Kernelized Covariance for Action Recognition

Jacopo Cavazza*†, Andrea Zunino*†, Marco San Biagio* and Vittorio Murino*‡
* Pattern Analysis & Computer Vision – Istituto Italiano di Tecnologia, Via Morego 30, 16163, Genova, Italy
† Università degli Studi di Genova – Dipartimento di Ingegneria Navale, Elettrica, Elettronica e delle Telecomunicazioni,
Via All'Opera Pia, 11A, 16145, Genova, Italy
‡ Università di Verona – Dipartimento di Informatica, Strada le Grazie 15, 37134, Verona, Italy
{jacopo.cavazza,andrea.zunino,marco.sanbiagio,vittorio.murino}@iit.it

Abstract—**In this paper we aim at increasing the descriptive power of the covariance matrix, limited in capturing linear mutual dependencies between variables only. We present a rigorous and principled mathematical pipeline to recover the kernel trick for computing the covariance matrix, enhancing it to model more complex, non-linear relationships conveyed by the raw data. To this end, we propose *Kernelized-COV*, which generalizes the original covariance representation without compromising the efficiency of the computation. In the experiments, we validate the proposed framework against many previous approaches in the literature, scoring on par or superior with respect to the state of the art on benchmark datasets for 3D action recognition.**

Publicly available code: https://www.iit.it/pavis/code/kcar

I. Introduction

In the past three decades, motion capture systems – MoCap – have been engineered with the ultimate goal of tracking and recording human motion while guaranteeing high resolutions in both spatial and temporal domains. The acquired data consist of time series of joint/marker 3D positions and are broadly used for several different applications, *e.g.*, studying human motions in sport sciences, inferring biometric patterns for person identification or generating realistic motion sequences in computer animation to name a few [1]. Among these ones, action and activity recognition displays a crucial role in human-robot interaction, autonomous driving vehicles and video-surveillance [2]. However, devising effective methods to analyze MoCap data is demanding due to the many yet unsolved problems related, for instance, to missing acquisitions of joints coordinates or to highly corrupted data.

Previous attempts to face these issues either rely on some distance learning techniques (*e.g.*, subspace view invariant metric [3]) or applied stochastic techniques to model the degree of uncertainty in the data. For instance, a hidden Markov model is used in [4] to produce weak classifiers which are enhanced by AdaBoost. Furthermore, [5] proposed an action graph to model the dynamics for action recognition and exploited a bag of 3D points as feature representation.

Since the spatial and/or temporal dimensions of the recorded data can be heavy, dimensionality reduction [6] or feature selection [7] methods have been devised. However, in general, the classification is subsequent to a design phase of discriminative features such as actionlets [8], random occupancy patterns [9], pose-based sets [10], space-time trajectories [11], velocity and acceleration [12], normal vectors [13] or Lie group geometry embedding [14].

As a different paradigm to a customized class of task-specific features, generalizable representations driven by covariance matrix were shown to be promising, either encoding spatio-temporal derivatives of joint positions [15] or producing a hierarchical temporal pyramids of descriptors [16].

Recently, the new state of the art for action and activity recognition from MoCap data was set by [17], where several Gram matrices are computed to produce multiple representations of the joint positions of each trial and, once a fusion step is performed, a log-Euclidean kernel feeds the SVM classifier. Therein, the covariance is replaced by kernel matrices and this is motivated by the observation that the former can only understand linear relationships while the latter allows to model general ones. In this work, we pursue an opposite perspective, focusing on the covariance representation and rigorously devising a *kernelized* version to extend its discriminative power.

Indeed, by the direct usage of a kernel, we can avoid any preliminary explicit feature encoding (as, for instance, occurs in [15]) and, for a general class of kernel functions, we recover the kernel trick for covariance matrix estimation. As a result, its descriptiveness increases from linear to arbitrary relationships modelling, while the efficiency in the computation is preserved.

To the best of our knowledge, this problem was never faced before in this principled way in both machine learning and pattern recognition fields.

To sum up, we highlight the contributions of this paper.

- We propose a new kernelized representation for covariance matrix, namely ***Kernelized-COV***. By recovering the well-known kernel trick, we can capture more general inter-dependencies between variables in a way that the usual covariance descriptor becomes a particular case and the overall computational cost does not increase.
- In order to prove the effectiveness of our approach for action and activity recognition of MoCap data, we compare our method against different ones on MSR-Action3D [18], MSR-Daily-Activity [19], MSRC-Kinect12 [20] and HDM-05 [21] benchmark datasets. With respect to the state-of-the-art methods [17], the registered performance shows comparable results in the first two datasets and better scores in the remaining ones. This properly certifies that our kernelization is able to bridge the gap between covariance and kernel-based representation.

