An extension of kernel learning methods using a modified Log-Euclidean distance for fast and accurate skeleton-based Human Action Recognition

Enjie Ghorbel[a,b,*], Jacques Boonaert[a], Rémi Boutteau[b], Stéphane Lecoeuche[a], Xavier Savatier[b]

[a] *IMT Lille Douai, Univ. Lille, Unité de Recherche Informatique Automatique, F-59000 Lille, France*
[b] *Normandie Univ, UNIROUEN, ESIGELEC, IRSEEM, 76000 Rouen, France*

Abstract

In this article, we introduce a fast, accurate and invariant method for RGB-D based human action recognition using a Hierarchical Kinematic Covariance (HKC) descriptor.

Recently, non singular covariance matrices of pattern features which are elements of the space of Symmetric Definite Positive (SPD) matrices, have been proven to be very efficient descriptors in the field of pattern recognition.

However, in the case of action recognition, singular covariance matrices cannot be avoided because the dimension of features could be higher than the number of samples. Such covariance matrices (non singular and singular) belong to the space of Symmetric Positive semi-Definite (SPsD) matrices.

Thus, in order to classify actions, we propose to adapt kernel methods such as Support Vector Machines (SVM) and Multiple Kernel Learning (MKL) to the space of SPsD matrices by using a perturbed Log-Euclidean distance (Arsigny et al., 2006). The mathematical validity of this perturbed distance (called Modified Log-Euclidean distance) for SPsD is therefore studied.

The offline experiments are conducted on three challenging benchmarks, namely MSRAction3D, UTKinect and Multiview3D datasets. A fair comparison demonstrates that our approach competes with state-of-the-art methods in terms of accuracy and computational latency. Finally, our method is extended to an online scenario and experiments on MSRC12 prove the efficiency of this extension.

Keywords: Kernel methods, Symmetric Positive semi-Definite matrices, Human action recognition, SVM, covariance matrices, RGB-D cameras, Log-Euclidean distance.

1. Introduction

Automatically recognizing human actions represents an expanding research topic in the areas of computer vision and pattern recognition. This phenomenon is due to the wide range of human action applications such as e-health, video surveillance, Human Computer Interaction (HCI), entertainment, etc. The most common acquisition system used to recognize actions is surely the RGB camera. Detailed surveys of RGB-based action recognition methods can be found in (Poppe, 2010; Weinland et al., 2011). Unfortunately, these methods suffer from some limitations: their performance is negatively affected by occlusions, view-point variation, illumination changes and body segmentation.

With the availability of low-cost RGB-D cameras, a renewed interest for action recognition has been observed. Additionally to the classical RGB images, this kind of camera also provides depth images. Furthermore, the recent algorithm proposed by Shotton et al. (2013) allows the real-time human skeleton extraction from depth maps. Thus, new methods (Rahmani et al., 2016; Amor et al., 2016; Brun et al., 2016; Liu et al., 2016) have been pro-

[*] Corresponding author: Tel.: +33-662-485-356;
Email address: author@author.com (Enjie Ghorbel)

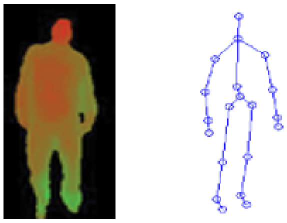

Figure 1: an example of depth (left) and skeleton (right) modalities

posed exploiting these new modalities, namely depth images and skeleton sequences (Figure 1).

Recent studies have shown that depth-based methods are generally more robust to noise and occlusions, while skeleton-based methods are more robust to view-point variation and are faster to compute (Hammouche et al., 2016; Ghorbel et al., 2015).

Since the rapidity of calculation is a very important factor in real-world applications, we consider that the skeleton modality is the best choice if the goal is to realize a trade-off between computational latency and recognition accuracy (Ghorbel et al., 2015).

