{\mathcal{U}}$. The image of a star-shaped neighborhood under the exponential map is a neighborhood of $\mathbf{P}$ on the manifold. This neighborhood is the *normal neighborhood*. The exponential map can be used to define suitable coordinates for normal neighborhoods. Let $\tilde{\mathcal{U}}$ be a star-shaped neighborhood at the origin in $T_{\mathbf{P}}\mathcal{M}$ and let $\mathcal{U}$ be its image under the exponential map, i.e., $\mathcal{U}$ is a normal neighborhood of $\mathbf{P}$. Let e_i, $\forall i=1,\ldots,n$ be an orthonormal coordinate system for $T_{\mathbf{P}}\mathcal{M}$.

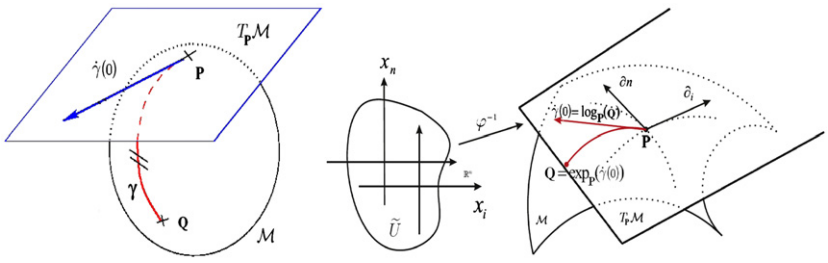

Fig. 1. *Left*: The *geodesic* $\gamma(t)$ defined by the starting point $\mathbf{P}$ and the initial velocity $\dot\gamma(0)$. The endpoint $\mathbf{Q}=\gamma(1)$ is computed by applying the *exponential map*, such that $\mathbf{Q}=\exp_{\mathbf{P}}[\dot\gamma(0)]$. *Right*: Local coordinate system, geodesic $\gamma(t)$, tangent space and exponential map at $\mathbf{P}\in\mathcal{M}$ (these images were adapted from the originals presented in [9,65] respectively).

Therefore $g(e_i,e_j)=0$ if $i\neq j$ and $g(e_i,e_j)=1$ if $i=j$. The *normal coordinate system* of $\mathbf{P}$ is the coordinate chart $(\mathcal{U},\varphi)$ which maps $\mathbf{Q}\in\mathcal{U}$ to the coordinates of $\log_{\mathbf{P}}(\mathbf{Q})$ in the orthonormal coordinate system $\Rightarrow \log_{\mathbf{P}}(\mathbf{Q})=\sum_{i=1}^{n}\varphi^i(\mathbf{Q})e_i$, where $\varphi^i(\mathbf{Q})$ is the ith coordinate of $\varphi(\mathbf{Q})\in\Re^n$ [10].

Connection: the *curvature* concept plays an important role in the expression of the KDE on manifolds. Before introducing the curvature notion, we need to precise the notion of *connection* ∇ [10]. It is crucial in geometry since it allows to transport quantities along curves in a consistent manner and, ultimately, to compare local geometries defined at different manifold locations [66]. The connection makes it possible to map any tangent space $T_{\mathbf{P}}\mathcal{M}$ onto another tangent space $T_{\mathbf{Q}}\mathcal{M}$.

The need of such a mapping arises: imagine that we want to transport a vector, in a parallel manner, from its original point $\mathbf{P}$ to $\mathbf{Q}$. In general, the *parallel transport* procedure is dependent on the choice of the coordinate system, which is not desirable. This dependency directly comes from the fact that the classical directional derivative does not behave well under changes of the coordinate system. It is possible to solve this problem, i.e. to make the differentiation intrinsic, by considering the *covariant derivative*. The covariant derivative is a way of specifying a derivative along tangent vectors of a manifold, i.e. the orthogonal projection of the usual derivative of the vector fields onto the tangent space.

The canonical affine connection on a Riemannian manifold is the *Levi-Civita connection* [67] and is directly defined from the covariant derivative. It parallel transports a tangent vector along a curve while preserving its inner product (it is compatible with the metric, i.e. the covariant derivative of the metric is zero). The Levi-Civita coefficients are defined, in each local chart by the Christoffel symbols of the second kind Γ_{ij}^k as follows:

$$\nabla_{ij}^k=\Gamma_{ij}^k=g^{kl}\Gamma_{ijl}=\frac{1}{2}g^{kl}\left(\frac{\partial g_{jl}}{\partial x^i}+\frac{\partial g_{il}}{\partial x^j}-\frac{\partial g_{ij}}{\partial x^l}\right) \tag{7}$$

$\forall i,j,k,l=1,\ldots,n$, using Einstein's summation convention [62] and g^{kl} being the inverse of the metric.

Riemannian curvature tensor ($\mathbf{R}$): the notion of curvature for Riemannian manifolds of dimension ≥ 3 cannot be fully described by a scalar quantity at each point $\mathbf{P}\in\mathcal{M}$. It can be expressed in terms of the metric tensor and its first and second derivatives. The Riemann curvature tensor measures the covariant derivatives non-commutativity. In local coordinates, it can be expressed through the Christoffel symbols as follows:

$$\mathbf{R}_{ijk}^l=\partial_j\Gamma_{ki}^l-\partial_k\Gamma_{ji}^l+\Gamma_{jm}^l\Gamma_{ki}^m-\Gamma_{km}^l\Gamma_{ji}^m \tag{8}$$

Ricci curvature tensor ($\mathcal{R}$): the Ricci tensor is defined as the contraction of the Riemann curvature tensor ($\mathbf{R}$) and can be thought of as the Laplacian of the Riemannian metric e.g. is a way to measure how much n-dimensional volumes in regions of an n-dimensional manifold differ from the volumes of equivalent regions in $\Re^n$. For Riemannian manifolds up to dimension three the Ricci tensor completely describe its curvature. For manifolds of dimension ≥ 4 it become insufficient. However, it plays an crucial role in Section 6 to define the KDE on the tensor space. The Ricci tensor is given as follows:

$$\mathcal{R}_{ij}=\mathbf{R}_{ijk}^k=\mathbf{R}_{ijkl}g^{kl} \tag{9}$$

5. Tensor manifold (SPD)

The space of $d\times d$ symmetric positive-definite matrices $\mathbf{S}_d^+$ is probably the most important set of matrices that one deals with in various branches of mathematics, numerical analysis, physics, mechanics, probability, medical imaging and other fundamental and engineering sciences.

Recall that a real $d\times d$ matrix $\mathbf{A}$ is symmetric if $\mathbf{A}=\mathbf{A}^T$. We denote by $\mathbf{S}_d$ the vector space of the $d\times d$ *symmetric matrices*. We say that a symmetric matrice $\mathbf{A}\in\mathbf{S}_d$ is *positive definite* if $x^T\mathbf{A}x>0$ for all nonzero $x\in\Re^d$. Although the space of symmetric matrices $\mathbf{S}_d$ is a vector space, the space of symmetric positive-definite matrices $\mathbf{S}_d^+$ (also called tensors by abuse of language) is a differentiable manifold with a natural Riemannian structure [9].

The specific forms of the operators (metric, inner product, geodesic distance, maps, etc.) defined in Section 4 for the general Riemannian manifolds depend on the manifold and the metric.

Because of its importance, the set of $\mathbf{S}_d^+$, as a Riemannian manifold, has been analyzed from several perspectives, e.g. different Riemannian metrics and intrinsic structures were defined [9,12–14]. In this section, we present the explicit formulae for the tensor manifold $\mathbf{S}_d^+$ endowed with the three metrics studied in this paper. Namely, we will present the conventional Euclidean metric (E), then we describe the geometry of $\mathbf{S}_d^+$ equipped with an Affine-Invariant Riemannian metric (AI) [12,13] derived from the Fisher information matrix [68], and finally we exploit the properties of a new metric, based on a novel vector space structure for tensors, called Log-Euclidean (LE) [14].

In the following, we will make an extensive use of a few functions on symmetric matrices, namely the matrix exponential/logarithm. The exp and log are the ordinary matrix exponential/logarithm operators. Not to be confused, $\exp_{\mathbf{P}}$ and $\log_{\mathbf{P}}$ are manifold specific operators, which are point dependent, $\mathbf{P}\in\mathbf{S}_d^+$. The exponential/logarithm of general matrices can be defined using series. In the case of symmetric matrices, we have some important simplifications and these operators can be computed easily [9]. Let $\mathbf{P}=\mathbf{U}\mathbf{D}\mathbf{U}^T$ be the eigenvalue decomposition of a symmetric matrix. These matrix operators are given by

$$\exp(\mathbf{P})=\sum_{k=0}^{\infty}\frac{\mathbf{P}^k}{k!}=\mathbf{U}\exp(\mathbf{D})\mathbf{U}^T \tag{10}$$

$$\log(\mathbf{P})=\sum_{k=1}^{\infty}\frac{(-1)^{k-1}}{k}(\mathbf{P}-I)^k=\mathbf{U}\log(\mathbf{D})\mathbf{U}^T \tag{11}$$

where $\exp(\mathbf{D})$ and $\log(\mathbf{D})$ are the diagonal matrices of the eigenvalue exponential and logarithm, respectively.

Through the mapping φ that associates to each $\mathbf{P}\in\mathbf{S}_d^+$ its n independent components σ_{kl} $(k\leq l\rightarrow k,\ l=1,\ldots,d)$, we see that $\mathbf{S}_d^+$ is isomorphic to $\Re^n$ with $n=\frac{1}{2}d(d+1)$. Thus, we can consider $\mathbf{S}_d^+$ as an n-dimensional manifold where the coordinates $x=(x^1,\ldots,x^n)^T$ will be the independent components of the matrix $\mathbf{P}$ and linearly accessed through φ, with $x^i=\sigma_i=\sigma_{kl}$ with $(i=1,\ldots,n)$ and $(k\leq l\rightarrow k,l=1,\ldots,d)$.

5.1. Euclidean metric (E)

Considering the standard Euclidean metric, the dissimilarity measure $D_{\mathrm{E}}(\mathbf{P},\mathbf{Q})$ between tensors $\mathbf{P},\mathbf{Q}\in\mathbf{S}_d^+$ is given by the Frobenius norm of the difference [7]

$$D_{\mathrm{E}}(\mathbf{P},\mathbf{Q})=|\mathbf{P}-\mathbf{Q}|_F=\sqrt{\mathrm{tr}((\mathbf{P}-\mathbf{Q})(\mathbf{P}-\mathbf{Q})^{\mathbf{T}})} \tag{12}$$

$$\nabla_{\mathbf{P}}D_{\mathrm{E}}^2(\mathbf{P},\mathbf{Q})=\mathbf{P}-\mathbf{Q} \tag{13}$$

where tr denotes the trace operator. The gradient of the squared Euclidean distance $\nabla_{\mathbf{P}}D_{\mathrm{E}}^2(\mathbf{P},\mathbf{Q})$ can be proved to correspond to the usual *difference* tangent vector. The empirical mean tensor $\overline{\mathbf{T}}_{\mathrm{E}}$ over a set of N tensors $\{\mathbf{T}_i\}$ is estimated as

$$\overline{\mathbf{T}}_{\mathrm{E}}=\frac{1}{N}\sum_{i=1}^{N}\mathbf{T}_i \tag{14}$$

5.2. Affine-invariant metric (AI)

Using the fact that the manifold of the multivariate normal distributions with zero mean can be identified with the tensor manifold $\mathbf{S}_d^+$, a Riemannian metric on $\mathbf{S}_d^+$ can be deduced/introduced in terms of the Fisher information matrix [68]. An Affine-Invariant Riemannian metric [12,13] for the tensor space $\mathbf{S}_d^+$, derived from the Fisher information matrix, is given $\forall \mathbf{P} \in \mathbf{S}_d^+$ by the twice covariant tensor

$$g_{ij} = g(E_i,E_j) = \langle E_i,E_j \rangle_{\mathbf{P}} = \tfrac{1}{2}\operatorname{tr}(\mathbf{P}^{-1}E_i\mathbf{P}^{-1}E_j) \tag{15}$$

$\forall i,j=(1,\ldots,n)$, where E_i is a $d \times d$ matrix. We denote by $\{\partial_i\}_{i=1,\ldots,n} = \{E_i\}_{i=1,\ldots,n}$ the canonical basis of the tangent space of the manifold $\mathbf{S}_d^+$ (e.g. the space of vector fields). We equally denote by $\{E_i^*\}_{i=1,\ldots,n}$ the dual basis of the cotangent space of $\mathbf{S}_d^+$ (e.g. the space of differential forms). The tangent space of $\mathbf{S}_d^+$ coincides with the space of $d \times d$ symmetric matrices $\mathbf{S}_d$ and the basis is given by

$$E_i = E_{kl} = \begin{cases} 1_{kk}, & k=l \\ (1_{kl}+1_{lk}), & k \neq l \end{cases} \tag{16}$$

$$E_i^* = E_{kl}^* = \begin{cases} 1_{kk}, & k=l \\ \frac{1}{2}(1_{kl}+1_{lk}), & k \neq l \end{cases} \tag{17}$$

with $(i=1,\ldots,n)$ and $(k \leq l \rightarrow k,\ l=1,\ldots,d)$, where 1_{kl} stands for the $d \times d$ matrix with 1 at row k and column l and 0 everywhere else. Recalling that $(\partial_1,\ldots,\partial_n)$ define a basis of the tangent space $\mathbf{T}_{\mathbf{P}}\mathcal{M}$, for any tangent vectors $u,v \in \mathbf{S}_d$, in tangent space $T_{\mathbf{P}}\mathcal{M}$, their inner product relative to point $\mathbf{P}$ is given by

$$\langle u,v \rangle_{\mathbf{P}} = \tfrac{1}{2}\operatorname{tr}(\mathbf{P}^{-1}u\mathbf{P}^{-1}v) \tag{18}$$

Let $\gamma : [0,1] \subset \Re \rightarrow \mathcal{M}$ be a curve in $\mathbf{S}_d^+$, with endpoints $\gamma(0)=\mathbf{P}$ and $\gamma(1)=\mathbf{Q}$, $\forall \mathbf{P},\mathbf{Q} \in \mathbf{S}_d^+$. The geodesic defined by the initial point $\gamma(0)=\mathbf{P}$ and the tangent vector $\dot{\gamma}(0)$ can be expressed [69] as

$$\gamma(t) = \exp_{\mathbf{P}}[t\dot{\gamma}(0)] = \mathbf{P}^{1/2}\exp[(t)\mathbf{P}^{-1/2}\dot{\gamma}(0)\mathbf{P}^{-1/2}]\mathbf{P}^{1/2} \tag{19}$$

which in the case of $t=1$ correspond to the exponential map $\exp_{\mathbf{P}} : T_{\mathbf{P}}\mathcal{M} \rightarrow \mathcal{M}$ with $\gamma(1) = \exp_{\mathbf{P}}(\dot{\gamma}(0))$. The respective logarithm map $\log_{\mathbf{P}} : \mathcal{M} \rightarrow T_{\mathbf{P}}\mathcal{M}$ is defined as

$$\dot{\gamma}(0) = \log_{\mathbf{P}}(\mathbf{Q}) = -\mathbf{P}\log(\mathbf{Q}^{-1}\mathbf{P}) \tag{20}$$

Notice that these operators are point dependent where the dependence is made explicit with the subscript. The geodesic distance $D_{\mathrm{AI}}(\mathbf{P},\mathbf{Q})$ between two points $\mathbf{P},\mathbf{Q} \in \mathbf{S}_d^+$, induced by the Affine-Invariant Riemannian metric, derived from the Fisher information matrix was proved ([70, Theorem: S.T. Jensen]) to be given as

$$D_{\mathrm{AI}}(\mathbf{P},\mathbf{Q}) = \sqrt{\tfrac{1}{2}\operatorname{tr}(\log^2(\mathbf{P}^{-1/2}\mathbf{Q}\mathbf{P}^{-1/2}))} \tag{21}$$

$$\nabla_{\mathbf{P}} D_{\mathrm{AI}}^2(\mathbf{P},\mathbf{Q}) = \mathbf{P}\log(\mathbf{Q}^{-1}\mathbf{P}). \tag{22}$$

This metric exhibits all the properties necessary to be a true metric such that, positivity, symmetry, triangle inequality and is also affine invariant and invariant under inversion. The gradient of the squared geodesic distance $\nabla_{\mathbf{P}} D_{\mathrm{AI}}^2(\mathbf{P},\mathbf{Q})$, is equal to the negative of the initial velocity $\dot{\gamma}(0)$ that define the geodesic [69,2].