The rest of the paper is outlined as follows. In Section II, we sketch some theoretical background about the covariance

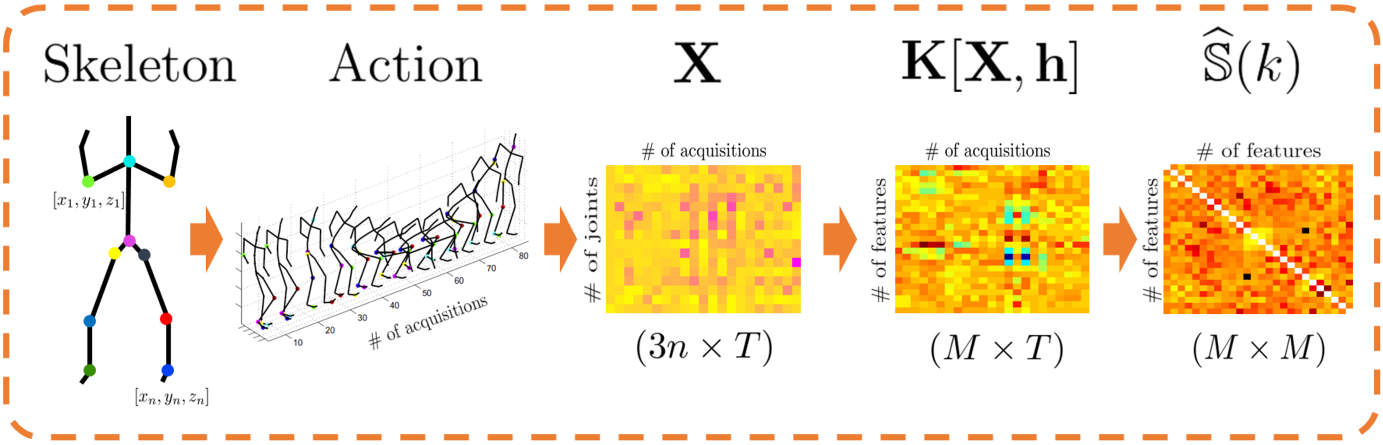

Fig. 1. Overview of the proposed framework. From the human skeleton, for each action, we extract MoCap data. The latter are represented through the matrix $\mathbf{X}$ which collects the three-dimensional coordinates, referring to the n joints, acquired during T successive instants. A kernel encoding is performed by means of the Gram matrix $\mathbf{K}[\mathbf{X}, \mathbf{h}]$, which is finally used to compute the kernelized covariance $\widehat{\mathbb{S}}(k)$.

matrix. In Section III, we present our framework which is experimentally validated in Section IV. Finally, Section V draws some conclusions and profiles future work.

II. Background

At an arbitrary timestamp t, a generic MoCap system represents the body of a human agent as the collection $\mathbf{x}(t) \in \mathbb{R}^{3n}$ of the three-dimensional locations $\mathbf{x}_1(t), \dots, \mathbf{x}_n(t)$ of n joints/markers positions, being $\mathbf{x}_i(t) = [x_i(t), y_i(t), z_i(t)]^\top \in \mathbb{R}^3$ the x, y and z coordinates for $i = 1, \dots, n$. In order to quantify how much any pair of the coordinates mutually change in time, the notion of covariance is classically exploited in statistics [22]. However, it cannot be computed in absence of a known distribution for the probability according to which the samples $\mathbf{x}(t)$ are drawn. However, this assumption is seldom verified in real cases and, as an alternative, the sampling covariance matrix $\widehat{\mathbb{S}}$ is usually exploited: this is due to the fact that it is an unbiased estimator of the original covariance[1] and can be computed using a finite number of samples $\mathbf{x}(t)$, $t = 1, \dots, T$, only. Precisely, it is defined as

$$\widehat{\mathbb{S}}(\mathbf{X}) = \frac{1}{T-1} \sum_{t=1}^{T} (\mathbf{x}(t) - \boldsymbol{\mu})(\mathbf{x}(t) - \boldsymbol{\mu})^\top, \tag{1}$$

where $\mathbf{X}$ represents the $3n \times T$ data matrix which stacks by columns all the temporal acquisitions $\mathbf{x}(1), \dots, \mathbf{x}(T)$, whose average is denoted by $\boldsymbol{\mu}$. In matrix notation, (1) becomes[2]

$$\widehat{\mathbb{S}}(\mathbf{X}) = \mathbf{X}\mathbf{P}\mathbf{X}^\top, \tag{2}$$

once defined $\mathbf{P}$ as the $T \times T$ matrix whose (s, t)-th entry is

$$\mathbf{P}_{ss} = \frac{1}{T} \quad \text{and} \quad \mathbf{P}_{st} = -\frac{1}{T^2 - T} \quad \text{if } s \neq t. \tag{3}$$

The usage of the covariance $\widehat{\mathbb{S}}$ to produce descriptors for classification tasks has been intensively studied [23], [24], [25], [26], [27], [28], [29], [17]. In particular, [23] proposed patch-specific covariance descriptors, efficiently computed with integral images. Other approaches rely on covariance to systematically encode mutual relationships inside the data and such idea was applied to many different applications such as face recognition [24], person identification [25] and more general classification tasks [26]. Further, covariance was proposed to measure similarities across data samples [27].