Action recognition which is a sub-field of pattern recognition, lies at the crossroads between two research areas and can be therefore decomposed into two main steps:

1) Descriptor computation, involving computer vision. First, relevant features are extracted from each instance. Second, these features are used to compute a unique size motion descriptor. This descriptor is expressed in a specific feature space to make different instances comparable.

2) Classification, involving machine learning. Based on the descriptors extracted from the annotated training data, a model of classification is learned to divide the feature space in significant regions according to the different action labels.

These two steps are closely intertwined. The classification step should be adapted to the chosen feature space and the feature space has to be designed to enhance the classifier performance.

In this paper, we focus on covariance descriptors which have attracted great interest of researchers. In 2006, they have been introduced as descriptors for the first time in the field of computer vision (Tuzel et al., 2006). Then, these features have been applied to object recognition (Tuzel et al., 2006), classification of image sets (Wang et al., 2012b), pedestrian detection (Jayasumana et al., 2013), face recognition (Pang et al., 2008), action recognition (Hussein et al., 2013), etc. This popularity is mainly due to the good properties of covariance matrices. Indeed, they can be used to fuse heterogeneous features, are robust to occlusion and partially invariant to rotation and scale. They also contain the information of correlation between features and are therefore very informative. Furthermore, it has been shown in two recent papers that covariance descriptors are adapted to the case of online action recognition (Kviatkovsky et al., 2014; Tang et al., 2018).

To classify non singular covariance descriptors (which are considered as elements of the space of SPD matrices), various classification algorithms, initially designed for euclidean spaces such as kNN and kernel methods, have been extended to the non linear space of SPD matrices as in (Jayasumana et al., 2013; Pang et al., 2008).

1.1. Problem formulation: Covariance matrices in action recognition and the space of Symmetric Positive semi-definite matrices

Let $\mathbf{x_i} \in \mathbf{R}^d$ be a d-dimensional feature vector $\forall i$ and let us suppose that $\mathbf{D}_{d \times N} = [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}]$ represents the data matrix, N being the number of samples. In (Tuzel et al., 2006), the region covariance descriptor is calculated as follows:

$$\mathbf{C} = \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{x_i} - \mu)(\mathbf{x_i} - \mu)^T \qquad (1)$$

with μ the mean of samples. Therefore, the covariance descriptor C represents a $d \times d$ matrix which is assumed to be non singular and to be consequently an element of the space of Symmetric Positive Definite (SPD) matrices denoted by Sym_d^{+*}.

The assumption claiming that covariance matrices are non singular can be reasonable for applications using descriptors for which N is largely superior to the number of features d, such as region descriptors. In these applications, the required number of features is very small compared to the number of samples (pixels in the case of (Tuzel et al., 2006)).

However, this assumption is not valid anymore, if an important number of features is used. In action recognition, it is very common to use d features, with $d > N$ (N represents the number of frames). This fact leads to singular covariance matrices.

In a more formal manner, if the covariance matrix is full-ranked ($rank(\mathbf{C}) = d$), the matrix is not singular and is therefore SPD ($C \in Sym_d^{+*}$). Nonetheless, the rank of a $d \times d$ covariance matrix respects this inequality $rank(\mathbf{C}) \leq min(d, n - 1)$ and it can be noted that if $d > n$, then $rank(\mathbf{C}) < d$, implying the singularity of the matrix $\mathbf{C}$. Such matrices are not positive definite since they have at least one eigenvalue equals to zero. In reality, these matrices are Symmetric Positive semi-Definite (SPsD). Indeed, $d \times d$ covariance matrices (non singular and singular) are elements of the space of SPsD matrices denoted by Sym_d^+.

To overcome this numerical limitation, the majority of papers have chosen to work with a very restricted number of features $d < n$ (Hussein et al., 2013; Kviatkovsky et al., 2014). Then, various geodesic distances have been proposed for the space Sym_d^{+*} making the classification in Sym_d^{+*} possible. Nevertheless, if the use of a more important amount of features is needed for a better discrimination, the distance-based classification methods developed for Sym_d^{+*} become unsuitable. Therefore, the main problem would be to know how to generalize distance-based machine learning algorithms to the space Sym_d^+?