Using this metric if $N>2$, a closed-form expression for the mean $\overline{\mathbf{T}}_{\mathrm{AI}}$ of a set of N tensors $\{\mathbf{T}_i\} \in \mathbf{S}_d^+$ cannot be obtained. The mean is only *implicitly* defined based on the fact that the Riemannian barycenter exists and is unique for nonpositive sectional curvature manifolds, which is the case of the manifold $\mathbf{S}_d^+$. In the literature [12,13,9], this problem is solved iteratively using a gradient descent algorithm given by

$$\overline{\mathbf{T}}_{\mathrm{AI}}^{t+1} = \exp_{\overline{\mathbf{T}}_{\mathrm{AI}}^t}(u) \tag{23}$$

$$u = -\frac{1}{N}\sum_{i=1}^{N} \nabla_{\overline{\mathbf{T}}_{\mathrm{AI}}^t} D_{\mathrm{AI}}^2(\overline{\mathbf{T}}_{\mathrm{AI}}^t,\mathbf{T}_i) = -\frac{1}{N}\overline{\mathbf{T}}_{\mathrm{AI}}^t \sum_{i=1}^{N} \log(\mathbf{T}_i^{-1}\overline{\mathbf{T}}_{\mathrm{AI}}^t) \tag{24}$$

The algorithm is based on the minimization of the variance. This boils down to evolving an initial guess of the mean along the geodesics with a velocity given by the gradient of the variance (tangent vector u).

5.3. Log-Euclidean metric (LE)

We now present the framework for the tensor space endowed with the Log-Euclidean metric [14]. Contrary to the Affine-Invariant, the Log-Euclidean metric induces a space with a null curvature. By trying to put a Lie group structure on the tensor space, Arsigny et al. [14] observed that the matrix exponential is a *diffeomorphism* (a one-to-one, continuous, differentiable mapping with a continuous, differentiable inverse) from the space of symmetric matrices $\mathbf{S}_d$ to the tensor space $\mathbf{S}_d^+$.

The important point here is that the logarithm of a tensor $\mathbf{P} \in \mathbf{S}_d^+$ is unique, well defined and is a symmetric matrix $u=\log(\mathbf{P})$. Conversely, the matrix exponential of any symmetric matrix u yields a tensor $\mathbf{P}=\exp(u)$, i.e. each symmetric matrix is associated to a tensor by the matrix exponential.

Thus, one can seamlessly transport all the operations defined in the vector space of symmetric matrices $\mathbf{S}_d$ to the tensor space $\mathbf{S}_d^+$, i.e. since there is a one-to-one mapping between the tensor space and the vector space of symmetric matrices, one can transfer to tensors the standard algebraic operations (addition $+$ and scalar multiplication $\cdot$) with the matrix exponential. This defines on tensors the *logarithmic multiplication* $\odot$ and the *logarithmic scalar multiplication* $\circledast$, given by

$$\mathbf{P} \odot \mathbf{Q} = \exp[\log(\mathbf{P}) + \log(\mathbf{Q})] \tag{25}$$

$$\lambda \circledast \mathbf{P} = \exp[\lambda \cdot \log(\mathbf{P})] = \mathbf{P}^{\lambda} \tag{26}$$

The operator $\odot$ is *commutative* and coincides with matrix multiplication whenever the two tensors $\mathbf{P},\mathbf{Q} \in \mathbf{S}_d^+$ commute in the matrix sense. With $\odot$ and $\circledast$ the tensor space $\mathbf{S}_d^+$ has by construction a commutative *Lie Group Structure*, i.e a space that is both a smooth manifold and a group in which algebraic operations (multiplication and inversion) are smooth mappings and a *Vector Space Structure*, which is not the usual structure directly inherited from square matrices. Here, the smoothness of $\odot$ comes from the fact that both the exponential and the logarithm mappings are smooth.

The operator $\odot$ gives a commutative Lie Group Structure to the tensors, for which any metric at the tangent space at the *identity* is extended into a bi-invariant Riemannian metric on the tensor manifold (metrics that are invariant by multiplication and inversion), e.g. the Euclidean metric on symmetric matrices is transformed into a *bi-invariant* Riemannian metric on the tensor manifold. Among Riemannian metrics in Lie groups, the most suitable in practice, when they exist, are bi-invariant metrics. These metrics are used in differential geometry to generalize to Lie groups a notion of mean that is consistent with multiplication and inversion. For our tensor Lie group, bi-invariant metrics exist and are particularly simple. Their existence simply results from the *commutativity* of logarithmic multiplication between tensors, and since they correspond to Euclidean metrics in the domain of logarithms they are called *Log-Euclidean* metrics.

By adding the operator $\circledast$, we get a complete structure of vector space on tensors. This means that most of the operations

that were generalized using minimizations for the Affine-Invariant metric do have a closed-form with a Log-Euclidean metric. Hence, the Riemannian framework for statistics is extremely simplified. Results obtained on logarithms are mapped back to the tensor domain with the exponential.

In the Log-Euclidean framework, the inner product $\langle u,v\rangle_{\mathbf{P}}$ for any tangent vectors $u,v\in\mathbf{S}_d$, in the tangent space $T_{\mathbf{P}}\mathcal{M}$, relative to the point $\mathbf{P}$ is given by

$$\langle u,v\rangle_{\mathbf{P}}=\langle \partial_{\mathbf{P}}\log\cdot u,\partial_{\mathbf{P}}\log\cdot v\rangle_{\mathbf{Id}} \tag{27}$$

The operator $\partial_{\mathbf{P}}\,log\cdot$ correspond to the *differential* of the matrix logarithm. Let $\gamma:[0,1]\subset\Re\rightarrow\mathcal{M}$ be a curve in $\mathbf{S}_d^+$, with $\gamma(0)=\mathbf{P}$ and $\gamma(1)=\mathbf{Q}$, $\forall\mathbf{P},\mathbf{Q}\in\mathbf{S}_d^+$. The geodesic defined by the point $\gamma(0)=\mathbf{P}$ and the tangent vector $\dot\gamma(0)$ can be expressed as

$$\gamma(t)=\exp_{\mathbf{P}}[t\dot\gamma(0)]=\exp[\log(\mathbf{P})+\partial_{\mathbf{P}}\log\cdot[t\dot\gamma(0)] \tag{28}$$

which in case of $t=1$ correspond to the exponential map $\exp_{\mathbf{P}}:T_{\mathbf{P}}\mathcal{M}\rightarrow\mathcal{M}$ with $\gamma(1)=\exp_{\mathbf{P}}(\dot\gamma(0))$. The respective logarithm map $\log_{\mathbf{P}}:\mathcal{M}\rightarrow T_{\mathbf{P}}\mathcal{M}$ is defined as

$$\dot\gamma(0)=\log_{\mathbf{P}}(\mathbf{Q})=\partial_{\log(\mathbf{P})}\exp\cdot[\log(\mathbf{Q})-\log(\mathbf{P})] \tag{29}$$

where $\partial_{\mathbf{P}}\exp\cdot$ correspond to the *differential* of the matrix exponential. Since the Log-Euclidean metrics correspond to Euclidean metrics in the logarithms domain, the shortest path between the tensors $\mathbf{P}$ and $\mathbf{Q}$ is a straight line in that domain. Hence, the interpolation between tensors is simplified, and is expressed as

$$\gamma(t)=\exp[(1-t)\log(\mathbf{P})+t\log(\mathbf{Q})] \tag{30}$$

The geodesic distance $D_{\mathrm{LE}}(\mathbf{P},\mathbf{Q})$ between the points $\mathbf{P},\mathbf{Q}\in\mathbf{S}_d^+$, induced by this metric is also extremely simplified as follows:

$$D_{\mathrm{LE}}(\mathbf{P},\mathbf{Q})=\sqrt{\mathrm{tr}[(\log(\mathbf{Q})-\log(\mathbf{P}))^2]} \tag{31}$$

As one can see, the Log-Euclidean distance is much simpler than the equivalent Affine-Invariant distance where matrix multiplications, square roots, and inverses are used. The greater simplicity of Log-Euclidean metrics can also be seen from the mean in the tensor space. In this case, the mean of a set of N tensors $\{\mathbf{T}_i\}\in\mathbf{S}_d^+$ is a direct generalization of the geometric mean of positive numbers and is given *explicitly* by

$$\overline{\mathbf{T}}_{\mathrm{LE}}=\exp\frac{1}{N}\sum_{i=1}^{N}\log(\mathbf{T}_i) \tag{32}$$

This closed form equation makes the computation of Log-Euclidean means straightforward. Practically, one simply use the usual tools of Euclidean statistics on the logarithms and map the results back to the tensor vector space with the exponential. This is theoretically fully justified because the tensor Lie group endowed with a bi-invariant metric (i.e. here a Log-Euclidean metric) is *isomorphic*, *diffeomorphic* and *isometric* to the additive group of symmetric matrices [14]. In terms of elementary operations like distance, geodesics and means, the Log-Euclidean provides much simpler formulae than in the Affine-Invariant case. However, we see that the exponential/logarithm mappings are complicated in the Log-Euclidean case by the use of the differentials of the matrix exponential/logarithm. For general matrices, one has to compute the series

$$\partial_{\mathbf{P}}\exp\cdot(u)=\sum_{k=1}^{+\infty}\frac{1}{k!}\left[\sum_{i=0}^{k-1}u^i\mathbf{P}u^{(k-i-1)}\right] \tag{33}$$

This cost would probably be prohibitive if we had to rely on numerical approximation methods. However, in the case of symmetric matrices, the differential is simplified. Using spectral properties of symmetric matrices, one can compute an explicit and very simple/efficiently closed-form expression for the differential of both matrix logarithm and exponential (see [8]). Let $u=\mathbf{R}\mathbf{D}\mathbf{R}^T$, where $\mathbf{D}$ is a diagonal matrix, and consider $\mathbf{Z}=\mathbf{R}\mathbf{P}\mathbf{R}^T$. As $\mathbf{D}$ is diagonal, one can access the (l,m) coefficient of the resulting matrix as

$$\partial_{\mathbf{P}}\exp\cdot(u)=\mathbf{R}^T\partial_{\mathbf{Z}}\exp\cdot(\mathbf{D})\mathbf{R} \tag{34}$$

$$[\partial_{\mathbf{Z}}\exp\cdot(\mathbf{D})]_{(l,m)}=\frac{\exp(d_l)-\exp(d_m)}{d_l-d_m}[\mathbf{Z}]_{(l,m)} \tag{35}$$

6. Background modeling: non-parametric

In this section, we present a proper derivation of the KDE on general Riemannian manifolds, mathematically defined by Pelletier [17] and we extend this concept to the tensor manifold ($\mathbf{S}_d^+$). The tensor manifold is endowed with two Riemannian metrics, i.e. Affine-Invariant and Log-Euclidean, and with the standard Euclidean metric to prove the benefits of take into account the Riemannian structure of the manifold.

The kernel density estimator is the most widely used practical method for nonparametrically estimate the underlying density of a random sample on $\Re^n$. By placing a smooth kernel, the resulting estimator will have a smooth density estimate. Sample spaces with a more complex intrinsic structure than the Euclidean space (e.g. Riemannian structure) arise in a variety of contexts and motivate the adaptation of popular nonparametric estimation techniques on $\Re^n$. However, applying a nonparametric approach outside Euclidean spaces is not trivial and requires careful use of the differential geometry.

One question arise: How to choose the metric depending on the nature and natural properties of the data that need to process? Following Pennec's research on medical imaging processing [71], the Affine-Invariant and Log-Euclidean metrics seem to be well adapted for DTIs and covariance matrices, providing powerful tools to process tensors (e.g. normal law, mean, interpolation, filtering, smoothing). Null and negatives eigenvalues are at an infinite distance of any tensor, so there is no risk to reach them in a finite time and gradient descents algorithms are well posed. The Affine-Invariant metrics gives to the tensor manifold a Hadamard structure (a space with non-positive curvature which is diffeomorphic to $\Re^n$) while the Log-Euclidean ones give a complete Euclidean structure. With both metrics, the mean always exists and is unique. The characteristic swelling effect problem [71,14], observed when the tensor manifold is endowed with the standard Euclidean metric, disappears using both Riemannian metrics. Thus, it seems that both Riemannian metrics fit into the same problems.

However, Arsigny [14] showed that in the DTI processing, applying the standard tools to process tensors (e.g. mean, interpolation, filtering, smoothing) using the Log-Euclidean metric, gives as output tensors more anisotropic that their Affine-Invariant counterparts. On the other hand, Arsigny [14] also showed that from a numerical point of view the computation of those tools using the Log-Euclidean is not only faster but also more stable than in the Affine-Invariant case. This property can be crucial in applications where large amounts of data are processed.

Over the years, the researchers showed that there is not an universal metric for one type of features (tensors): there are different families of metrics with similar or different characteristics, and one may rely on one or the other depending on the specificities of the application [71,14].

6.1. Non-parametric: intrinsic

Frequentist methods for nonparametric estimation on non-Euclidean spaces have been developed by Pelletier [17]. In [17], an

appropriate kernel method is presented on general Riemannian manifolds, which generalize the commonly used location-scale kernel on Euclidean spaces. Pelletier's idea was to build an analogue of a kernel on $\mathcal{M}$ by using a positive function of the *geodesic distance* on $\mathcal{M}$, which is then normalized by the *volume density function* to take into account the curvature. Pelletier's estimator is consistent with standard kernel estimators on Euclidean spaces $\Re^n$. It converges at the same rate as the Euclidean kernel estimator. Provided the bandwidth is small enough, the kernel is centered on the observation, i.e. the observation is an intrinsic mean of its associated kernel.