This latter direction actually grounds on the mathematical properties of positive definite matrices, exploiting Riemannian metrics on manifold for image classification: once moved from a finite to an infinite dimensional space, the performance enhances [28], [29] and only recently deep learning approaches have shown to be superior. However, one of the main limitation related to covariance matrix is that it only enables to capture linear inter-relationships [22]. For instance, principal component analysis actually exploits a covariance matrix to remove linear correlation of data points [30]. Among the attempts for modeling more complicated relationships, additional statistics, such as entropy and mutual information [26], and kernels [17] have been adopted. As a different paradigm, one can model non-linear behaviors by preliminary applying a preprocessing step and encode raw data by means of a transformation which increases the feature space. For instance, [15] applied such idea for spatial and temporal derivatives for gesture recognition, [26] considered both different color spaces and edge detectors for image classification, and [25] used filter bank responses as features to estimate head orientation. In this latter approach, once defined the feature map Φ and the transformed data matrix $\boldsymbol{\Phi}(\mathbf{X})$ whose t-th column is $\Phi(\mathbf{x}(t))$, the covariance (2) is now

[1] For convenience, in the following, we will concisely refer to the estimator $\widehat{\mathbb{S}}$ as the covariance itself, omitting the "sampling" attribute.

[2] For a matter of space, the technical proof of deriving equation (2) from (1) was moved to the Supplementary Material.

expressed by

$$\widehat{\mathbb{S}}(\mathbf{\Phi}(\mathbf{X})) = \mathbf{\Phi}(\mathbf{X})\mathbf{P}\mathbf{\Phi}(\mathbf{X})^\top. \tag{4}$$

Despite $\widehat{\mathbb{S}}(\mathbf{\Phi}(\mathbf{X}))$ is able to capture general relationships embedded in the raw data $\mathbf{X}$, the main bottleneck with (4) is the requirement of explicit computation for $\mathbf{\Phi}(\mathbf{X})$. Indeed, due to feature space augmentation performed by Φ, the higher dimensionality of such a matrix is more demanding in terms of both storage and computational cost required to calculate (4) instead of (2). Additionally, although infinite feature spaces are common for many classes of feature maps (*e.g.*, the one corresponding to a Gaussian kernel), this case has to be excluded in (4) since $\mathbf{\Phi}(\mathbf{X})$ is infinite dimensional and therefore impossible to compute exactly. In the following Section, we will face the problem of obtaining $\widehat{\mathbb{S}}$ without involving $\mathbf{\Phi}(\mathbf{X})$.

III. Method

Leveraging on the theory of kernel methods [31], every symmetric and positive definite kernel function $k \colon \mathbb{R}^{3n} \times \mathbb{R}^{3n} \to \mathbb{R}$ can be expressed as

$$k(\mathbf{x}, \mathbf{z}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle_{\mathcal{H}}, \tag{5}$$

where the inner product is computed in the Hilbert space[3] $\mathcal{H}$ which defines the range of the feature map $\Phi \colon \mathbb{R}^{3n} \to \mathcal{H}$. In (5), the kernel trick [31] replaces the arbitrary relationships in the original data space with a linear reformulation in $\mathcal{H}$: most importantly, Φ can be actually skipped, since only requiring the computation of the kernel k (*e.g.*, this happens for support vector machines [30]). In our case, we will employ k to obtain the representation $\widehat{\mathbb{S}}(k)$, equivalent to (4), that is $\widehat{\mathbb{S}}(k) = \widehat{\mathbb{S}}(\Phi(\mathbf{X}))$, while also skipping the computation of Φ. The following statement moves the first step in this direction.

Lemma 1. *Assume that there exist $\mathbf{h}_j \in \mathbb{R}^{3n}$ such that $\Phi(\mathbf{h}_j) = \mathbf{e}_j$ for every $j = 1, \dots, \dim(\mathcal{H})$, being $\mathbf{e}_j$ the unitary element of the canonical base of $\mathcal{H}$ as a vectorial space. Then, there exists a $\dim(\mathcal{H}) \times T$ matrix $\mathbf{K}[\mathbf{X}, \mathbf{h}]$, depending only on the kernel k, the data $\mathbf{X}$ and $\mathbf{h}_j$, such that, if we define $\widehat{\mathbb{S}}(k) = \mathbf{K}[\mathbf{X}, \mathbf{h}]\mathbf{P}\mathbf{K}[\mathbf{X}, \mathbf{h}]^\top$, we get $\widehat{\mathbb{S}}(k) = \widehat{\mathbb{S}}(\mathbf{\Phi}(\mathbf{X}))$.*