1.2. Contributions

In this article, we propose four main contributions:

a) The development and the validation of two different covariance descriptors based on the skeleton kinematic information for human action recognition.

b) The mathematical analysis of the perturbed Log-Euclidean distance for positive semi-definite matrices used to extend kernel-based methods and consequently to classify human actions (based on covariance descriptors).

c) A fair comparison of our method with state-of-the-art approaches (by collecting available codes) in terms of computational latency and accuracy on three different benchmarks.

d) The online extension of our approach using a simple sliding window.

This paper is organized as follows: Section 2 presents an overview of the state-of-the art. Then, Section 3 summarizes the mathematical tools used in this work. In Section 4, the perturbed Log-Euclidean distance that we call Modified Log-Euclidean distance is mathematically analyzed while in Section 5, kernel methods are extended to the space Sym_d^+. In Section 6, two novel human action descriptors based on covariance matrices are introduced and the proposed extension of kernel methods are used for classification. Section 7 presents the experiments realized on three challenging benchmarks. Section 8 is dedicated to the extension of our method to online action recognition by conducting experiments on the dataset MSRC12. Finally, Section 9 formulates conclusion and perspectives.

2. Related Work

In this section, we present an overview of the state-of-the-art related to the two topics of interest: skeleton-based action recognition and distance-based learning using covariance matrices.

2.1. Skeleton-based action recognition

As mentioned in Section 1, skeleton-based descriptors have two main advantages compared to other descriptors: they are relatively accurate, robust to viewpoint variation and generally fast to compute (low computational latency) (Ghorbel et al., 2015).

A recent survey (Zhu et al., 2016) has categorized action recognition methods into two distinct groups based on the representation of actions: hand-crafted representations and learning-based representations.

Instead of selecting specific features, learning-based methods learn by themselves the adequate features as in (Hou et al., 2016; Wang et al., 2016; Qiao et al., 2017).

Hand-crafted methods are the most common approaches used in the literature (Ohn-Bar and Trivedi, 2013; Vemulapalli et al., 2014; Wang et al., 2012a). They are based on the classical schema of action recognition, where low-level features are first extracted, the final descriptor is then modeled using low level features and a classifier is finally used to train a classification model such as Support Vector Machine (SVM) or k Nearest Neighbors (kNN).

Since covariance matrices belong to the group of hand-crafted descriptors, we propose to review only this kind of methods. Hand-crafted methods can be divided according to the descriptor nature into four sets, namely: pose-based descriptors, geometric descriptors, kinematic descriptors and finally statistical descriptors.

2.1.1. Pose-based descriptors

Pose-based descriptors represent the most intuitive skeleton-based representation. The idea is to directly use the information of joint positions to build a descriptor.

Inspired by the bag-of-word representation, Li et al. (2010) introduced the 3D bag of points. Then, using these bag of points, an action graph is built. On the other hand, Xia et al. (2012) proposed to construct the histogram of 3D joints. Nevertheless, these two approaches remain sensitive to anthropometric variability because of the use of the absolute joint positions. To overcome this limitation, Yang and Tian (2012) designed EigenJoints which are calculated thanks to the concatenation of spatial and temporal distances between joints. The word ”eigen” refers to principal component analysis (PCA) applied on the features to reduce their high dimension. This first generation of representation is interesting but is clearly less accurate than more recent ones. Over the last years, it has been noted the emergence of more sophisticated descriptors.