Consider a probability distribution with a density f on a Riemannian manifold $(\mathcal{M},g)$ and let $\{\mathbf{Z}_1,\ldots,\mathbf{Z}_N\}$ be N i.i.d. random objects on $\mathcal{M}$ with density f. The density estimator of f is defined as the map $f_{N,K}M0003A;\mathcal{M}\to\Re_+$ which, to each $\mathbf{Z}\in\mathcal{M}$, associates the value $f_{N,K}(\mathbf{Z})$ given by

$$f_{N,K}(\mathbf{Z})=\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\theta_{\mathbf{Z}_i}(\mathbf{Z})}\frac{1}{h^n}K\left(\frac{D(\mathbf{Z},\mathbf{Z}_i)}{h}\right) \tag{36}$$

where $D(\mathbf{Z},\mathbf{Z}_i)$ is the geodesic distance between points $\mathbf{Z},\mathbf{Z}_i\in\mathcal{M}$, $\theta_{\mathbf{Z}_i}(\mathbf{Z})$ is the volume density function, (n) is the manifold dimension, (h) is the bandwidth or smoothing parameter, (N) is the number of samples and $K(\cdot)$ is a nonnegative function (we define $K(\cdot)$ as the Normal *pdf*).

In a Euclidean space, the integral of the kernel is independent of the point at which it is centered and the density function integrates to one. For a Riemannian manifold, the integral depends on the point at which the kernel it is centered, e.g. depends on the local geometry of $\mathcal{M}$ in a neighborhood of the observation. This is necessary for obtaining an estimator which is consistent with kernel estimators on Euclidean space, and which possesses the same properties under a similar bunch of assumptions.

It is possible to ensure that the integral is the same irrespective of where it is centered by using the volume density function, i.e. measuring how much n-dimensional volumes in regions of an n-dimensional Riemannian manifold differ from the volumes of equivalent regions in $\Re^n$.

For $\mathbf{P},\mathbf{Q}\in\mathcal{M}$, the volume density function $\theta_{\mathbf{P}}(\mathbf{Q})$ on $\mathcal{M}$ is defined by ([72, p. 174])

$$\theta_{\mathbf{P}}:\mathbf{Q}\to\theta_{\mathbf{P}}(\mathbf{Q})=\frac{\mu_{\exp_{\mathbf{P}}^{*}g}}{\mu_{g_{\mathbf{P}}}}(\exp_{\mathbf{P}}^{-1}\mathbf{Q}) \tag{37}$$

which is the quotient of the canonical measure of the Riemannian metric $\exp_{\mathbf{P}}^{*}g$ on $T_{\mathbf{P}}\mathcal{M}$ (pullback of the metric-tensor g by the exponential-map $\exp_{\mathbf{P}}$) by the Lebesgue measure of the Euclidean structure $g_{\mathbf{P}}$ on $T_{\mathbf{P}}\mathcal{M}$. In other words, if $\mathbf{Q}$ belongs to a normal neighborhood of $\mathbf{P}$, then $\theta_{\mathbf{P}}(\mathbf{Q})$ is the density of the pullback of the volume measure on $\mathcal{M}$ to $T_{\mathbf{P}}\mathcal{M}$ with respect to the Lebesgue measure on $T_{\mathbf{P}}\mathcal{M}$ via the inverse exponential-map $\exp_{\mathbf{P}}^{-1}$.

In terms of geodesic normal coordinates at $\mathbf{P}$, $\theta_{\mathbf{P}}(\mathbf{Q})$ equals the *square-root of the determinant of the metric-tensor* g expressed in these coordinates at $\exp_{\mathbf{P}}^{-1}\mathbf{Q}$. Let $G_{\mathbf{P}}=[g_{ij}]_{\mathbf{P}}$ be the local representation of the Riemannian metric (Section 4), if $y=\varphi(\mathbf{Q})=(y^1,\ldots,y^n)^T$ denotes the normal coordinates of $\mathbf{Q}$ in a normal coordinate system centered at $\mathbf{P}$ then $\theta_{\mathbf{P}}(\mathbf{Q})=(\sqrt{|G_{\mathbf{P}}(y)|})$. In a normal neighborhood, θ is strictly positive and $\theta_{\mathbf{P}}(\mathbf{Q})=\theta_{\mathbf{Q}}(\mathbf{P})$ [73,17].

6.1.1. Intrinsic: Euclidean metric

The distance $D_{\mathrm{E}}(\mathbf{P},\mathbf{Q})$ between tensors $\forall\mathbf{P},\mathbf{Q}\in\mathbf{S}_d^+$, induced by the Euclidean metric is given by Eq. (12).

Square-root determinant metric $(\sqrt{|G_{\mathbf{P}}(y)|})$: As mentioned previously, Pelletier's estimator is consistent with kernel estimators on Euclidean spaces, i.e. when $\mathcal{M}$ is the Euclidean space $\Re^n$, the estimator expression reduces to the one of a standard kernel estimator on $\Re^n$ [17]. Consider that $(\mathcal{M},g)$ corresponds to the Euclidean space $(\Re^n,\delta)$, where δ denotes the usual canonical Euclidean metric, and consider the canonical identification of the tangent space $T_{\mathbf{P}}\mathcal{M}$ at some point $\mathbf{P}$ of $(\Re^n,\delta)$, with $\Re^n$. Note that any two tangent spaces at different points on the manifold are also canonically identified. This defines trivially a normal chart, the domain of which is the entire manifold. In this chart, the components of the metric tensor form the identity matrix, hence $\forall\mathbf{P},\mathbf{Q}\in\mathcal{M}$ the calculus of the $\theta_{\mathbf{P}}(\mathbf{Q})$ is simplified to $\sqrt{|G_{\mathbf{P}}(y)|}=1$ (when the space is flat the volume density function is unity everywhere [72, p. 154]).

6.1.2. Intrinsic: Affine-Invariant metric

The geodesic distance $D_{\mathrm{AI}}(\mathbf{P},\mathbf{Q})$ between two tensors $\forall\mathbf{P},\mathbf{Q}\in\mathbf{S}_d^+$, induced by the Affine-Invariant Riemannian metric, derived from the Fisher information matrix, is given by Eq. (21).

Square-root determinant metric $(\sqrt{|G_{\mathbf{P}}(y)|})$: Generalizing the *pdf* concept requires a measure $d\mathcal{M}$ on the manifold which, in case of Riemannian manifolds, is induced in a natural way by the metric $G(x)$ for a given local coordinate system [74]. As any metric in an Euclidean space, the Riemannian metric induces an infinitesimal volume element $d\mathcal{M}(x)=\sqrt{|G(x)|}\,dx$ in any chart (volume of the parallelepiped spanned by the vectors of an orthonormal basis of the tangent space). The difference is that the measure is now different at each point since the local expression of the metric is changing. The *reference measure* $d\mathcal{M}(x)$ on the manifold can be used to measure *random events* on the manifold (generalization of random variables), and to define their *pdf* (if it exists), i.e. the function $p(x)$ on the manifold such that the respective *probability measure* is given by $dP(x)=p(x)d\mathcal{M}(x)$. The induced measure $d\mathcal{M}$ actually represents the notion of *uniformity* according to the chosen Riemannian metric. With the probability measure dP of a random element, we can integrate functions $\phi(x)$ from the manifold to any vector space, thus defining the expected value of this function. This notion of expectation corresponds to the one we defined on real random variables and vectors. Seeing that the Taylor expansion of the metric was defined in [75], Pennec [74] used the Taylor expansion of the measure $d\mathcal{M}$ [76] in a normal coordinate system around the mean value to generalized a Normal law to Riemannian manifolds. In our case, we consider the normal coordinate system around $\mathbf{P}\in\mathcal{M}$. The Taylor expansion of the measure $d\mathcal{M}$ around the origin is given as

$$d\mathcal{M}_{\mathbf{P}}(y)=(\sqrt{|G_{\mathbf{P}}(y)|})\,dy\approx\left(1-\frac{y^T\mathcal{R}y}{6}\right)dy \tag{38}$$

where y is the normal coordinates of $\mathbf{Q}\in\mathcal{M}$ and $\mathcal{R}$ is the Ricci tensor in the considered normal coordinate system. The expression for the volume density function given by Eq. (37) was deduced in ([77, p. 169]), which is equal to the expression showed in Eq. (38). To define the Ricci tensor for the tensor space $\mathbf{S}_d^+$, we have to choose an affine connection, since this will influence the curvature properties. The existence/uniqueness of the Riemannian barycenter requires that the space exhibit a non-positive sectional curvature.

The canonical affine connection on a Riemannian manifold is known as the Levi-Civita connection (or covariant derivative). It is the only one to be compatible with the metric (covariant derivative of the metric is zero), i.e. the only one by which the parallel transport of a vector does not affect its length. Therefore, we will work with the Levi-Civita connection in the remaining developments. Using the local coordinates, the Christoffel symbols of the

second kind [12] for the tensor space $\mathbf{S}_d^+$ can also be expressed in terms of the elements of the canonical and dual basis $\{E_i\}_{i=1,\ldots,n}$ and $\{E_i^*\}_{i=1,\ldots,n}$

$$\Gamma_{ij}^k = \Gamma(E_i,E_j;E_k^*) = E_k^* \cdot (\nabla_{E_i}^F E_j) \tag{39}$$

$\forall i,j,k=1,\ldots,n$. Provided that [12] ($\mathbf{P}\in\mathbf{S}_d^+$):

$$\frac{\partial g(E_i,E_j)}{\partial x_k} = -\frac{1}{2}\,\mathrm{tr}(\mathbf{P}^{-1}E_k\mathbf{P}^{-1}E_i\mathbf{P}^{-1}E_j) - \frac{1}{2}\,\mathrm{tr}(\mathbf{P}^{-1}E_i\mathbf{P}^{-1}E_k\mathbf{P}^{-1}E_j) \tag{40}$$

the unique affine connection (Levi-Civita) associated with the Fisher information metric was derived from Eq. (7) as

$$\Gamma(E_i,E_j;E_k^*) = -\tfrac{1}{2}\,\mathrm{tr}(E_i\mathbf{P}^{-1}E_jE_k^*) - \tfrac{1}{2}\,\mathrm{tr}(E_j\mathbf{P}^{-1}E_iE_k^*) \tag{41}$$

Riemannian curvature tensor ($\mathbf{R}$): As shown in [12], the Riemann curvature tensor ($\mathbf{R}$) for the tensor space $\mathbf{S}_d^+$, derived from the Fisher information metric, and the classical Levi-Civita affine connection, is given by

$$\begin{aligned}\mathbf{R}_{ijkl} &= \mathbf{R}(E_i,E_j,E_k,E_l)\\ &= \tfrac{1}{4}\,\mathrm{tr}(E_j\mathbf{P}^{-1}E_i\mathbf{P}^{-1}E_k\mathbf{P}^{-1}E_l\mathbf{P}^{-1})\\ &\quad -\tfrac{1}{4}\,\mathrm{tr}(E_i\mathbf{P}^{-1}E_j\mathbf{P}^{-1}E_k\mathbf{P}^{-1}E_l\mathbf{P}^{-1})\end{aligned} \tag{42}$$

Ricci curvature tensor ($\mathcal{R}$): The Ricci tensor ($\mathcal{R}$) calculus is performed on the basis of closed-form expressions for the metric and the Riemann tensor $\mathbf{R}$ (Eq. (9)) and simply involves traces of matrix products. Symbolic computations easily lead to the components of the Ricci in terms of the components of $\mathbf{P}^{-1}$. Comparing the Ricci ($\mathcal{R}$) with the metric, we confirm that the tensor space endowed with this metric is not an *Einstein* manifold [10,12] i.e. it is a space of non-constant non-positive curvature, for which there does not exist a constant L such that $\mathcal{R}_{ij}=Lg_{ij}$. Therefore, we need take into account the Riemannian tensor ($\mathbf{R}$) to deal with the curvature.

6.1.3. Intrinsic: Log-Euclidean metric

The geodesic distance $D_{\mathrm{LE}}(\mathbf{P},\mathbf{Q})$ between tensors $\forall\mathbf{P},\mathbf{Q}\in\mathbf{S}_d^+$, induced by the Log-Euclidean metric, is given by Eq. (31).

Square-root determinant metric ($\sqrt{|G_{\mathbf{P}}(y)|}$): It was proved in [14] that the Lie group of the $\mathbf{S}_d^+$ matrices is *isomorphic* (algebraic structure of vector space is conserved) and *diffeomorphic* to the additive group of symmetric matrices $\mathbf{S}_d$. It was also proved that the Lie group of $\mathbf{S}_d^+$ matrices endowed with a Log-Euclidean metric is also *isometric* (distances are conserved) to the space of symmetric matrices $\mathbf{S}_d$ endowed with the associated Euclidean metric. The Log-Euclidean metric induces on the tensor space $\mathbf{S}_d^+$ a space with a null curvature, i.e. endowed with the Log-Euclidean metric, the tensor space $\mathbf{S}_d^+$ is a flat Riemannian space (its sectional curvature (see [76, p. 107]) is null everywhere). As proved in ([72, p. 154]), when the Riemannian space is flat the volume density function is unity everywhere. Analyzing the problem from a different perspective, consider that the volume density function is equal to the square-root of the determinant of the metric-tensor (Section 6.1). The underlying isometry of the Log-Euclidean metric result in a metric tensor that is in fact a orthogonal matrix and hence the determinant of the metric tensor is always equal to one [63]. Taking into account these facts, the calculus $\forall\mathbf{P},\mathbf{Q}\in\mathbf{S}_d^+$ of the volume density function $\theta_{\mathbf{P}}(\mathbf{Q})$ is extremely simplified to $(\sqrt{|G_{\mathbf{P}}(y)|}=1$.

6.2. Non-parametric: extrinsic

The KDE was *intrinsically* formulated in Section 6.1 to operate on the tensor manifold $\mathbf{S}_d^+$. Depending on the metric chosen, that approach specifically requires the computation of the volume density function, which can be hard to carry out for some applications. Therefore, it would be interesting to *extrinsically* reformulate the KDE on the tensor manifold and evaluate its performance and efficiency.

In this section, we will analyze the feasibility of designing an extrinsic KDE to operate on the tensor manifold $\mathbf{S}_d^+$ endowed with the two Riemannian metrics (i.e. Affine-Invariant and Log-Euclidean). The extension is extrinsic in the sense that the inherent density estimation is performed on the tangent spaces. By first mapping the data to a tangent space, that is a vector space, we can use a standard Euclidean KDE approach [15,16]. We start by defining mappings from neighborhoods on the manifold to the Euclidean space, similar to coordinate charts. Our maps are the logarithm maps $\log_{\mathbf{P}}$ that map the neighborhood of points $\mathbf{P}\in\mathcal{M}$ to the tangent space $T_{\mathbf{P}}\mathcal{M}$. Since this mapping is a homeomorphism around the neighborhood of the point, the manifold structure is locally preserved. This requires choosing a suitable tangent space on which to map. In this work, the data was mapped onto the tangent space at the mean point of the samples data. Since the Karcher mean $\mu\in\mathcal{M}$ of a set of points on the Riemannian manifold is the point on $\mathcal{M}$ that minimizes the sum of squared Riemannian distances [9], and the mapping preserves the structure of the manifold locally, the tangent plane at the mean μ is a good choice. This procedure can be seen as a way of linearizing the manifold around the mean point μ since the tangent space $T_\mu\mathcal{M}$ provides a first order approximation of the manifold around μ. Basically, this is the same as consider a normal coordinate system $(\mathcal{U},\varphi)$ around the mean point μ.