Proof. Using (4), the (i, j)-th entry of $\widehat{\mathbb{S}}(\mathbf{\Phi}(\mathbf{X}))$ rewrites

$$\widehat{\mathbb{S}}_{ij}(\mathbf{\Phi}(\mathbf{X})) = \sum_{s,t=1}^{T} \langle \Phi(\mathbf{x}(s)), \mathbf{e}_i \rangle_{\mathcal{H}} \mathbf{P}_{st} \langle \Phi(\mathbf{x}(t)), \mathbf{e}_j \rangle_{\mathcal{H}}. \tag{6}$$

In (6), once exploited the assumption that $\Phi(\mathbf{h}_j) = \mathbf{e}_j$, for some $\mathbf{h}_j$, we can define the $\dim(\mathcal{H}) \times T$ matrix $\mathbf{K}[\mathbf{X}, \mathbf{h}]$ whose (i, s)-th entry $k(\mathbf{x}(s), \mathbf{h}_i)$ is $\langle \Phi(\mathbf{x}(s)), \mathbf{e}_i \rangle_{\mathcal{H}} = \langle \Phi(\mathbf{x}(s)), \Phi(\mathbf{h}_i) \rangle_{\mathcal{H}}$ and consequently we deduce

$$\widehat{\mathbb{S}}(k) = \mathbf{K}[\mathbf{X}, \mathbf{h}]\mathbf{P}\mathbf{K}[\mathbf{X}, \mathbf{h}]^\top = \widehat{\mathbb{S}}(\mathbf{\Phi}(\mathbf{X})), \tag{7}$$

which proves the thesis. □

Lemma 1 certifies that we are able to compute the covariance in terms of the sole kernel k. However, some issues pertain to the practical feasibility of the assumption

$$\Phi(\mathbf{h}_j) = \mathbf{e}_j, \tag{8}$$

for any j, which is nevertheless fundamental for our purposes.

Actually, (8) is quite restrictive since the range of Φ is forced to contain the whole canonical base of $\mathcal{H}$. For instance, if $\mathcal{H} = \mathbb{R}^M$, (8) consists in a set of M equations that have to be solved in an M-dimensional space and, even if we assume that $\Phi(\mathbf{x}) = \mathbf{x}$, the resulting linear system can be either undetermined or impossible. Clearly, in case of a more general shape for Φ, it is not trivial to check whether the assumption (8) is verified. Hence, it seems natural to opt for a different feature map, which can replace Φ in generating the kernel function k, also satisfying (8). Thus, in the rest of the paper, we will focus on a specific class of stochastic feature maps $\mathbf{\Psi}$, actually fulfilling hypothesis (8), so that the induced linear kernel approximates k in a both stochastic and analytical sense. Therefore, we select the family of functions

$$k(\mathbf{x}, \mathbf{z}) = \sum_{\ell=0}^{\infty} a_\ell \langle \mathbf{x}, \mathbf{z} \rangle^\ell \tag{9}$$

where the dot product $\langle \mathbf{x}, \mathbf{z} \rangle$ is computed in $\mathbb{R}^{3n}$ and $a_\ell \geq 0$ for any ℓ. It is worth nothing that, due to the non-negativeness of these coefficients, since a linear combination of kernels is still positive definite, then (9) admits the representation (5). Also, (9) covers both finite and infinite linear combinations and therefore is comprehensive of a broad class of kernel functions. For instance, it is easily checked that (9) generalizes both the polynomial kernel $k(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle^\ell + a_0$ and the exponential-dot product kernel $k(\mathbf{x}, \mathbf{z}) = \exp\left(\dfrac{\langle \mathbf{x}, \mathbf{z} \rangle}{\sigma^2}\right)$, $\sigma > 0$. In this setting, we now introduce the following lemma which gives the fundamental tool to construct $\mathbf{\Psi}$.

Lemma 2. *Let $\boldsymbol{\omega} = [\omega_1, \dots, \omega_{3n}]$ a collection of $3n$ independent samples jointly distributed as a mixture of discrete Dirac's deltas and define $\psi(\mathbf{x}) = \langle \boldsymbol{\omega}, \mathbf{x} \rangle$. Then, the expectation of $\psi(\mathbf{x})\psi(\mathbf{z})$ under the distribution of $\boldsymbol{\omega}$ is*

$$\mathbb{E}_{\boldsymbol{\omega}}[\psi(\mathbf{x})\psi(\mathbf{z})] = \langle \mathbf{x}, \mathbf{z} \rangle. \tag{10}$$