2.1.2. Geometric descriptors

This kind of descriptor is designed by representing skeleton motions using geometric concepts. In (Evangelidis et al., 2014), skeletal quads are introduced. These features represent quadruples composed of four adjacent joints which contain the information of similarity between segments. We can also cite the work of Vemulapalli et al. (2014), where actions have been represented by associating to each couple of neighbour segments a transformation matrix T (with T an element of the Special Euclidean group $SE(3)$). Each skeleton (composed of n joint connections) of the sequence is represented by a point of $SE(3)^n$. These points evolving over time are interpolated on the Lie algebra of $SE(3)^n$ named $se(3)^n$. The obtained curves are therefore compared via a Dynamic Time Warping algorithm (DTW). This algorithm has shown its efficiency, however the high amount of approximation can lead to low accuracy (if the data are noisy) and to an important calculation time.

2.1.3. Kinematic descriptors

Since skeleton representation has been very often used in bio-mechanic studies (Johansson, 1973), many papers have based their work on kinematic entities such as position, velocity and acceleration of joints. These values are computed thanks to the joint position information (Zanfir et al., 2013; Ghorbel et al., 2016). Zanfir et al. (2013) proposed to concatenate these features and to weight each term by an empirical value. They classified actions using a kNN algorithm. In (Ghorbel et al., 2016), the discrete kinematic values are interpolated using a cubic spline interpolation. To make the features invariant to velocity variation and anthropometric variability, a temporal and a skeleton normalization are respectively proposed. A linear SVM is then trained to carry out classification.

2.1.4. Statistical descriptors

This class of descriptor use statistical tools in order to propose a discriminative action representation. It can be noted in the state-of-the-art an important interest for action covariance descriptors. In (Hussein et al., 2013), joint positions are used to build a covariance descriptor. Because covariance matrices are symmetric, only the upper triangle of the matrix is considered and converted into a vector. This vector is then used to train a linear SVM. This work did not take into account the particular geometry of the covariance matrix space assuming a space vector structure. In (Tang et al., 2018), where an action recognition framework is proposed, covariance matrices are assumed to be symmetric positive definite. If the matrix is singular, the nearest symmetric positive definite matrix is found. A kNN algorithm is trained to classify actions using a geodesic distance defined on Sym_d^{+*}, the Log-Euclidean distance instead of using the Euclidean distance.

2.2. Distance-based learning using covariance matrices

As mentioned in the Section 1, covariance matrices have been widely used in computer vision for tasks such as object recognition, face recognition, pedestrian recognition, etc. Given that non singular covariance matrices are elements of the Riemannian manifold of the SPD matrices, many researchers have made attempts to formulate the geodesic distance of the space Sym_d^{+*}. Indeed, these distances are important knowing that they can be used to extend meaningful distance-based machine learning algorithms. Förstner and Moonen (2003) introduced the Affine-Invariant distance by considering the Riemannian structure of Sym_d^{+*}. To alleviate the excessive execution time required to calculate Affine-Invariant distance, a novel distance has been introduced by Arsigny et al. (2006), called the Log-Euclidean distance. Other distances for the space Sym_d^{+*} have been proposed such as Stein distance (Sra, 2011), Cholesky distance (Klingenberg, 2013), etc. Nevertheless, the most popular remain the Affine-Invariant and the Log-Euclidean distances as they take into account the Riemannian geometry of the SPD matrix space. Based on these distances, distance-based learning algorithms for the space of SPD matrices have been proposed in order to make use of SPD matrices as descriptors in pattern recognition applications. In (Tang et al., 2018; Kviatkovsky et al., 2014), Affine-Invariant and Log-Euclidean distances are respectively used to classify actions. Recently, Jayasumana et al. (2013) have extended Radial Basis Function (RBF) kernel to the space of SPD matrices. After that, this novel kernel has been combined with different kernel-based algorithms such as SVM, MKL and PCA.