At some time t, let $\{\mathbf{Z}_i\}_{i=1,\ldots,N}$ be the set of N points on $\mathcal{M}$ (past samples or observations) and $\mathbf{Z}_0\in\mathcal{M}$ is the actual sample that we want to classify. First, we compute the mean $\mu_t\in\mathcal{M}$ of the all samples $\{\mathbf{Z}_i\}_{i=0,\ldots,N}$. Then, we map (project) all the points $\{\mathbf{Z}_i\}_{i=0,\ldots,N}$ to the tangent space $T_{\mu_t}\mathcal{M}$ using the logarithm map $\log_{\mu_t}(\mathbf{Z}_i)$, $i=0,\ldots,N$. Let $z_i=\varphi(\mathbf{Z}_i)=(z_i^1,\ldots,z_i^n)^T$ denote the normal coordinates of $\mathbf{Z}_i$, $\forall i=0,\ldots,N$ in the normal coordinate system at μ_t. Seeing that the normal coordinate system defines a vector space, we can apply the standard Euclidean KDE on $\Re^n$ [15,16].

6.2.1. Extrinsic: Affine-Invariant metric

As had been pointed out in Section 5.2, by using the Affine-Invariant metric a closed-form expression for the mean on the tensor manifold ($\mathbf{S}_d^+$) cannot be obtained. The mean is only *implicitly* defined since the Riemannian barycenter exists and is unique for nonpositive sectional curvature manifolds. The gradient descent algorithm presented in Section 5.2 essentially alternates the computation of the barycenter in the exponential chart centered at the current estimation of the mean value, and performs a re-centering step of the chart at the point of the manifold that corresponds to the computed barycenter (geodesic marching step). An exact implementation of this iterative algorithm can be a costly procedure.

In order to speed up the process, we will use a method based on a online K-means on the tensor manifold (endowed with the Affine-Invariant metric) proposed by Caseiro et al. [11]. At each frame (time t), the mean value $\mu_t\in\mathbf{S}_d^+$ is updated using a learning rate (ρ). The new mean μ_t combine the prior information $\mu_{t-1}\in\mathbf{S}_d^+$ with the actual sample $\mathbf{Z}_0\in\mathbf{S}_d^+$. To take into account the Riemannian geometry of the manifold $\mathbf{S}_d^+$, Caseiro et al. [11] derived an approximation equation to update the tensor mean, based on the concept of interpolation between tensors. The interpolation can be seen as a walk along the geodesic joining the tensors. After some mathematical simplifications [11] the mean update equation turns into

$$\mu_t = (\mu_{t-1})^{(1-\rho)/2}(\mathbf{Z}_0)^{\rho}(\mu_{t-1})^{(1-\rho)/2} \tag{43}$$

It is clear that this KDE extrinsic formulation is much simpler than the intrinsic counterpart, mainly due to two reasons. Firstly (at each time

t), it is not necessary to compute the Ricci curvature tensor $\mathcal{R}$. Secondly (at each time t), the N distances between tensors are computed in the Euclidean space provided by the tangent space $T_{\mu_t}\mathcal{M}$, i.e. it is not necessary to use the geodesic distance given by Eq. (21).

6.2.2. Extrinsic: Log-Euclidean metric

The tensor manifold $\mathbf{S}_d^+$ endowed with the Log-Euclidean metric is a special case that imposes a more in-depth analysis, i.e. due to the special properties of the Log-Euclidean metric there are two different paradigms that we need to analyze in order to design an extrinsic KDE on the tensor manifold. Basically, we need to define a mapping in order to project the data from the manifold to an Euclidean space. In the case of the Log-Euclidean metric, we have two different ways to project the data to an Euclidean space.

(1) We can use the logarithm map $\log_\mu$ given by Eq. (29) to project the data to the tangent space at the mean point $T_{\mu_t}\mathcal{M}$. Recall that although the logarithm map in the Log-Euclidean case seems complicated by the use of the differential of the exponential matrix, we can compute this differential explicitly in a very simple and closed-form fashion using the Eq. (23).

As in the Affine-Invariant case, the mean point at each time μ_t can be calculated by two different approaches: the first, is to use the Eq. (32) at each time t to compute the mean $\mu_t \in \mathcal{M}$ of the all samples $\{\mathbf{Z}_i\}_{i=0,\ldots,N}$ (actual and past samples).

Recall that the Log-Euclidean metric provides a closed-form solution (Eq. (32)) to compute the mean of a set of samples which is much simpler than the equivalent Affine-Invariant (Eq. (34)); the second, is to use an approach similar to the K-means used in Section 6.2.1 to compute the mean at time t, in a online fashion. In the Log-Euclidean case, we do not use Eq. (43) to combine the prior information $\mu_{t-1} \in \mathbf{S}_d^+$ with the actual sample $\mathbf{Z}_0 \in \mathbf{S}_d^+$, but instead we can use the interpolation equation given by Eq. (30).

Although this extrinsic approach provides a good approximation and is conceptually similar to the extrinsic Affine-Invariant counterpart, in practice, if we analyze and compare this extrinsic algorithm with the intrinsic version (Section 6.1.3) we conclude that the extrinsic is much more complex and naturally is much more time consuming.

By inducing a space with a null curvature, the Log-Euclidean metric considerably simplifies the volume density function computation, meaning that this extrinsic version does not provide any benefits over the intrinsic counterpart.

(2) The Log-Euclidean framework defines a mapping where the tensor space $\mathbf{S}_d^+$ is isomorphic, diffeomorphic, and isometric to the associated Euclidean space of symmetric matrices $\mathbf{S}_d$. This mapping is precisely the simple matrix logarithm ($\log_{\mathbf{I}} \mathbf{P} = \log \mathbf{P}$), $\forall \mathbf{P} \in \mathbf{S}_d^+$, i.e. tensors are transformed into symmetric matrices using $\log \mathbf{P}$ ($\mathbf{I}$ is a identity $d \times d$ matrix). Since the Log-Euclidean transforms Riemannian computations on tensors into Euclidean computations on vectors in the logarithms domain, practically one simply uses the usual tools of Euclidean statistics on the logarithms and maps the results back to the tensor vector space with the exponential. Notice that in practice this extrinsic version in which the mapping is defined by the simple matrix logarithm log is mathematically equivalent to the intrinsic counterpart (Section 6.1.3).

7. Experimental results

In order to evaluate and confirm the effectiveness of the proposed non-parametric framework on tensor field for foreground segmentation, we conduct a considerable number of experiments on a variety of challenging video sequences presented in the previous literature, which include both indoor and outdoor environments with complex backgrounds (e.g. dynamic backgrounds, illumination changes, camera jitters and image noise).

We now present a brief description of each one of the eight sequences used. The sequence 1 (HighWayI) is a highway scenario where the vast majority of car colors are shades of gray (similar to the background). The sequence 2 (Railway) is the moving camera sequence used by Caseiro et al. [2], which involved a camera mounted on a tall tripod. The wind caused the tripod to sway back and forth causing nominal motion of the camera. The sequence 3 (HighWayIII) is a highway scenario where there is typically a steady stream of vehicles. The sequence 4 (HalwayI) shows a busy hallway where people are walking or standing still. The sequence 5 (Campus) is a noisy sequence from outdoor campus site where cars approach to an entrance barrier and students are walking around. The sequence 6 (HighWayII) is a highway scenario where the camera presents some motion and the image is noisy. The sequence 7 (Ducks) is from an outdoor scene that contains two ducks swimming on a pond, with dynamic background composed of subtle illumination variations along with ripples in the water and swaying trees in the upper part of the scene. The sequence 8 (Fountain) is a particularly challenging outdoor situation, with several sources of dynamic motion, e.g. a spouting fountain with nonperiodic motions and the swaying tree branches above.

The sequences 1, 5, 6 are selected from the ATON project (http://cvrr.ucsd.edu/aton/shadow) or VISOR repository (http://www.openvisor.org/) [78]. The sequence 2, 7, 8 are selected from Seikh's work [38] (http://www.cs.cmu.edu/~yaser/). The sequences 3 and 4 are selected from Brisson's work [79] (http://cvrr.ucsd.edu/aton/shadow/). We will perform several experiments using the mentioned sequences in order to compare our non-parametric framework on tensor domain with the appropriate state of the art methods. The main goals of these experiments are as follows:

Goal 1 proves the benefits of the tensor-based methods compared with the standard feature vectorial approaches. The tensor-based methods enable the conversion of the image into a more information rich form (to yield latent discriminating features, e.g. color, gradients, filters responses, etc.) and the integration of spatial texture, considering the correlation between pixels (pixel based and region based information embedded by tensor matrices). We will show that the effective modeling of the spatial correlations of neighbors pixels by the use of these suitable tensor-based descriptors results in a high discriminative power.

Goal 2 demonstrates the advantage of take into account the underlying geometric structure of the tensor manifold. The tensor space is a Riemannian manifold, meaning that the space of tensors do not conform to Euclidean geometry, therefore the standard Euclidean metric is not appropriate in order to exploit all the information presented in the tensor components. We claim and intend to demonstrate that the use of the well-founded differential geometrical properties of the tensor manifold has a deep impact on the tensor statistics and, hence, it is possible to improve dramatically the quality of the segmentation results.

Goal 3 proves the benefits of the proposed non-parametric technique on the tensor domain in more complex scenarios where simple parametric models do not accurately explain the physical processes, i.e. the nonparametric nature of complex, time varying and non-static backgrounds cannot be well modeled by a single or a combination of parametric distributions.

Goal 4 demonstrates that new points of view on the tensor space can lead to significantly simpler computations and therefore faster foreground detection algorithms (as proved previously by Caseiro et al. in [2] for the tensor-based GMM paradigm). Prove

that from a practical point of view the two Riemannian metrics proposed to endow the tensor manifold (Affine-Invariant and Log-Euclidean) yield similar segmentation results despite of the Log-Euclidean metric be much more simple and faster.

Goal 5 demonstrates that the extrinsic reformulation of the KDE on the tensor manifold can be a good option to speed up the density estimation process, particularly in the case when the manifold is endowed with the Affine-Invariant metric, while some of the benefits of the nonparametric estimation are preserved.

The parametric approach proposed by Stauffer [18] (GMM) and the non-parametric counterpart presented by Elgammal [15,16] (KDE) are the two most widely used techniques to foreground detection using vectorial space features. Therefore, we will use these two vectorial methods as baseline to prove the benefits of the tensor-based approaches (goal 1), using two types of features sets, i.e. a set with color data [**r, g, b**] and a set with gray level incremented with gradients [**I, Ix, Iy**]). To the best of our knowledge, Caseiro et al. [2,11] were the only ones to use the paradigm of background modeling on tensor field to foreground segmentation. Therefore, in order to accomplish the goal 3 we will compare our non-parametric framework on the tensor domain (KDE[**T**]) with the parametric counterpart (GMM[**T**]) proposed in [2,11]. In order to prove the benefits of take into account the Riemannian structure of the tensor manifold (goal 2) and to demonstrate that from a practical point of view, the Log-Euclidean metric is the best choice to endow the tensor manifold (goal 4), both the GMM[**T**] and KDE[**T**] frameworks will be tested using the two proposed Riemannian metrics, (i.e. Affine-Invariant (AI) and Log-Euclidean (LE)) and the standard Euclidean metric (E). We also compared the intrinsic tensor-based KDE (KDE-Int[**T**]) with the extrinsic counterpart (KDE-Ext[**T**])—(goal 5). In this case, we only compared the intrinsic with the extrinsic KDE when the tensor manifold is endowed with the Affine-Invariant metric because, as we concluded in Section 6.2.2 the extrinsic KDE in the Log-Euclidean case does not provide any benefits over the intrinsic counterpart.

In this evaluation, we use a tensor in which are encoded the gray level information [**I**] and texture [**Ix, Iy**] features (gradients). This results in a tensor with $d=3$ and $\mathbf{T} \in \mathbf{S}_3^+$, i.e. the tensor manifold is 6-dimensional ($n = 6$) The structure tensor (ST) and the region covariance matrix (RCM) are not specific descriptors, but a scheme for designing descriptors, therefore the advantage of the nonparametric paradigm over the parametric counterpart remains independent of the information included in the tensor, i.e. the proof of concept does not change. It is important to remark that the tensor-based experiments presented in this section (KDE[**T**] and GMM[**T**]) use the same tensor components. The experiments are divided in two parts. In the first part (Experiment 1— Section 7.1), the sequences (1–6) are used to evaluate the foreground segmentation performance of the proposed non-parametric framework using the structure tensor (ST) as feature (see Figs. 2–4 and Tables 1 and 2). In the second part (Experiment 2— Section 7.2), the last four sequences (5–8) are used to evaluate the proposed non-parametric framework using the region covariance matrix (RCM) as feature (see Figs. 5 and 6 and Tables 4–6). The structure tensor and the region covariance matrices are both calculated for each image pixel using a patch with dimension 3×3 ($w = 3$ and $S = 9$). In order to establish the superiority of the tensor observation model, the vectors [**r, g, b**] and [**I, Ix, Iy**]) were generated by integrating over the same regions (patches) used to compute the tensors, as in Eq. (3).