Proof. Using the definition of ψ, the property of the mixture of Dirac's delta distribution and the linearity of the expectation $\mathbb{E}_{\boldsymbol{\omega}}$, the thesis comes after the following chain of equivalences

$$\mathbb{E}_{\boldsymbol{\omega}}[\psi(\mathbf{x})\psi(\mathbf{z})] = \mathbb{E}_{\boldsymbol{\omega}}[\langle \boldsymbol{\omega}, \mathbf{x} \rangle \langle \boldsymbol{\omega}, \mathbf{z} \rangle] = \mathbb{E}_{\boldsymbol{\omega}}\left[\sum_{i,j=1}^{3n} \omega_i \omega_j \mathbf{x}_i \mathbf{z}_j\right]$$
$$= \sum_{i,j=1}^{3n} \mathbb{E}_{\boldsymbol{\omega}}[\omega_i \omega_j] \mathbf{x}_i \mathbf{z}_j = \sum_{i,j=1}^{3n} \delta_{ij} \mathbf{x}_i \mathbf{z}_j = \langle \mathbf{x}, \mathbf{z} \rangle,$$

[3] For additional details about $\mathcal{H}$ as well as for an extended presentation of the proposed method, please, refer to the Supplementary Material.

where δ_{ij} denotes the Kronecker symbol. □

Once sampled a random number $N \in \mathbb{N}$ with probability $\frac{1}{p^{N+1}}$, define $\mathbf{\Psi}(\mathbf{x}) = \frac{1}{\sqrt{M}}[\Psi_1(\mathbf{x}), \dots, \Psi_M(\mathbf{x})]$ where $\Psi_1, \dots, \Psi_M$ are all identical copies of the function

$$\mathbf{x} \longmapsto \sqrt{a_N p^{N+1}} \prod_{j=1}^{N} \langle \boldsymbol{\omega}_j, \mathbf{x} \rangle, \tag{11}$$

where $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_N$ are independently distributed according to $\boldsymbol{\omega}$. Equation (11) and Lemma 2 allow to extend to our case [32, Lemma 7], which states that the linear kernel $\langle \mathbf{\Psi}(\mathbf{x}), \mathbf{\Psi}(\mathbf{z}) \rangle$ obtained through $\mathbf{\Psi}$ is an unbiased estimator of the original function $k(\mathbf{x}, \mathbf{z})$. Similarly, using the same arguments of Section 4.1 in [32], we obtain that $\langle \mathbf{\Psi}(\mathbf{x}), \mathbf{\Psi}(\mathbf{z}) \rangle \approx k(\mathbf{x}, \mathbf{z})$ uniformly over any compact set of $\mathbb{R}^{3n}$.

Since we proved that $\mathbf{\Psi}$ approximates the kernel k in the sense explained above, the final stage is solving the issue related to (8).

Proposition 1. *The map* $\mathbf{\Psi}$ *satisfies the assumption* (8)*, that is, for every* $i = 1, \dots, M$*, it results*

$$\frac{1}{\sqrt{M}}[\Psi_1(\mathbf{h}_i), \dots, \Psi_M(\mathbf{h}_i)] = \mathbf{e}_i. \tag{12}$$

Proof. The relationship (12) displays a system of equations, stochastically dependent on the randomness of $\mathbf{\Psi}$. Actually, in our case, it is enough to solve the system (12) and prove the existence of $\mathbf{h}_1, \dots, \mathbf{h}_M$ under a specific realization of N and $\boldsymbol{\omega}$, the two sources of randomness in $\mathbf{\Psi}$. In other words, we can solve (12) in a maximum likelihood sense by considering the samples of N and $\boldsymbol{\omega}$ which verify (12) with probability 1. Thus, we use a prior on N so that $N = 1$ and, once absorbed into $\mathbf{h}_i$ all the multiplicative constant defining $\mathbf{\Psi}$, then (12) becomes

$$[\langle \boldsymbol{\omega}_1, \mathbf{h}_i \rangle, \dots, \langle \boldsymbol{\omega}_M, \mathbf{h}_i \rangle] = \mathbf{e}_i, \quad i = 1, \dots, M. \tag{13}$$

Precisely, (13) is a linear system of size M in the M unknowns $\mathbf{h}_i$. If we then assume that the Dirac delta distribution of $\boldsymbol{\omega}_j$ is concentrated in j with probability 1, (13) is solvable if and only if $\langle \boldsymbol{\omega}_j, \mathbf{h}_i \rangle = \delta_{ij}$ for any $i, j = 1, \dots, M$. This is actually verified once chosen $\mathbf{h}_i$ to be the i-th element of the orthonormal basis of $\mathbb{R}^{3n}$. □

With Proposition 1, all issues related to the computability for $\widehat{\mathbb{S}}(k)$ is solved. Additionally, one can also easily understand that, with the previous choice of $\mathbf{h}_i$, once selected a linear kernel $k(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle$, then $\widehat{\mathbb{S}}(k)$ is equal to the $\widehat{\mathbb{S}}(\mathbf{X})$, so that the classical covariance is a particular case of our framework.