In the context of action recognition, obtained covariance matrices are mostly singular because of the need of a too important number of features, as explained in Section 1.1. Therefore, the distance-learning methods become unsuitable since singular covariance matrices are not SPD, but are Symmetric Positive semi-Definite (SPsD). Some attempts have been done in order to overcome this limitation. The theoretical paper of Bonnabel and Sepulchre (2009) proposed a metric for SPsD matrices of fixed rank. However, it is difficult to ensure a fixed rank for all covariance matrices. For this reason, some researchers have proposed to apply a perturbation on usual SPD distances as in (Wang et al., 2012c; Tang et al., 2018). However, in both papers, the applied perturbation has not been analyzed theoretically and experimentally. Thus, we propose to use a perturbed Log-Euclidean distance (a distance designed for the SPD space) after proving its mathematical validity and studying its experimental behavior. Instead of directly using it to classify actions as presented in (Tang et al., 2018), we propose to make use of it to extend kernel-based classification.

3. Mathematical background

In this section, we review the mathematical tools related to our method. First, we present the mathematical notations used in this paper. Second, we present the Riemannian manifold of SPD matrices. Then, we recall the recent work of (Jayasumana et al., 2013) who have extended kernel learning approaches to data expressed in the Riemannian manifold of SPD matrices.

3.1. Notations

The transpose of a matrix $\mathbf{M}$ is denoted $\mathbf{M^T}$.

The inverse of an invertible matrix $\mathbf{M}$ is denoted $\mathbf{M^{-1}}$.

The diagonal matrix of a a diagonalizable matrix $\mathbf{A}$ is denoted by $\mathbf{D_A}$,

$Sym_d^{+*}(\mathbb{R})$ is the space of $d \times d$ Symmetric Positive Definite (SPD) matrices.

$Sym_d^{+}(\mathbb{R})$ is the space of Symmetric Positive semi-Definite $d \times d$ matrices.

3.2. Riemannian manifold of Symmetric Positive Definite matrices

The space of Symmetric Positive Definite $d \times d$ matrices Sym_d^{+*} represents one of the well-known example of Riemannian manifold (Arsigny et al., 2007). Riemannian manifolds are not necessarily linear and measuring similarity between its elements with a Euclidean distance is very often unsuitable.

Thus, many attempts have been made in computer vision to propose an appropriate geodesic distance such as Affine-Invariant distance (Förstner and Moonen, 2003), Log-Euclidean distance (Arsigny et al., 2006), Cholesky distance (Klingenberg, 2013), Root Stein Divergence distance (Sra, 2011), etc.

In what follows, we will only present the Log-Euclidean (Arsigny et al., 2006), which is needed for the understanding of this paper.

Log-Euclidean distance. In (Arsigny et al., 2006), the authors introduce a novel distance for Sym_d^{+*} in order to overcome the high complexity of the Affine-Invariant distance (Förstner and Moonen, 2003).

The Log-Euclidean distance d_{LE} between the two matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ is therefore defined by Equation (2).

$$d_{LE}(\mathbf{M}_1, \mathbf{M}_2) = \|Log(\mathbf{M}_1) - Log(\mathbf{M}_2)\|_F \tag{2}$$

It is invariant to inversion, to translation in the logarithmic space but is not completely invariant to affine transformations (contrary to Affine-Invariant distance). The main advantage of this distance is its rapidity of calculation.

3.3. A generalization of RBF-kernel learning for the space of Symmetric Positive Definite matrices

Kernel learning methods such as Support Vector Machines (SVM) and Multiple Kernel Learning (MKL) have shown their efficiency in a wide range of applications. Nonetheless, these methods are meaningful only if the features are expressed in a vector space. Nowadays, many attempts have been made to generalize these methods to non linear spaces such as Riemannian manifolds. Recently, Jayasumana et al. (2013) have extended machine learning methods based on the Radial Basis Function (RBF) kernel to the space Sym_d^{+*}. We recall the theorem demonstrated in their paper:

Theorem 3.1. *Let (M, d) be a space equipped with a distance d and let $k : M \times M \rightarrow \mathbb{R}$ be a function with $K_G^M(\mathbf{x}_i, \mathbf{x}_j) = exp(-\frac{d^2(\mathbf{x}_i,\mathbf{x}_j)}{2\sigma^2})$, and $\mathbf{x}_i, \mathbf{x}_j \in M$. Therefore, K_G^M is a positive definite kernel $\forall \sigma$ if and only if it exists a prehilbertian space V and a function $\phi : M \rightarrow V$ with $d(\mathbf{x}_i, \mathbf{x}_j) = \|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|_V$.*

Based on the stated theorem, the authors have finally formulated Corollary 3.1.