In order to compare the GMM and KDE algorithms, we followed the standard procedures considered in the literature for conduct experimental validations in the vectorial case, namely we followed Elgammal's work [15]. A summary of the significant parameters values used in the experiments is shown in Tables 3 and 7. The GMM algorithm [18] is controlled by four main parameters: maximum number of Gaussian distributions allowed for each pixel (M), number of Gaussian distributions effectively used for each pixel (K), learning rate (α) and threshold (T_g). The KDE algorithm [15] is controlled by five main parameters: type of model (long or short term), number of samples (N), size of the

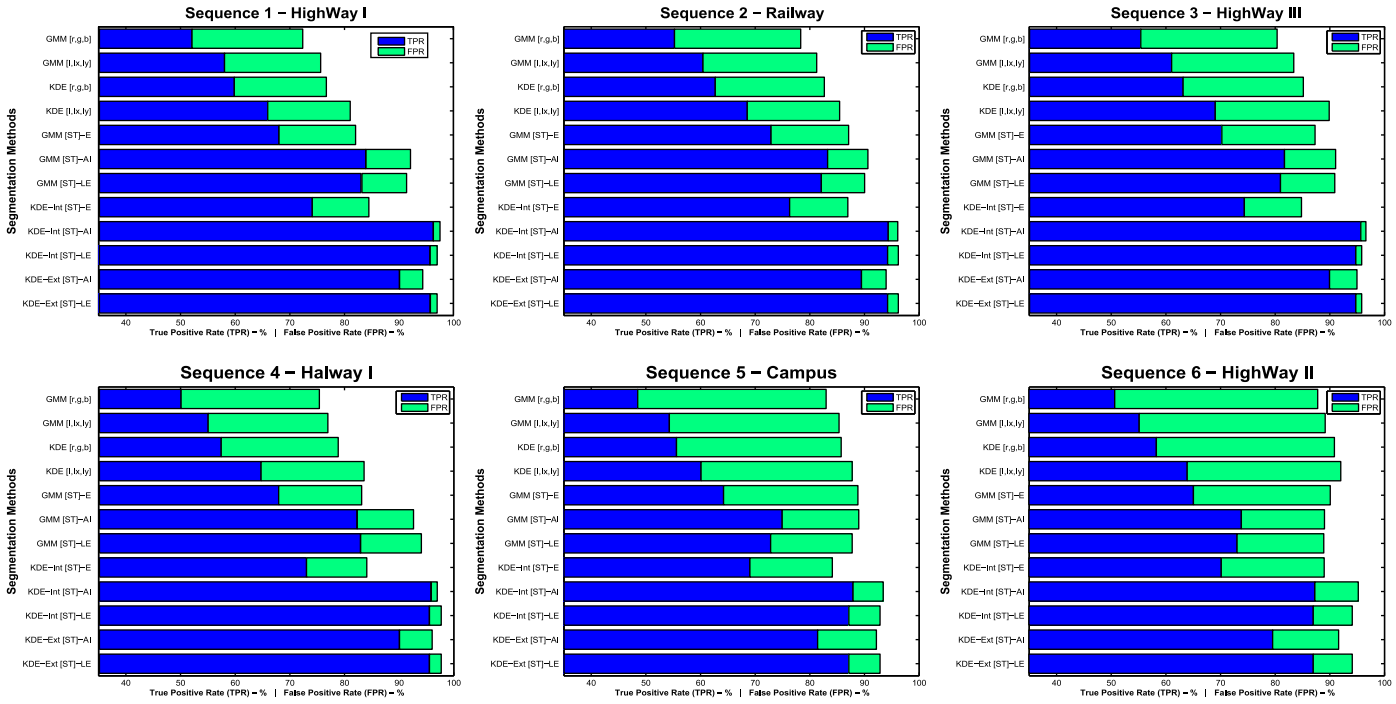

Fig. 2. Experiment 1 ↦ Quantitative performance evaluation on the sequences (1–6), using the structure tensor (ST), in terms of: true positive ratio (TPR) and false positive ratio (FPR) ↦ (E, Euclidean; AI, Affine-Invariant; LE, Log-Euclidean; Int, Intrinsic; Ext, Extrinsic). Notice that this figure contains the same information that in Table 1.

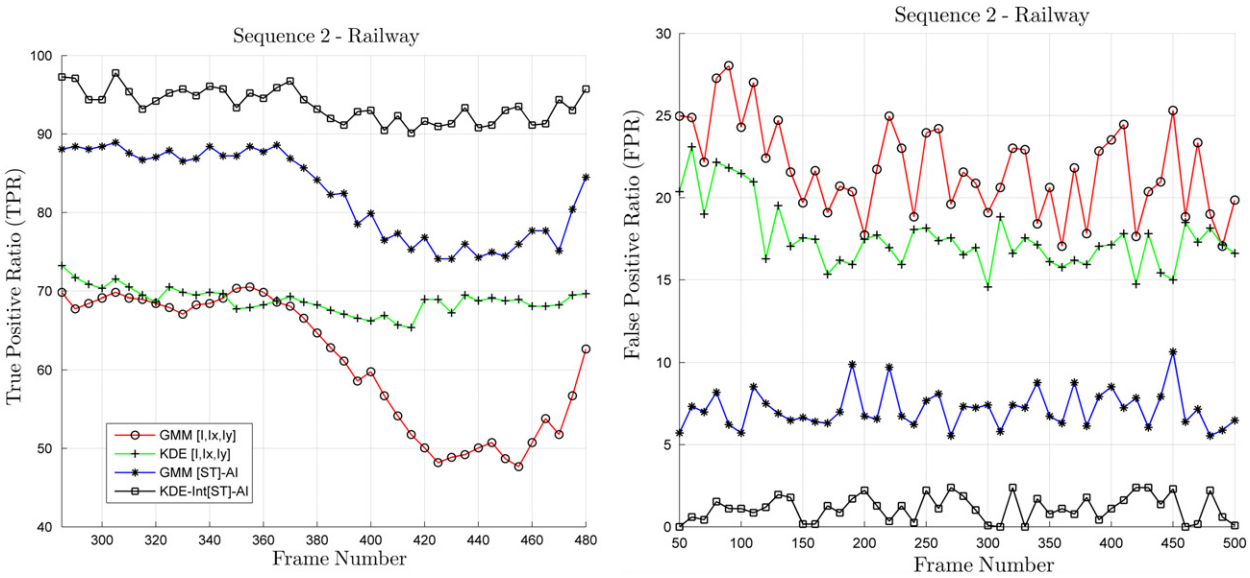

Fig. 3. Experiment 1 ↦ Quantitative evaluation on the sequence 2, using the structure tensor (ST), in terms of: true positive ratio (TPR) and false positive ratio (FPR) ↦ (AI, Affine-Invariant; Int, Intrinsic). Notice that, in the sequence 2 (total=500 frames), the scene is empty (without foreground objects) for the first 276 frames (this sequence is the only one that has groundtruth available for all the frames).

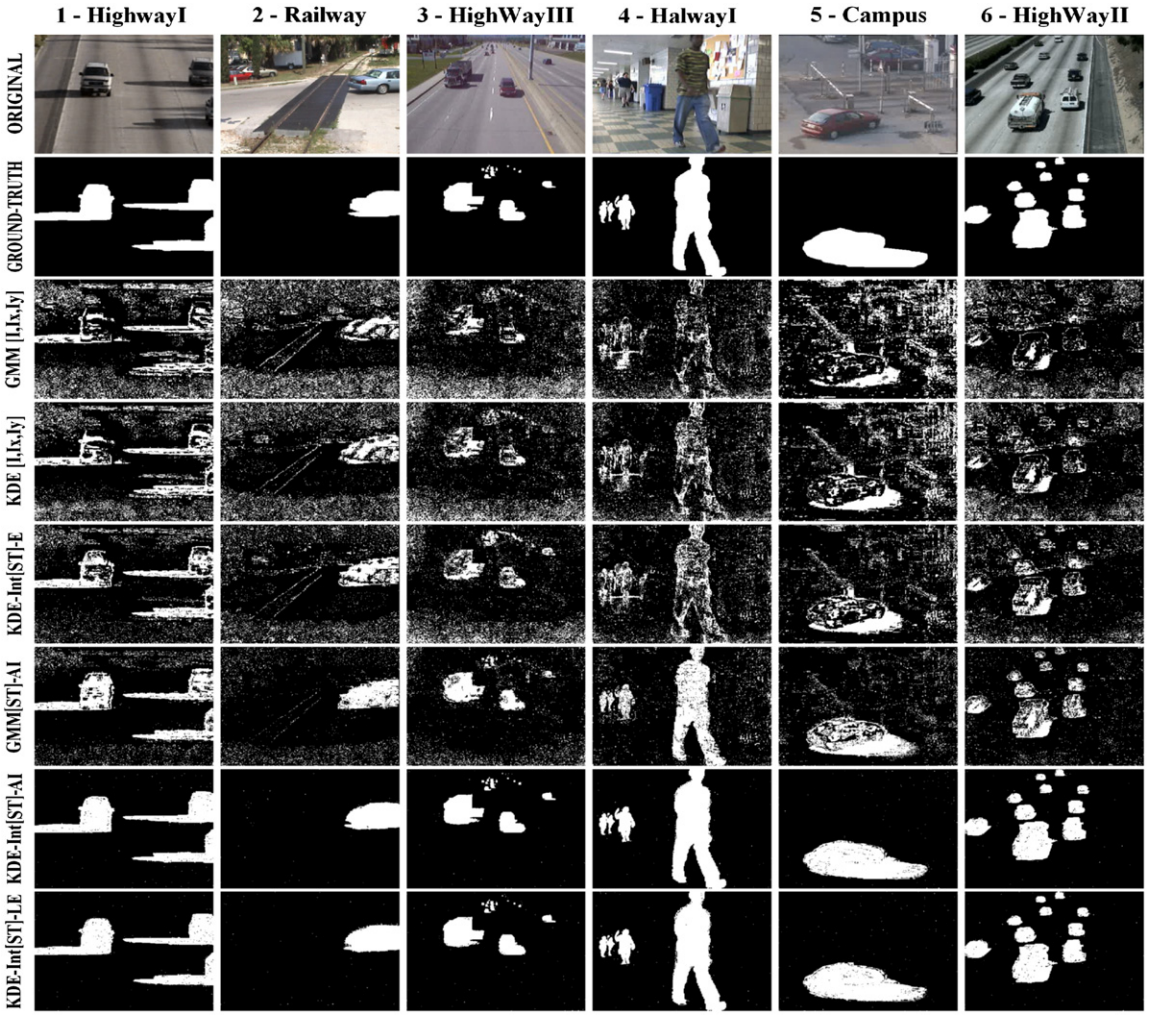

Fig. 4. Experiment 1 ↦ Examples of segmentation results on the sequences (1–6), using the structure tensor (ST) as feature.

Table 1
Experiment 1 ↦ Quantitative performance evaluation on the sequences (1–6), using the structure tensor (ST) as feature, in terms of true positive ratio (TPR) and false positive ratio (FPR) ↦ (E, Euclidean; AI, Affine-Invariant; LE, Log-Euclidean; Int, Intrinsic; Ext, Extrinsic).

Methods	**1-HighWayI**		**2-Railway**		**3-HighWayIII**		**4-HalwayI**		**5-Campus**		**6-HighWayII**		**AVERAGE**	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
GMM[**r**, **g**, **b**]	52.10	20.25	55.23	23.10	55.40	24.95	50.10	25.30	48.50	34.48	50.63	37.14	51.99	27.54
GMM[**I**, **Ix**, **Iy**]	58.05	17.58	60.45	20.80	61.05	22.35	55.05	21.90	54.30	31.05	55.12	34.03	57.34	24.62
KDE[**r**, **g**, **b**]	59.83	16.85	62.70	19.95	63.15	22.00	57.45	21.40	55.60	30.13	58.25	32.58	59.50	23.82
KDE[**I**, **Ix**, **Iy**]	65.95	15.10	68.55	16.90	69.03	20.85	64.73	18.85	60.10	27.65	63.90	28.10	65.38	21.24
GMM[ST]-E	68.02	14.00	72.90	14.20	70.24	17.05	67.95	15.18	64.21	24.59	65.04	25.04	68.06	18.34
GMM[ST]-AI	83.90	07.95	83.25	07.38	81.70	09.35	82.27	10.25	74.94	14.01	73.81	15.20	79.48	10.69
GMM[ST]-LE	83.00	08.21	82.10	07.92	80.96	09.94	82.93	10.96	72.82	14.93	73.02	15.86	78.64	11.30
KDE-Int[ST]-E	74.10	10.36	76.30	10.65	74.35	10.46	73.05	11.03	69.04	15.08	70.13	18.83	72.83	12.74
KDE-Int[ST]-AI	**96.25**	**01.02**	**94.35**	**01.74**	**95.65**	**00.95**	**95.78**	**01.12**	**87.90**	**05.52**	**87.25**	**07.95**	**93.36**	**03.05**
KDE-Int[ST]-LE	**95.64**	**01.17**	**94.23**	**01.96**	**94.75**	**01.08**	**95.53**	**01.95**	**87.14**	**05.71**	**86.97**	**07.13**	**92.88**	**03.17**
KDE-Ext[ST]-AI	90.05	04.10	89.45	04.51	89.95	05.01	90.03	05.93	81.42	10.76	79.53	12.07	86.74	07.06
KDE-Ext[ST]-LE	95.64	01.17	94.23	01.96	94.75	01.08	95.53	01.95	87.14	05.71	86.97	07.13	92.38	03.17

Table 2
Experiment 1 ↦ Comparative performance evaluation between the algorithms, on the sequences (1–6), using the structure tensor (ST). The differential values (Δ) were calculated using the information presented in the column AVERAGE of the Table 1 as (Δ = Methods A−Methods B). Notice that (ΔTPR → +) = Good and (ΔFPR → −) = Good.

Δ = A−B

Methods B	**Methods S**									
	GMM[ST]-AI		GMM[ST]-LE		KDE-Int[ST]-AI		KDE-Int[ST]-LE		KDE-Ext[ST]-AI	
	ΔTPR	ΔFPR	ΔTPR	ΔFPR	ΔTPR	ΔFPR	ΔTPR	ΔFPR	ΔTPR	ΔFPR
GMM[**I, Ix, Iy**]	+22.14	−13.93	–	–	–	–	–	–	–	–
KDE[**I, Ix, Iy**]	–	–	–	–	+27.99	−18.20	–	–	–	–
GMM[ST]-E	+11.92	−07.65	–	–	–	–	–	–	–	–
GMM[ST]-AI	–	–	−00.84	+00.61	+13.89	−07.65	–	–	+06.76	−03.63
GMM[ST]-LE	–	–	–	–	–	–	+14.24	−08.14	–	–
KDE-Int[ST]-E	–	–	–	–	+20.04	−09.69	–	–	–	–
KDE-Int[ST]-AI	–	–	–	–	–	–	−00.49	+00.12	-06.13	+04.01

sampling window (W), threshold (T_k) and bandwidth (h). We remark that, although we define in the GMM case, a relatively high value for (M), almost no pixel reach that maximum at any point of time during the experiments. Please refer to [18,15] for more details about these parameters. For each video sequence, the same parameters were used across all the vectorial and tensor-based GMM/KDE algorithms. We highlight the fact that we compared results using a set of parameters for the KDE paradigm without perform a over tuning of them, against the best parameters found for the GMM technique. This allows us to prove without any doubts the benefits of the tensor-based KDE paradigm over the GMM counterpart.

Note that, in the KDE algorithm [15], given a new pixel sample, there are two alternative mechanisms to update the background (the sample set). *Selective update*: add the new sample to the model (sample set) only if it is classified as a background sample. *Blind update*: just add the new sample to the model (sample set), irrespective of whether it belongs to background or foreground. In both cases, when a new sample is added, the oldest sample is removed from the sample set to ensure that the probability density estimation is based on recent samples. There are tradeoffs to both of these mechanisms and to avoid them two models (short term model and long term model) were proposed and a combination of both was used in [15]. *Short term model*: This is the very recent model of the scene. It adapts to changes quickly to allow very sensitive detection. This model consists of the most recent N background sample values. The sample set is updated using selective update mechanism. *Long term model*: This model captures a more stable representation of the scene background and adapts to changes slowly. It consists of N sample points taken from a larger window in time (with W samples). The sample is updated using a blind-update mechanism.