The theoretical discussion leads to derive Algorithm 1 and to apply the proposed kernelized covariance for the task of action and activity recognition. For a better understanding, we also visualize such pipeline in Figure 1.

Algorithm 1: Pseudo-code of our paradigm.

```
Input: Set of actions, kernel function $k$ as in (9).
Output: Kernelized covariance matrix $\widehat{\mathbb{S}}(k)$ (used as
        input to a classifier).

Procedure:
For each action, extract the data matrix $\mathbf{X}$ collecting all
  the $T$ temporal acquisitions $\mathbf{x}(1), \dots, \mathbf{x}(T)$, each of
  them encoding the 3D coordinates of the $n$ joints;
For each data matrix $\mathbf{X}$, select $\mathbf{h}_1, \dots, \mathbf{h}_M$ as in
  Proposition 1 and compute the Gram matrix $\mathbf{K}[\mathbf{X}, \mathbf{h}]$
  according to Lemma 1;
Compute the linear operator $\mathbf{P}$ defined in (3);
By means of $\mathbf{K}[\mathbf{X}, \mathbf{h}]$ and $\mathbf{P}$, computed in the previous
  steps, use (7) to calculate the kernelized covariance $\widehat{\mathbb{S}}(k)$;
```

Computational cost. The complexity of our trial-specific kernelized covariance is $O(M^2T^2)$. Thus, differently from previous approaches [27], [33], [28], [29], the proposed framework is very efficient if compared to the cubic complexity of methods like [33] which require eigen-decomposition. Under a mathematical point of view, our kernelized covariance is a natural generalization of the classical covariance matrix, which can be retrieved as a particular case in our paradigm once fixed the kernel function (9) to be a linear one. On the other hand, the computational cost still remains the same if compared with the classical covariance descriptor.

IV. Experimental results

In this section, we present the experimental results obtained with our *Kernelized-COV* method on different publicly available MoCap datasets for action recognition. Precisely, the following algorithms were compared in our experiments: *Region-COV* [23] (covariance region descriptor), temporal pyramid of covariance descriptors (*Hierarchy of COVs*) [16] and, finally, an infinite covariance operator which exploits Bregman divergence, namely *COV-*$J_{\mathcal{H}}$*-SVM* [29]. Furthermore, we also report the comparison against the recent state-of-the-art methods, namely *Ker-RP-POL* and *Ker-RP-RBF* [17].

In all the experiments, we followed [17] in performing SVM classification by means of a global log-Euclidean kernel applied upon Gram matrices, directly computed over joints coordinates, encoding each single trial. Nevertheless, differently from [17], in order to represent each multivariate time series of joints trajectories, the data encoding of any trial was realized through our kernelized covariance matrix $\widehat{\mathbb{S}}(k)$, where k is the exponential-dot product kernel (see Section III). For a fair comparison, our kernelization was plugged into the publicly available code[4] and, for classification, we used the *SVM and*

[4] http://www.uow.edu.au/~leiw/

TABLE I

Comparative performance of the proposed kernelized-COV benchmarking previous methods in the literature [23], [16], [29] based on covariance matrix and the state-of-the-art approach [17]. Best results in bold.

Method	MSR-Action3D	MSR-Daily-Activity	MSRC-Kinect12	HDM-05
Region-COV [23]	74.0%	85.0%	89.2%	91.5%
Hierarchy of COVs [16]	90.5%	-	91.7%	-
COV-$J_{\mathcal{H}}$-SVM [29]	80.4%	75.5%	89.2%	82.5%
Ker-RP-POL [17]	96.2%	**96.9%**	90.5%	93.6%
Ker-RP-RBF [17]	**96.9%**	96.3%	92.3%	96.8%
Kernelized-COV (proposed)	96.2%	96.3%	**95.0%**	**98.1%**

Kernel Methods Matlab Toolbox[5] using the wrapper directly provided by the authors. Finally, we fixed $M = 3n$ and, as done by [17], the kernel parameter $\sigma > 0$ is chosen by cross validation.

In all the experiments, we only used the 3D skeleton coordinates available in the following datasets:

- MSR-Action3D [18], where there are 20 classes of mostly sport-related action (*e.g.*, *jogging* or *tennis-serve*) involving 10 subjects. Since each subject performs each action 2 or 3 times, the overall number of trials is 567. For each of them, Kinect sensor is used to acquire depth maps, from which 20 joints are extracted to model the human pose of any of the human agents.
- MSR-Daily-Activity [19], captured by using a Kinect device and it is composed by 16 different classes related to every-day actions such as *read book* or *lie down on sofa*. All of them are performed by 10 subjects. The main difficulty of this dataset originates from the fact that any activity class is performed in an either standing/sitting position, with a consequent misleading motion pattern to mess up the classification.
- MSRC-Kinect12 [20], consisting of sequences of human movements, represented as body-part locations, and the associated gesture to be recognized by the system. 594 sequences of approximate total length of six hours and 40 minutes are collected from 30 people performing 12 gestures: in total, 6,244 gesture instances. The motion files contain Kinect estimated trajectories of 20 joints.
- HDM-05 [21], containing more than tree hours of systematically recorded and well-documented MoCap data using a 240Hz VICON system to acquire the gestures of 5 non-professional actors via 31 markers. Motion clips have been manually cut out and annotated into roughly 100 different motion classes: on average, 10-50 realizations per class are available.