Corollary 3.1. *Let suppose that $K_G^{SPD} : Sym_d^{+*} \times Sym_d^{+*} \rightarrow \mathbb{R}$ with $K_G^{SPD}(\mathbf{x}_i, \mathbf{x}_j) = exp(-\frac{d_{LE}^2(\mathbf{x}_i,\mathbf{x}_j)}{2\sigma^2})$ and $d_{LE}(\mathbf{x}_i, \mathbf{x}_j) = \|log(\mathbf{x}_i) - log(\mathbf{x}_j)\|_F$ for $\mathbf{x}_i, \mathbf{x}_j \in Sym_d^{+*}$. Then, K_G^{SPD} is a positive definite kernel $\forall \sigma \in \mathbb{R}$.*

It is easy to notice that the distance d_{LE} is unsuitable to SPsD matrices. Let $\mathbf{x}$ be a singular SPsD matrices. Therefore, $\mathbf{x}$ is diagonalizable and a transformation matrix $\mathbf{P}$ exists such as $\mathbf{x} = \mathbf{P}^{-1}\mathbf{D}_\mathbf{x}\mathbf{P}$. Therefore, $Log(\mathbf{x})$ is equal to $\mathbf{P}^{-1}Log(\mathbf{D}_\mathbf{x})\mathbf{P}$. However, because $\mathbf{x}$ is positive semi-definite, the diagonal matrix $\mathbf{D}_\mathbf{X}$ contains at least one eigenvalue which is equal to 0, leading to the calculation of $Log(0)$ which is not defined. To avoid this limitation, we propose a simple perturbation of the Log-Euclidean distance in the next section, allowing the calculation of distances between SPsD matrices, despite the presence of null eigenvalues.

4. Perturbation of the Log-Euclidean distance: A Distance for the space of Symmetric Positive semi-Definite matrices Sym_d^+

Instead of proposing a distance by analyzing the particular geometry of Sym_d^+, we modify the Log-Euclidean designed for Sym_d^{+*} which has been already used to extend kernel methods. This subsection describes step by step the formulation and the validity of the proposed distance that is called Modified Log-Euclidean distance.

First, we state a theorem showing that it exists a one-to-one correspondence between SPsD and a subset of SPD. Then, based on this theorem, we construct the distance by using a mapping function between these two spaces thanks to an extension of the Log-Euclidean distance.

Theorem 4.1. *$\forall \epsilon > 0$, it exists a bijective relation ψ between Sym_d^+ and a set S defined as:*

$$\begin{array}{cccc} \psi & : & Sym_d^+ & \rightarrow & S \\ & & \mathbf{M} & \mapsto & \mathbf{M} + \epsilon\mathbf{I}_d \end{array}$$

with $S \subset Sym_d^{+*}$. $\mathbf{I}_d$ represents the Identity matrix of dimension d.

[Proof of Theorem 4.1]

First of all, we show that $\forall \mathbf{M} \in Sym_d^+$, $\psi(\mathbf{M})$ is Symmetric Positive Definite (SPD):

1) Symmetric: $\mathbf{M}$ is symmetric and $\mathbf{I}_d$ is symmetric, as well as $\epsilon\mathbf{I}_d$. Therefore, $\mathbf{M} + \epsilon\mathbf{I}_d$ is symmetric.