We did not provide a step-by-step algorithm. But we recall that the idea is to generalize the nonparametric background model proposed by Elgammal [15], from pixel domain to tensor domain. Therefore, using the KDE derivations for the tensor manifold, the algorithm to foreground detection is basically similar to [15]. However, for a fair comparison between the GMM and the KDE paradigms, we did not implement some of the algorithm's stages described by Elgammal in [15]. The framework proposed by Elgammal [15] combine short-term and long-term models to achieve more robust detection results. In our work, we only used one type of model in each video sequence tested. The second stage of Elgammal's framework aims to suppress false detections that are due to small and unmodeled movements in the scene background. Taking into account that step is considered as a postprocessing stage (a kind of spatial filtering), we did not implement it. Finally, we also did not consider the shadows suppress stage proposed in [15]. The goal is only to compare the ability of each paradigm to estimate the underlying density of the data.

The kernel bandwidth (h) was estimated directly from the data of the sample set, following the method proposed by Elgammal [15]. The value h for a given pixel is computed as $h = m/(0.68*\sqrt{2})$, where m is the median absolute deviation over the sample for consecutive values of the pixel (in the tensor-based algorithms the geodesic distances were used). See more details in [15].

The performance comparison of the methods is based primarily on a quantitative evaluation in terms of true positive ratio

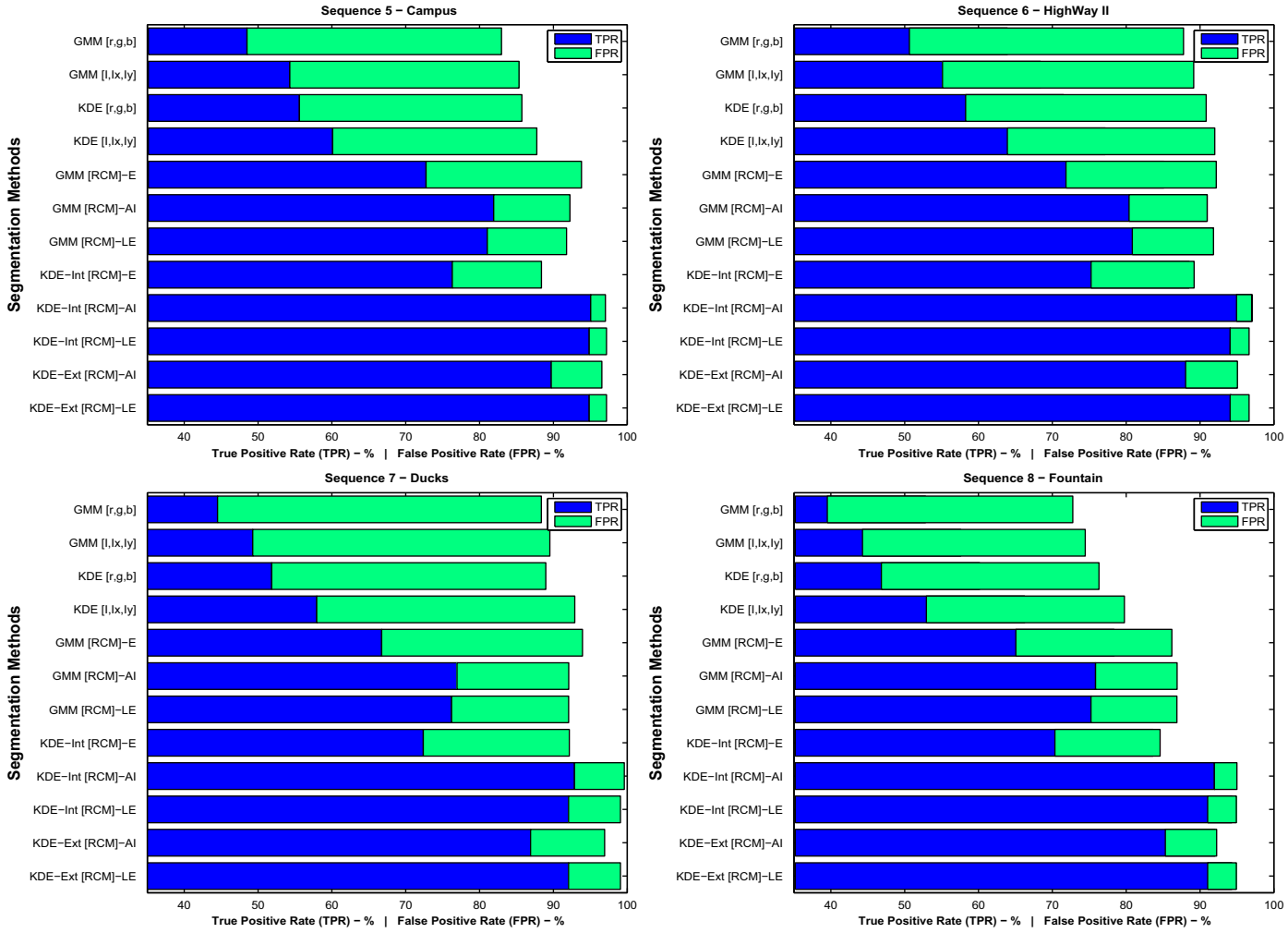

Fig. 5. Experiment 2 ↦ Quantitative performance evaluation on the sequences (5–8), using region covariance matrix (RCM), in terms of: true positive ratio (TPR) and false positive ratio (FPR) ↦ (E, Euclidean; AI, Affine-Invariant; LE, Log-Euclidean; Int, Intrinsic; Ext, Extrinsic). Note that this figure contains the same information that in Table 4.

(TPR) and false positive ratio (FPR)

$$TPR = \frac{TP}{TP+FN} \tag{44}$$

$$FPR = \frac{FP}{FP+TN} \tag{45}$$

where the true positives (TP) are the foreground pixels correctly detected, the false positives (FP) are the background pixels erroneously detected as foreground. (FN) and (TN) correspond to false negatives and true negatives respectively. (FP+TN) corresponds to the ground-truth background and (TP+FN) is the ground-truth foreground. Note that the results presented here are raw data without any postprocessing, e.g. no morphological operators were used in the presentation of the results.

7.1. Experiment 1—structure tensor (ST)

As shown in the Figs. 2–4 and Tables 1 and 2. The vector-based methods in general cannot accurately detect the moving objects, neither in dynamic scenes nor in the case of foreground objects in which the color/intensity information is similar to the background. These methods assume that the scenes are of static structures with limited perturbation. They do not consider the correlation between pixels, meaning that their performance will notably deteriorate when the scenes to be modeled are dynamic natural scenes, which include image noise, camera motion, some illumination variation, and repetitive motions like swaying vegetation, waving trees, rippling water, etc. They label large numbers of moving background pixels as foreground when compared to the tensor-based counterpart (FPR) and also output a huge amount of false negatives on the inner areas of the moving object (TPR). The values of the column AVERAGE displayed in Table 1 clearly demonstrate this fact when comparing the vectorial approaches (GMM[**I,Ix,Iy**] , KDE[**I,Ix,Iy**]) vs the tensorial counterparts (GMM[ST]-AI , KDE-Int[ST]-AI).

In those scenes, although some pixels significantly changes over time, they should be considered as background. In all the experiments, the tensor-based methods outperform largely the vectorial approaches and achieve accurate detection in the sense that they handle some variations of the dynamic background, considering also the correlation between pixels. They use features that effectively model the spatial correlations of neighbors pixels, which is very important to accurately label those moving background pixels. See the tensor-based benefits on Table 2 ↦ Lines: GMM[**I**,**Ix**,**Iy**], KDE[**I**,**Ix**,**Iy**].

The vector-based GMM methods performs poorly at the beginning of the sequences that do not include foreground objects and detect as foreground a lot of background pixels. This behavior is justified by the fact that these methods only use simple

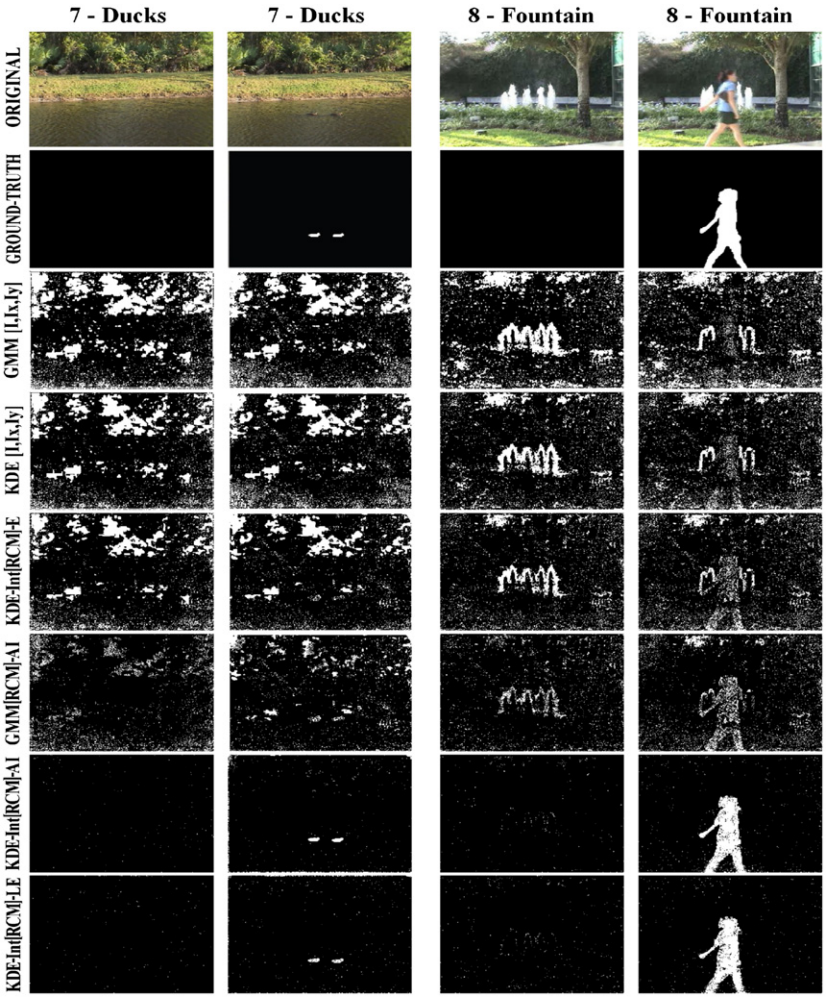

Fig. 6. Experiment 2 ↦ Examples of segmentation results on the sequences (7 and 8), using the RCM as feature.

Table 3
Experiment 1 ↦ parameter values used in the experiments with the structure tensor (ST) as feature. These parameters were used across all vectorial and tensor-based GMM/KDE algorithms. Note that $\overline{K}$ is the number of distributions effectively used in average for each pixel (GMM).

	1-HighWayI	2-Railway	3-HighWayIII	4-HalwayI	5-Campus	6-HighWayII
GMM parameters						
Maximum number of distributions allowed (M)	25	25	25	25	25	25
Number of distributions effectively used ($\overline{K}$)	4	3	3	3	4	4
Learning rate (α)	0.02	0.01	0.01	0.01	0.15	0.25
Threshold (T_g)	0.750	0.700	0.600	0.675	0.750	0.850
KDE parameters						
Type of model	Long term	Short term	Long term	Long term	Long term	Long term
Number of samples (N)	50	50	50	50	50	50
Number of frames sampling window (W)	250	50	250	100	100	200
Threshold (T_k)	20e−5	5e−5	20e−5	10e−5	5e−5	20e−5

features, and so, need to take longer time to train the background models than the tensor-based methods. On the other hand, the tensor frameworks handle dynamic motions immediately and achieve higher accuracy detection at the beginning of the sequences. The spatial correlations provide a substantial evidence for labeling the center pixel and they are exploited to sustain high levels of detection accuracy.

The Riemannian framework was proposed to derive the proper tools to work within the tensor while taking into account its special properties. At this point, our claim is that the special properties of the tensor space should be more naturally handled by working with Riemannian metrics in both parametric (GMM) and non-parametric (KDE) tensor-based frameworks. It must consequently yield more adequate tools to deal with tensors than the Euclidean counterpart, e.g. the Euclidean metric by seeing the tensor space as a linear space is completely blind to its curvature, which implies an inability to exploit all the discriminative information presented in the tensor components. All the presented experiments contribute to clearly validate our claim. In both GMM and KDE tensor-based frameworks, it is visible a dramatic improvement on the segmentation quality, especially in the inner areas of the moving object. This improvement is notable when moving from the conventional Euclidean metric to the Log-Euclidean metric and it is even stronger when using the

Table 4
Experiment 2 ↦ Quantitative performance evaluation on the sequences (5–8), using the region covariance matrix (RCM) as feature, in terms of: true positive ratio (TPR) and false positive ratio (FPR) ↦ (E, Euclidean; AI, Affine-Invariant; LE, Log-Euclidean; Int, Intrinsic; Ext, Extrinsic).

Methods	**5-Campus**		**6-HighWayII**		**7-Ducks**		**8-Fountain**		**Average**	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
GMM [**r**, **g**, **b**]	48.50	34.48	50.63	37.14	44.52	43.85	39.52	33.25	45.79	37.18
GMM [**I**, **Ix**, **Iy**]	54.30	31.05	55.12	34.03	49.28	40.24	44.28	30.18	50.75	33.88
KDE [**r**, **g**, **b**]	55.60	30.13	58.25	32.58	51.85	37.12	46.85	29.47	53.14	32.33
KDE [**I**, **Ix**, **Iy**]	60.10	27.65	63.90	28.10	57.94	34.95	52.94	26.82	58.72	29.38
GMM [RCM]-E	72.75	21.05	71.85	20.35	66.75	27.19	65.05	21.14	69.10	22.43
GMM [RCM]-AI	81.92	10.32	80.39	10.58	76.95	15.13	75.85	11.05	78.78	11.77
GMM [RCM]-LE	81.05	10.73	80.83	10.99	76.21	15.85	75.23	11.64	78.33	12.30
KDE-Int [RCM]-E	76.30	12.08	75.25	13.95	72.38	19.78	70.34	14.26	73.57	15.02
KDE-Int [RCM]-AI	**95.05**	**02.00**	**94.95**	**02.05**	**92.86**	**06.75**	**91.95**	**03.05**	**93.70**	**03.46**
KDE-Int [RCM]-LE	**94.85**	**02.35**	**94.07**	**02.58**	**92.05**	**07.03**	**91.03**	**03.89**	**93.00**	**03.96**
KDE-Ext [RCM]-AI	89.71	06.85	88.05	07.01	86.91	10.03	85.30	06.95	87.49	07.71
KDE-Ext [RCM]-LE	94.85	02.35	94.07	02.58	92.05	07.03	91.03	03.89	93.00	03.96

Table 5
Experiment 2 ↦ Comparative evaluation between the structure tensor (ST) and region covariance matrix (RCM), on the sequence 5 (Campus). The differential values (Δ) were calculated using the information presented in the column Campus of the Tables 1 and 4 as (Δ = Methods A−Methods B). Notice that (ΔTPR → +) = Good and (ΔFPR → −) = Good.