In all cases, we used the same splits adopted in [17]: for MSR-Action3D, MSR-Daily-Activity and MSRC-Kinect12, training is performed on odd-index subject, while the even-index ones are left for testing (cross-subject pipeline of [18]), while, in HDM-05, the training split exploits all the data from the “`bd`” and “`mm`” subjects and testing is performed on “`bk`”, “`dg`” and “`tr`”.

Furthermore, for the HDM-05 dataset we removed some severely corrupted samples [16] and, as performed by [17], selected only the following classes: *clap above head*, *deposit floor*, *elbow to knee*, *grab high*, *hop both legs*, *jog*, *kick forward*, *lie down floor*, *rotate both arms backward*, *sit down chair*, *sneak*, *squat*, *stand up lie* and *throw basketball*. All the data are pre-processed in a common way. In particular, in MSR-Action3D and MSR-Daily-Activity, we computed the velocity and acceleration from the raw positions of the joints adopting either first and second order finite different scheme respectively as in [12].

Table I shows the results of *Kernelized-COV* on the four different datasets in comparison with all the other methods. Therein, in the case of MSR-Action3D and MSR-Daily-Activity, our proposed method is able to achieve comparable results with a small deviation from the state-of-the-art [17], but it outperforms all the other competitors. More impressively, on MSRC-Kinect12, *Kernelized-COV* improves the-state-of-the-art [17] by 2.7%. Even in the last dataset, namely HDM-05, the accuracy of the proposed method is 1.3% higher of the best score achieved by the other competitors. In this case, referring to [16], we did not report the accuracy on HDM-05 due to the different experimental settings: *Hierarchy of COVs* scored 95.41% on a simplified 11-class problem, while, in the same conditions, we scored 98.8%. Furthermore, it is worth noting that, on all the considered datasets our *Kernelized-COV* works even better than a recent infinite covariance operator [29], more discriminatively encoding the data.

The improvements in classification accuracies demonstrate the effectiveness of *Kernelized-COV*. Moreover, our proposed principled way of encoding non-linearities conveyed by the data is always superior to classical covariance based methods such as [23], [16], [29] and does not suffer the gap in performance showed by covariance representation in [17].

As a final remark, it is interesting to compare the performance of our *Kernelized-COV* with other not covariance-based methods. To this aim, we take into account the MSR-Action3D dataset and we compared with many previous approaches in the literature, already introduced in Section I. From this analysis, the results presented in Table II give a further evidence of the effectiveness of the proposed use of the kernelized covariance,

[5] http://asi.insa-rouen.fr/enseignants/~arakoto/toolbox/index.html

TABLE II
COMPARISON AGAINST OTHER CLASSICAL APPROACHES FOR ACTION AND ACTIVITY RECOGNITION FROM MOCAP DATA.

Method	MSR-Action3D
Action Graph [5]	79.0%
Random Occupancy Patterns [9]	86.0%
Actionlets [8]	88.2%
Pose Set [10]	90.0%
Moving Pose [12]	91.7%
Lie Group [14]	92.5%
Normal Vectors [13]	93.1%
Kernelized-COV (proposed)	**96.2%**

which is able to overcome [13], the best score reported, by a margin of 3.1%.

V. CONCLUSIONS & FUTURE PERSPECTIVES

This paper presents a principled mathematical paradigm to recover the applicability of kernel trick for covariance matrix, in order to better model more general class of relationships other than the linear ones. This enhances the descriptiveness of the classical covariance matrix which is retrievable as a particular case of our general theoretical framework. Experimentally, *Kernelized-COV* closes the gap between covariance and kernel-based representations in many action recognition datasets, namely MSR-Action3D, MSR-Daily-Activity, MSRC-Kinect12 and HDM-05. The proposed method is able to improve the previous best accuracies, setting the new state-of-the-art performance on the last two datasets.

As a future work, we either tackle the applicability of this novel framework to other classification problems and we will also investigate how a similar pipeline can be extended to more general classes of kernel functions.

REFERENCES

[1] T. B. Moeslund, A. Hilton, and V. Krüger, “A survey of advances in vision-based human motion capture and analysis,” *CVIU*, vol. 104, no. 2, pp. 90–126, 2006.