2) Positive definite: Let $\mathbf{X} \in \mathbb{R}^d$ such as $\mathbf{X} = [x_1, x_2, ..., x_d]^T$,

$$\begin{aligned} \mathbf{X}^T\psi(\mathbf{M})\mathbf{X} &= \mathbf{X}^T(\mathbf{M} + \epsilon\mathbf{I}_d)\mathbf{X} \\ &= \mathbf{X}^T\mathbf{M}\mathbf{X} + \epsilon\mathbf{X}^T\mathbf{X} \\ &= \mathbf{X}^T\mathbf{M}\mathbf{X} + \epsilon\sum_{i=1}^{d} x_i^2 > 0 \end{aligned} \tag{3}$$

We can deduce that $\phi(\mathbf{M})$ is symmetric positive definite and consequently $S \subset Sym_d^{+*}$.

For a fixed $\epsilon > 0$, let us suppose that $\mathbf{Y} \in S$. Then, it exists therefore a matrix $\mathbf{M} \in Sym_d^+$ with $\mathbf{Y} = \mathbf{M} + \epsilon\mathbf{I}_d$. So, $\mathbf{M} = \mathbf{Y} - \epsilon\mathbf{I}_d$ which is unique. Thus , the relation ψ is proven to be bijective.

The distance on the space Sym_d^+ is computed based on the Log-Euclidean proposed for the space Sym_d^{+*}, using the function $\psi : Sym_d^+ \rightarrow Sym_d^{+*}$ defined as $\psi(\mathbf{M}) = \mathbf{M} +$

$\epsilon\mathbf{I}_d$ for $\epsilon > 0$. Let $\mathbf{A}, \mathbf{B}$ be two elements of Sym_d^+ and let suppose that $\mathbf{A}_1 = \mathbf{A} + \epsilon\mathbf{I}_d$ and $\mathbf{B}_1 = \mathbf{B} + \epsilon\mathbf{I}_d$. We define the Modified Log-Euclidean (MLE) distance d_{MLE} between A and B as follows:

$$\begin{aligned} d_{MLE}(\mathbf{A}, \mathbf{B}) &= \|Log(\psi(\mathbf{A})) - Log(\psi(\mathbf{B}))\|_F \\ &= \|Log(\mathbf{A} + \epsilon\mathbf{I}_d) - Log(\mathbf{B} + \epsilon\mathbf{I}_d)\|_F \\ &= \|Log(\mathbf{A}_1) - Log(\mathbf{B}_1)\|_F = d_{LE}(\mathbf{A}_1, \mathbf{B}_1) \end{aligned} \tag{4}$$

This extension is particularly useful for singular matrices (with a null determinant). Choosing ϵ very small compared to the Eigen-values allows the construction of a mapping function which constraint the matrices to be symmetric positive definite matrices without making them too far from their initial position. Thus, the distance is measured on the space Sym_d^{+*} and the Modified-Log-Euclidean metric inherit many properties of Log-Euclidean distance. In the following, we show first that this measure validates all the distance conditions. Then, we show that if ϵ is chosen very small, the approximation does not extremely affect the calculation of the distance.

4.1. Distance for Sym_d^+

Here, we show the validity of the proposed distance. Let $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ be elements of Sym_d^+. We suppose that $\mathbf{A}_1 = \mathbf{A} + \epsilon\mathbf{I}$, $\mathbf{B}_1 = \mathbf{B} + \epsilon\mathbf{I}$ and $\mathbf{C}_1 = \mathbf{C} + \epsilon\mathbf{I}$. The Modified Log-Euclidean distance d_{MLE} is proven to be a distance on Sym_d^+ because the 4 necessary conditions are respected, namely:

1) Positivity: $d_{MLE}(\mathbf{A}, \mathbf{B}) = \|Log(\mathbf{A} + \epsilon\mathbf{I}_d) - Log(\mathbf{B} + \epsilon\mathbf{I}_d)\|_F = \|Log(\mathbf{A}_1) - Log(\mathbf{B}_1)\|_F \geq 0$ because $\mathbf{A}_1, \mathbf{B}_1 \in Sym_d^{+*}$.