Δ = A−B

Methods B=ST	**Methods A=RCM**							
	GMM[RCM]-AI		**GMM[RCM]-LE**		**KDE-Int[RCM]-AI**		**KDE-Int[RCM]-LE**	
	ΔTPR	ΔFPR	ΔTPR	ΔFPR	ΔTPR	ΔFPR	ΔTPR	ΔFPR
GMM[ST]-AI	+06.98	−03.76	–	–	–	–	–	–
GMM[ST]-LE	–	–	+08.23	-04.20	–	–	–	–
KDE-Int[ST]-AI	–	–	–	–	+07.15	−03.52	–	–
KDE-Int[ST]-LE	–	–	–	–	–	–	+ 07.71	−03.36

Table 6
Experiment 2 ↦ Comparative evaluation between the structure tensor (ST) and region covariance matrix (RCM), on the sequence 6 (HighWayII). The differential values (Δ) were calculated using the information presented in the column HighWayII of the Tables 1 and 4 as (Δ = Methods A−Methods B). Notice that (ΔTPR → +) = Good and (ΔFPR → −) = Good.

Δ = A−B

Methods B=ST	**Methods A=RCM**							
	GMM[RCM]-AI		**GMM[RCM]-LE**		**KDE-Int[RCM]-AI**		**KDE-Int[RCM]-LE**	
	ΔTPR	ΔFPR	ΔTPR	ΔFPR	ΔTPR	ΔFPR	ΔTPR	ΔFPR
GMM[ST]-AI	+06.56	−04.62	–	–	–	–	–	–
GMM[ST]-LE	–	–	+07.81	−04.87	–	–	–	–
KDE-Int[ST]-AI	–	–	–	–	+07.70	−05.90	–	–
KDE-Int[ST]-LE	–	–	–	–	–	–	+07.10	−04.55

Table 7
Experiment 2 ↦ Parameters values used in the experiments with the region covariance matrix (RCM) as feature. These parameters were used across all vectorial and tensor-based GMM/KDE algorithms. Note that $\overline{K}$ is the number of distributions effectively used in average for each pixel (GMM).

	5-Campus	**6-HighWayII**	**7-Ducks**	**8-Fountain**
GMM parameters				
Maximum number of distributions allowed (M)	25	25	25	25
Number of distributions effectively used ($\overline{K}$)	4	4	6	5
Learning rate (α)	0.15	0.25	0.05	0.05
Threshold (T_g)	0.750	0.850	0.950	0.925
KDE parameters				
Type of model	Long term	Long term	Short term	Short term
Number of samples (N)	50	50	50	50
Number of frames sampling window (W)	100	200	50	50
Threshold (T_k)	5e−5	20e−5	15e−5	15e−5

Affine-Invariant metric. The results proved that the option by a Riemannian metric has a deep impact on the tensor statistics and, hence, on the segmentation results. See for example, the benefits of the Affine-Invariant metric over the Euclidean metric in Table 2 $\mapsto$ Lines: GMM[ST]−**E**, KDE-Int[ST]−**E**.

To be viable, a foreground detection algorithm need to work properly in complex environments where for instance the background may be multi-modal and where there is a significant and constant activity in the scene. Although the multi-modal parametric paradigm on tensor domain provided by the GMM perform relatively well on the analyzed environments, its performance, however, depends on appropriately setting a number of parameters, like the a priori probability of observing the background. The robustness of that approach highly depends on the nature of the observed scene. The GMM paradigm is based on the assumption that each pixel views background states more often than foreground ones, therefore the states with higher prior probabilities (weights) and higher concentrations would be considered as background. Based on this premise, the first states with higher weights and concentrations, whose combined priori probabilities are greater than a pre-defined weight threshold are considered as the representative models of background. If the background is multi-modal and the scene activity is high, it is sometime impossible to find a threshold allowing all the states representing the background to be labeled accordingly while preventing some foreground states to be labeled as background as well. All the results presented demonstrate that our non-parametric reformulation of the tensor-based GMM proposed in [2] improved considerably the segmentation performance. See for example, the KDE-based benefits over the GMM in Table 2 $\mapsto$ Columns: KDE − Int[ST]−AI, KDE − Int[ST]-LE$\mapsto$ Lines: GMM[ST]−AI, GMM[ST]−LE.

As shown by Elgammal [15,16] the KDE have been successful, to model, on Euclidean sample spaces, the nonparametric nature of complex backgrounds. The results clearly prove, in the tensor domain, the advantages of the non-parametric paradigm over the parametric counterpart proposed in [2] in a similar way to what Elgammal [15,16] did in the vectorial domain. This method can deal with multi-modality in background tensor distributions without specifying the number of modes. It is shown that the tensor-based KDE algorithms endowed with the Riemannian metrics obtains a much cleaner segmented background (less false positives) than GMM counterparts and the foreground segmented is cleaner (less false negatives), better connected for each object, almost noiseless, and furthermore the contours of the foreground objects are well delineated. Again, the values of the column **AVERAGE** displayed in Table 1 clearly demonstrate the benefits of the tensorial nonparametric formulations (KDE-Int[ST]-AI, KDE-Int[ST]-LE) vs the tensorial parametric counterparts (GMM [ST]-AI, GMM[ST]-LE).

Since the dynamic motions do not repeat exactly, it causes some performance degradation on the GMM tensor-based, which detect several background pixels as foreground and also labeled a considerably number of foreground pixels as background on the inner areas of the moving objects. The proposed KDE tensor-based method outperforms the GMM tensor-based, and achieves very high accuracy in the detection of the moving objects. The foreground regions are accurately segmented using the tensor-based KDE even though their sizes are small. Some of these regions are mistakenly identified by the tensor-based GMM. Although the proposed method also misses some pixels, the overall performance of our method is globally better.

As the work presented by Caseiro et al. [2] showed for the tensor-based GMM paradigm, new points of view on the tensor space can lead to significantly simpler computations and therefore faster foreground detection algorithms. In the experiments presented herein, we also concluded that the tensor-based algorithms endowed with the Log-Euclidean metric has the same excellent theoretical properties as the Affine-Invariant metric. In the case of the tensor-based GMM paradigm, from a practical point of view, the segmentation results are similar but are obtained much faster, with an average computation time ratio of at least 2 in favor of the Log-Euclidean framework. In the case of the tensor-based KDE paradigm, the conclusions regarding the segmentation results are similar to those of the GMM counterpart, i.e. from a practical point of view the segmentation results between the two Riemannian metrics are also very similar. The values of the Table 2 (Columns: GMM-[ST]-LE, KDE-Int[ST]-LE $\mapsto$ Lines: GMM-[ST]-AI, KDE-Int[ST]-AI) clearly highlight the residual difference in the performance between the two Riemannian metrics. Regarding the computational cost, the tensor-based KDE[ST]-AI is obviously not a competitive method. The time consuming is highly dependent of the number of samples used, mainly because of two reasons: Firstly, although at each time t we only need to compute once the Ricci curvature tensor $\mathcal{R}$, it is necessary to calculated N times the normal coordinates of the point $\mathbf{Z}$ in the normal coordinate system centred at $\mathbf{Z}_i$; Secondly (at each time t), it is necessary to compute N times the geodesic distance between the tensors $\mathbf{Z}$ and $\mathbf{Z}_i$. The most striking difference between the two Riemannian metrics in the KDE case resides in their computational cost, due to the space with a null curvature induced by the Log-Euclidean metric. The KDE[ST]-LE is much more simple since we do not need to compute the volume density function and at each time t we only need to compute the matrix operation $\log(\mathbf{Z})$. In fact, if we compare the tensor-based GMM[ST]-LE algorithm [2] with the KDE[ST]-LE proposed herein, we conclude that the KDE version is more simple. Despite of, in the KDE[ST]-LE be necessary to compute N times the nonnegative function $K(\cdot)$, the GMM[ST]-LE in practice is more complex because it involves the computation of several log and exp matrix operations, which are more time consuming than the function $K(\cdot)$ used in this work. We conclude that, although the tensor-based KDE[ST]-AI is slower than both the GMM[ST]-AI|LE versions, the tensor-based KDE[ST]-LE is faster than all the others tensor-based approaches. In fact, the KDE[ST]-LE is in average approximately three times faster than the GMM[ST]-LE method, and the KDE achieves much better segmentation results in all the sequences tested.

In all the sequences tested, the results proved that the extrinsic reformulation of the KDE-Ext[ST]-AI preserve some of the nonparametric benefits, while speedup the process. See the extrinsic KDE-based benefits over the GMM-based counterpart in Table 2 $\mapsto$ Column: KDE-Ext[ST]-AI $\mapsto$ Line: GMM-[ST]-AI. Although the segmentation results achieved by the extrinsic KDE are slightly worse than the intrinsic counterpart, the advantages of the nonparametric paradigm over the parametric version (GMM) remains. The values of the Table 2 (Column: KDE-Ext[ST]-AI $\mapsto$ Line: KDE-Int[ST]-AI) highlight the loss in the performance of the extrinsic tensor-based KDE over the intrinsic tensor-based KDE. Due to the less complexity, this extrinsic KDE algorithm is in average approximately four times faster than the intrinsic version.

7.2. Experiments 2—region covariance matrix (RCM)

From the results of the experiments presented in Section 7.1, we conclude that the sequences 5 and 6 are particularly difficult scenarios from the background modeling point of view. It is evident that the motion of the camera and the significative image noise cause some degradation in the performance of the tensor-based algorithms using the structure tensor (ST) as feature, when compared with the rates achieved in the sequences 1–4. The

region covariance matrix (RCM) has some special properties that can help in more difficult scene conditions. The noise corrupting individual samples are largely filtered out with the average filter during the covariance computation. The covariance is invariant to the mean changes such as identical shifting of color values, which is very valuable when scenes are under some varying illumination conditions, i.e. due to the zero-mean normalization by subtraction of the sample mean the descriptor achieves some invariance in the case of photometric and illumination changes.

In order to evaluate the potential benefits of the covariance matrices as feature, in this second part of the experiments we tested again the sequences 5 and 6. We also used the sequences 7 and 8 to evaluate the foreground segmentation performance of the proposed tensor-framework using the RCM feature. In fact, the sequences 7 and 8 are the most challenging sequences presented in this work. In the sequence 7, the upper part of the scene contains heavily swaying trees with nonperiodic motions. The challenges in the lower part of the scene are that the background is composed of subtle illumination variations along with ripples in the water and the color of the ducks and background is similar. In the sequence 8, the nonperiodic motions of the spouting fountain and the swaying tree branches above constitute the main challenges.

At this point, our claim is that the special properties of the RCM are more suitable to deal with the changeling sequences 5 and 6. In all the tensor-based experiments (GMM and KDE endowed with all the metrics) and in both the sequences (5 and 6), the RCM outperforms the structure tensor as feature (see Tables 5 and 6). It is notable that in the sequence 5 (Campus) the RCM deals better with the image noise problem by filtering out the samples during the covariance computation. See the RCM benefits when applied to the sequence 5 in the Table 5. The sequence 6 (HighWayII) is even more changeling because it present image noise and some camera motion. In this case, it is also clearly visible the improvement on the segmentation quality provided by the use of the RCM feature, especially in the false positive rate. The RCM benefits when applied to the sequence 6 is clearly visible in Table 6.

Regarding the sequences 7 and 8, all the conclusions obtained in the Section 7.1 for the sequences 1–6 using the structure tensor (ST), remains when the region covariance matrix (RCM) is used as feature, i.e. all the five goals described previously are also confirmed using the RCM as feature (see Figs. 5 and 6 and Table 4).

8. Conclusions

Kernel density estimators (KDEs) have been successful to model, on Euclidean sample spaces, the nonparametric nature of complex and time varying physical processes. Taking into account the Riemannian structure of the tensor manifold, we derived a novel nonparametric Riemannian framework on the tensor field, with application to foreground segmentation. The tensor was used to convert the image into a more information rich form (tensor field), to yield latent discriminating features. We presented the necessary background about differential geometry, i.e. we focus on the main geometric concepts of Riemannian manifolds, nonparametric estimation on such manifolds and the respective extensions to the tensor manifold, endowed with two Riemannian metrics (Affine-Invariant and Log-Euclidean). The explicit formulation of a KDE on the tensor manifold endowed with the two Riemannian metrics, respecting the non-Euclidean nature of the space, as well as, the nonparametrically reformulation of the tensor-based algorithms previously proposed to foreground segmentation are the core contributions of the paper.

In overall, the paper shows that the consequent usage of the underlying Riemannian structure of the tensor manifold for model derivation, in conjunction with a suitable nonparametric estimation scheme for the underlying density, yields the most accurate and reliable approach to foreground detection from tensor-valued images presented so far (i.e. yields the most accurate technique to estimate the underlying density of the tensor data).

A careful comparison of the Log-Euclidean and Affine-Invariant metrics on the KDE algorithms described in this paper, showed that there are very few differences on the results on foreground segmentation from real video sequences, but the Log-Euclidean proved to be considerably faster. In fact, the most striking difference between the several KDE versions proposed resides in theirs computational costs. Thus, for this type of application and for these sequences, the KDE using the Log-Euclidean metric seems to be perfectly suited. In what regards, the best tensor-descriptor (structure tensor vs region covariance), the results confirmed that the RCM is the best choice to do foreground detection. Although it is a little more time consuming due to the zero-mean normalization, its special properties proved to be important in sequences with noise and scenes under some varying illumination conditions. Moreover, by inducing a space with a null curvature, the Log-Euclidean metric considerably simplifies the tensor-based KDE, which results in an algorithm that is in fact more accurate and faster than all the tensor-based GMM versions.

We demonstrated in this paper that there are indeed several generalizations of the kernel density estimator to the tensor manifold. This is important, since situations in image segmentation, texture classification, object detection, tracking as well as in others branches of science, such as applied mathematics, physics, mechanics, medical imaging, etc., where tensors need to be processed, are highly varied. As a consequence, the relevance of each generalization of the KDE and of the associated metric framework may depend on the application considered.

Acknowledgments

The work of Rui Caseiro, Pedro Martins, and João F. Henriques was supported by the Fundação para a Ciência e Tecnologia through the PhD grants SFRH/BD74152/2010, SFRH/BD45178/2008 and SFRH/BD75459/2010, respectively. This work was also supported by the projects Brisa, Auto-Estradas de Portugal and PTDC/EEA-CRO/122812/2010 (Differential Geometry for Computer Vision and Pattern Recognition—DG2CVPR). The authors thank also the reviewers for their valuable suggestions and comments.