[2] M. Vrigkas, C. Nikou, and I. Kakadiaris, “A review of human activity recognition methods,” *Frontiers in Robotics and AI*, vol. 2, no. 28, 2015.

[3] Y. Sheikh, M. Sheikh, and M. Shah, “Exploring the space of a human action,” in *ICCV*, 2005.

[4] F. Lv and R. Nevatia, “Recognition and segmentation of 3d human action using hmm and multi-class adaboost,” in *ECCV*, 2006.

[5] W. Li, Z. Zhang, and Z. Liu, “Action recognition based on a bag of 3d points,” in *CVPR workshop*, 2010.

[6] X. Yang and Y. Tian, “Eigenjoints-based action recognition using naive bayes nearest neighbor,” in *CVPR workshop*, 2012.

[7] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, “Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition,” in *CVPR workshop*, 2013.

[8] J. Wang, Z. Liu, Y. Wu, and J. Yuan, “Mining actionlet ensemble for action recognition with depth cameras,” in *CVPR*, 2012.

[9] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, “Robust 3d action recognition with random occupancy patterns,” in *ECCV*, 2012.

[10] C. Wang, Y. Wang, and A. L. Yuille, “An approach to pose-based action recognition.” in *CVPR*, 2013.

[11] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo, “Space-time Pose Representation for 3D Human Action Recognition,” in *ICIAP workshop*, 2013.

[12] M. Zanfir, M. Leordeanu, and C. Sminchisescu, “The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection,” in *ICCV*, 2013.

[13] X. Yang and Y. Tian, “Super normal vector for activity recognition using depth sequences,” in *CVPR*, 2014.

[14] R. Vemulapalli, F. Arrate, and R. Chellappa, “Human action recognition by representing 3d skeletons as points in a lie group,” in *CVPR*, 2014.

[15] A. Sanin, C. Sanderson, M. T. Harandi, and B. C. Lovell, “Spatio-temporal covariance descriptors for action and gesture recognition,” *CoRR*, vol. abs/1303.6021, 2013.

[16] M. Hussein, M. Torki, M. Gowayyed, and M. El-Saban., “Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations,” *IJCAI*, 2013.

[17] L. Wang, J. Zhang, L. Zhou, C. Tang, and W. Li, “Beyond covariance: Feature representation with nonlinear kernel matrices,” in *ICCV*, 2015.

[18] W. Li, Z. Zhang, and Z. Liu, “Action recognition based on a bag of 3d points,” in *CVPR workshop*, 2010.

[19] J. Wang, Z. Liu, Y. Wu, and J. Yuan, “Mining actionlet ensemble for action recognition with depth cameras,” in *CVPR*, 2012.

[20] S. Fothergill, H. M. Mentis, P. Kohli, and S. Nowozin, “Instructing people for training gestural interactive systems,” in *ACM-CHI*, 2012.

[21] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, “Documentation mocap database HDM-05,” Universität Bonn, Tech. Rep. CG-2007-2, June 2007.

[22] J. D. Hamilton, *Time series analysis*. Princenton University Press, 1994.

[23] O. Tuzel, F. Porikli, and P. Meer, “Region covariance: A fast descriptor for detection and classification,” in *ECCV*, 2006.

[24] Y. Pang, Y. Yuan, and X. Li, “Gabor-based region covariance matrices for face recognition,” *TCSVT*, vol. 18, no. 7, pp. 989–993, 2008.

[25] D. Tosato, M. Spera, M. Cristani, and V. Murino, “Characterizing humans on riemannian manifolds,” *TPAMI*, vol. 35, no. 8, pp. 1972–1984, 2013.

[26] M. San Biagio, M. Crocco, M. Cristani, S. Martelli, and V. Murino, “Heterogeneous Auto-Similarities of Characteristics (HASC): Exploiting Relational Information for Classification,” in *ICCV*, 2013.

[27] M. San Biagio, S. Martelli, M. Crocco, M. Cristani, and V. Murino, “Encoding classes of unaligned objects using structural similarity cross-covariance tensors,” in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, 2013, pp. 133–140.

[28] M. Ha Quang, M. San Biagio, and V. Murino, “Log-Hilbert-Schmidt metric between positive definite operators on Hilbert spaces,” in *NIPS*, 2014.

[29] M. Harandi, M. Salzmann, and F. Porikli, “Bregman divergences for infinite dimensional covariance matrices,” in *CVPR*, 2014.

[30] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[31] B. Schölkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, ser. Adaptive Computation and Machine Learning, 2002.

[32] P. Kar and H. Karnick, “Random features maps for dot product kernels,” in *JMLR*, 2012.

[33] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi, “Kernel Methods on the Riemannian Manifold of Symmetric Positive Definite Matrices,” in *CVPR*, 2013.