References

[1] R. Caseiro, J.F. Henriques, P. Martins, J. Batista, A nonparametric Riemannian framework on tensor field with application to foreground segmentation, in: IEEE International Conference on Computer Vision (ICCV), 2011.
[2] R. Caseiro, P. Martins, J. Batista, Background modeling on tensor field for foreground segmentation, in: British Machine Vision Conference (BMVC), 2010, pp. 96.1–96.12.
[3] R.L. Garcia, M. Rousson, R. Deriche, C. Alberola-Lopez, Tensor processing for texture and colour segmentation, in: Scandinavian Conference on Image Analysis, 2005, pp. 1117–1127.
[4] J. Bigun, G. Granlund, J. Wiklund, Multidimensional orientation estimation with applications to texture analysis and optical flow, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (8) (1991) 775–790.
[5] J. Malcolm, A.Tannenbaum, A graph cut approach to image segmentation in tensor space, in: IEEE Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–8.
[6] S. Han, W. Tao, X. Wu, Texture segmentation using independent-scale component-wise Riemannian-covariance Gaussian mixture model in *kl*

measure based multi-scale nonlinear structure tensor space, Pattern Recognition 44 (3) (2011) 503–518.
[7] C. Lenglet, M. Rousson, R. Deriche, DTI segmentation by statistical surface evolution, IEEE Transactions on Medical Imaging 25 (6) (2006) 685–700.
[8] P. Fillard, X. Pennec, V. Arsigny, N. Ayache, Clinical DT-MRI estimation, smoothing, and fiber tracking with log-Euclidean metrics, IEEE Transactions on Medical Imaging 26 (11) (2007) 1472–1482.
[9] X. Pennec, P. Fillard, N. Ayache, A Riemannian framework tensor computing, International Journal of Computer Vision 66 (1) (2006) 41–66.
[10] B. O'Neill, Semi-Riemannian Manifolds: With Applications to Relativity, Academic Press, 1983.
[11] R. Caseiro, J.F. Henriques, J. Batista, Foreground segmentation via background modeling on Riemannian manifolds, in: IEEE International Conference on Pattern Recognition (ICPR), 2010.
[12] L. Skovgaard, Riemannian geometry of the multivariate normal model, Scandinavian Journal of Statistics 11 (1984) 211–233.
[13] C. Lenglet, M. Rousson, R. Deriche, O. Faugeras, Statistics on manifold of multivariate normal distributions: theory application diffusion tensor MRI processing, Journal of Mathematical Imaging and Vision 25 (3) (2006) 423–444.
[14] V. Arsigny, P. Fillard, X. Pennec, N. Ayache, Geometric means in a novel vector space structure on symmetric positive-definite matrices, SIAM Journal on Matrix Analysis and Applications 29 (1) (2007) 328–347.
[15] A. Elgammal, D. Harwood, L.S. Davis, Non-parametric model for background subtraction, in: European Conference on Computer Vision (ECCV), 2000, pp. 751–767.
[16] A. Elgammal, R. Duraiswami, D. Harwood, L.S. Davis, R. Duraiswami, D. Harwood, Background and foreground modeling using nonparametric kernel density for visual surveillance, Proceedings of the IEEE 90 (7) (2002) 1151–1163.
[17] B. Pelletier, Kernel density estimation on Riemannian manifolds, Statistics & Probability Letters 73 (3) (2005) 297–304.
[18] C. Stauffer, W. Grimson, Adaptive background mixture models for real-time tracking, in: IEEE Computer Vision and Pattern Recognition (CVPR), 1999, pp. 246–252.
[19] F.P.O. Tuzel, P. Meer, Region covariance: a fast descriptor for detection and classification, in: European Conference on Computer Vision (ECCV), 2006, pp. 589–600.
[20] S. Elhabian, K. El-Sayed, S. Ahmed, Moving object detection in spatial domain using background removal techniques-state-of-art, Recent Patents on Computer Science 1 (1) (2008) 32–54.
[21] T. Bouwmans, F.E. Baf, B. Vachon, Statistical background modeling for foreground detection: a survey, in: Handbook of Pattern Recognition and Computer Vision, vol. 4(3). World Scientific Publishing, 2010, pp. 181–199.
[22] A. Mittal, A. Monnet, N. Paragios, Scene modeling and change detection in dynamic scenes: a subspace approach, Computer Vision and Image Understanding 113 (1) (2009) 63–79.
[23] T. Bouwmans, F.E. Baf, B. Vachon, Background modeling using mixture of Gaussians for foreground detection—a survey, Recent Patents on Computer Science 1 (3) (2008) 219–237.
[24] T. Bouwmans, Subspace learning background modeling: survey, Recent Patents on Computer Science 2 (3) (2009) 223–234.
[25] Y.-T. Chen, C.-S. Chen, C.-R. Huang, Y.-P. Hung, Efficient hierarchical method for background subtraction, Pattern Recognition 40 (10) (2007) 2706–2715.
[26] C.R. Wren, A. Azarbayejani, T. Darrell, A.P. Pentland, Pfinder: real-time tracking of the human body, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 780–785.
[27] N. Friedman, Russell, Image segmentation in video sequences: a probabilistic approach, in: Thirteenth Conference on Uncertainty in Artificial Intelligence, 1997, pp. 175–181.
[28] O. Tuzel, F. Porikli, P. Meer, A Bayesian approach to background modeling, in: IEEE Computer Vision and Pattern Recognition (CVPR), 2005.
[29] S. Zhang, S. Liu, Background subtraction on distributions, in: European Conference Computer Vision, 2008, pp. 276–289.
[30] K.A. Patwardhan, G. Sapiro, V. Morellas, Robust foreground detection in video using pixel layers, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (4) (2008) 746–751.
[31] N. Oliver, B. Rosario, A.P. Pentland, A bayesian computer vision system for modeling human interactions, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 831–843.
[32] A. Monnet, A. Mittal, N. Paragios, V. Ramesh, Background modeling and subtraction of dynamic scenes, in: IEEE International Conference on Computer Vision (ICCV), 2003, pp. 1305–1312.
[33] M. Seki, T. Wada, H. Fujiwara, K. Sumi, Background subtraction based on cooccurrence of image variations, in: IEEE Computer Vision and Pattern Recognition (CVPR), 2003, pp. 65–72.
[34] Z. Li, P. Jiang, H. Ma, J. Yang, D. Tang, A model for dynamic object segmentation with kernel density estimation based on gradient features, Image and Vision Computing 27 (6) (2009) 817–823.
[35] S. Jabri, Z. Duric, H. Wechsler, A. Rosenfeld, Detection and location of people in video images using adaptive fusion of color and edge information, in: IEEE Computer Vision and Pattern Recognition (CVPR), 2000, pp. 627–630.
[36] O. Javed, K. Shafique, M. Shah, A hierarchical approach to robust background subtraction using color and gradient information, in: IEEE Workshop on Motion and Video Computing, 2002, pp. 22–27.
[37] R. Pless, Spatio-temporal background models for surveillance, Journal on Applied Signal Processing 14 (2005) 2281–2291.
[38] Y. Sheikh, M. Shah, Bayesian modeling dynamic scenes for object detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (11) (2005) 1778–1792.
[39] S. Babacan, T. Pappas, Spatiotemporal algorithm for joint video segmentation and foreground detection, in: European Signal Processing Conference, 2006.
[40] V. Mahadevan, N. Vasconcelos, Spatiotemporal saliency in dynamic scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (1) (2010) 171–177.
[41] V. Mahadevan, N. Vasconcelos, Background subtraction in highly dynamic scenes, in: IEEE Computer Vision and Pattern Recognition (CVPR), 2008, pp. 1–6.
[42] Antoni B. Chan, Vijay Mahadevan, Nuno Vasconcelos, Generalized Stauffer Grimson background dynamic scenes, Machine Vision and Applications 22 (5) (2011) 751–766.
[43] M. Heikkila, M. Pietikainen, A texture-based method for modeling the background and detecting moving objects, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (4) (2006) 657–662.
[44] J. Yao, J. M. Odobez, Multi-layer background subtraction based on color and texture, in: IEEE Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–8.
[45] S. Zhang, H. Yao, S. Liu, Dynamic background modeling and subtraction using spatio-temporal local binary patterns, in: IEEE International Conference on Image Processing (ICIP), 2008, pp. 1556–1559.
[46] S. Zhang, H. Yao, S. Liu, Dynamic background subtraction based on local dependency histogram, in: European Conference on Computer Vision (ECCV), 2008.
[47] Bineng Zhong, Hongxun Yao, Shaohui Liu, Xiaotong Yuan, Local histogram of figure/ground segmentations for dynamic background subtraction, EURASIP Journal on Advances in Signal Processing (2010) 782101, 〈http://dx.doi.org/10.1155/2010/782101〉.
[48] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikainen, S.Z. Li, Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes, in: IEEE Computer Vision Pattern Recognition (CVPR), 2010, pp. 1301–1306.
[49] A. Bugeau, P. Perez, Detection and segmentation of moving objects in highly dynamic scenes, in: IEEE Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–8.
[50] A. Bugeau, P. Perez, Detection and segmentation of moving objects in complex scenes, Computer Vision and Image Understanding 113 (4) (2009) 459–476.
[51] L. Cheng, M. Gong, Realtime background subtraction from dynamic scenes, in: IEEE International Conference on Computer Vision (ICCV), 2009, pp. 2066–2073.
[52] J. Pilet, C. Strecha, P. Fua, Making background subtraction robust to sudden illumination changes, in: European Conference on Computer Vision (ECCV), 2008, pp. 567–580.
[53] Y. Sheikh, O. Javed, T. Kanade, Background subtraction for freely moving cameras, in: IEEE International Conference on Computer Vision (ICCV), 2009, pp. 1219–1225.
[54] M. Gong, L. Cheng, Foreground segmentation of live videos using locally competing 1SVMs, in: IEEE Computer Vision and Pattern Recognition (CVPR), 2011.
[55] Suyash P. Awate, Hui Zhang, James C. Gee, Fuzzy nonparametric DTI segmentation for robust cingulum-tract extraction, Medical Image Computing and Computer Assisted Intervention (MICCAI), in: Proceedings of Fuzzy Non Parametric DTI Cingulum Tract Extraction_Awate 07, vol. 4791, 2007, pp. 294–301.
[56] F.P.O. Tuzel, P. Meer, Pedestrian detection via classification on Riemannian manifolds, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (10) (2008) 1713–1727.
[57] O.T.F. Porikli, P. Meer, Covariance tracking using model update based on lie algebra, in: IEEE Computer Vision and Pattern Recognition (CVPR), 2006, pp. 728–735.
[58] C. Harris, M. Stephens, A combined corner and edge detector, in: Fourth Alvey Vision Conference, 1988, pp. 147–151.
[59] H.-Y. Wang, K.-K. Ma, Accurate optical flow estimation using adaptive scale-space and 3D structure tensor, in: International Conference on Image Processing (ICIP), 2002, pp. 301–304.
[60] S. Zenzo, A note on the gradient of multi-image, Computer Vision, Graphics, Image Processing 33 (1) (1986) 116–125.
[61] J. Bigun, Recognition of local symmetries in gray value images by harmonic functions, in: IEEE International Conference on Pattern Recognition (ICPR), 1988, pp. 345–347.
[62] J.M. Lee, Introduction to Smooth Manifolds, Springer, 2003.
[63] W. Boothby, An Introduction to Differentiable Manifolds and Riemannian Geometry, Academic Press, 1987.
[64] A. Robles-Kelly, E.R. Hancock, A Riemannian approach to graph embedding, Pattern Recognition 40 (3) (2007) 1042–1056.
[65] F. Zhang, E.R. Hancock, New Riemannian techniques for directional and tensorial image data, Pattern Recognition 43 (4) (2010) 1590–1606.
[66] T. Levi-Civita, G. Ricci, Méthodes de calcul différentiel absolu et leurs applications, Mathematische Annalen B 54 (1900) 125–201.
[67] T. Levi-Civita, Nozione di parallelismo in varieta qualunque, Rendiconti del Circolo Matematico di Palermo 42 (1917) 173–205.
[68] C.R. Rao, Information and the accuracy attainable in the estimation statistical parameters, Bulletin of the Calcutta Mathematical Society 37 (1945) 81–91.

[69] M. Moakher, A differential geometric approach to the geometric mean symmetric of positive-definite matrices, SIAM Journal on Matrix Analysis and Applications 26 (3) (2005) 735–747.
[70] C. Atkinson, A.F. Mitchell, Rao distance measure, Sankhya: The Indian Journal of Statistics 43 (3) (1981) 345–365.
[71] X. Pennec, Statistical Computing on Manifolds for Computational Anatomy, L'Habilitation Diriger des Recherche, 2006.
[72] A. Besse, Manifolds All of Whose Geodesics are Closed, Springer, 1978.
[73] T. Willmore, Riemannian Geometry, Oxford University Press, 1993.
[74] X. Pennec, Intrinsic statistics on Riemannian manifolds: basic tools for geometric measurements, Journal of Mathematical Imaging and Vision 25 (1) (2006) 127–154.
[75] I. Chavel, Riemannian Geometry—A Modern Introduction, Cambridge University Press, 1993.
[76] S. Gallot, D. Hulin, J. Lafontaine, Riemannian Geometry, Springer, 1990.
[77] S. Gallot, D. Hulin, J. Lafontaine, Riemannian Geometry, Springer, 2004.
[78] A. Prati, I. Mikic, M. Trivedi, R. Cucchiara, Detecting moving shadows: algorithms and evaluation, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (7) (2003) 918–923.
[79] N. Brisson, A. Zaccarin, Learning and removing cast shadows, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (7) (2007) 1133–1146.

Rui Caseiro received the B.Sc. degree in electrical engineering (specialization in automation) from the University of Coimbra, Coimbra, Portugal, in 2005. Since 2007, he has been involved in several research projects, which include the European project "Perception on Purpose" and the National project "Brisa-ITraffic". He is currently a Ph.D. student and researcher with the Institute of Systems and Robotics and the Department of Electrical and Computer Engineering, Faculty of Science and Technology, University of Coimbra. His current research interests include the interplay of differential geometry with computer vision and pattern recognition.

Pedro Martins received his M.Sc. degree in Computer and Electrical Engineering from the University of Coimbra, Portugal in 2008. He is a Ph.D. student at Institute of Systems and Robotics, at University of Coimbra, Portugal. His main research include image alignment, tracking and facial expression analysis and synthesis.

João F. Henriques received his M.Sc. degree in Electrical Engineering from the University of Coimbra, Portugal in 2009. He is currently a Ph.D. student at the Institute of Systems and Robotics, University of Coimbra, Portugal. His research interests include combinatorial optimization, kernel methods and machine learning in general, with a special focus on visual surveillance applications.

Prof. Jorge Batista received the M.Sc. and Ph.D. degree in Electrical Engineering from the University of Coimbra in 1992 and 1999, respectively. He joins the Department of Electrical Engineering and Computers, University of Coimbra, Coimbra, Portugal, in 1987 as a research assistant where he is currently an Associate Professor. He is a founding member of the Institute of Systems and Robotics (ISR) in Coimbra, where he is a Senior Researcher. His research interest focus on a wide range of computer vision and pattern analysis related issues, including real-time vision, video surveillance, video analysis, non-rigid modeling and facial analysis. He is an IEEE member.