2) Separation: $d_{MLE}(\mathbf{A}, \mathbf{B}) = \|Log(\mathbf{A}_1) - Log(\mathbf{B}_1)\|_F \Leftrightarrow \mathbf{A}_1 = \mathbf{B}_1 \Leftrightarrow \mathbf{A} = \mathbf{B}$.

3) Symmetry: $d_{MLE}(\mathbf{B}, \mathbf{A}) = \|Log(\mathbf{B}_1) - Log(\mathbf{A}_1)\|_F = d_{LE}(\mathbf{B}_1, \mathbf{A}_1) = d_{LE}(\mathbf{A}_1, \mathbf{B}_1) = d_{MLE}(\mathbf{A}, \mathbf{B})$.

4) Triangle Inequality: $d_{LE}(\mathbf{A}_1, \mathbf{C}_1) \leq d_{LE}(\mathbf{A}_1, \mathbf{B}_1) + d_{LE}(\mathbf{B}_1, \mathbf{C}_1) \Leftrightarrow d_{MLE}(\mathbf{A}, \mathbf{C}) \leq d_{MLE}(\mathbf{A}, \mathbf{B}) + d_{MLE}(\mathbf{B}, \mathbf{C})$.

4.2. Choice of the parameter ϵ

In this subsection, we show that if ϵ is chosen very small compared to covariance matrix eigenvalues, the approximation of distance is relatively accurate even if this measure is calculated in the space of SPD Matrices.

Since the matrices of the space Sym_d^+ are symmetric, they are also diagonalizable. Let $\mathbf{D}_M$ be the diagonal matrix obtained by the diagonalization of $\mathbf{M}$ and $\mathbf{P}$ be the transformation matrix satisfying $\mathbf{M} = \mathbf{P}\mathbf{D}_M\mathbf{P}^{-1}$ such as the eigenvalues in the matrix $\mathbf{D}_M$ are organized from the smallest to the highest eigenvalue (ensuring the uniqueness of P). We recall that $Log(\mathbf{M}) = \mathbf{P}Log(\mathbf{D}_M)\mathbf{P}^{-1}$. Thus, the calculation of the logarithm depends widely from the eigenvalues of $\mathbf{M}$.

Let suppose that $\mathbf{M}_1 = \psi(\mathbf{M})$. As it belongs to Sym_d^{+*}, $\mathbf{M}_1$ is also diagonalizable. Thus, it exists a matrix $\mathbf{P}_1$ with $\mathbf{M}_1 = \mathbf{P}_1\mathbf{D}_{M_1}\mathbf{P}_1^{-1}$ such as the eigenvalues in the matrix $\mathbf{D}_{M_1}$ are organized from the smallest to the highest eigenvalue. In fact,

$$\begin{aligned} \mathbf{D}_{\mathbf{M}_1} &= \mathbf{P}_1^{-1}\mathbf{M}_1\mathbf{P}_1 \\ &= \mathbf{P}_1^{-1}(\mathbf{M} + \epsilon\mathbf{I}_d)\mathbf{P}_1 \\ &= \mathbf{P}_1^{-1}\mathbf{M}\mathbf{P}_1 + \epsilon\mathbf{I}_d \end{aligned} \tag{5}$$

Since $\mathbf{D}_{M_1}$ is an ordered diagonal matrix and $\mathbf{I}_d$ is an Identity matrix then $\mathbf{P}_1^{-1}\mathbf{M}\mathbf{P}_1$ is a diagonal matrix containing ordered eigenvalues of M and $\mathbf{P}_1 = k\mathbf{P}$, with $k \in \mathbb{R}$. Indeed, the order ensures the uniqueness of the transformation matrix with a variable factor scale.

$$\mathbf{D}_{\mathbf{M}_1} = (k\mathbf{P}^{-1})\mathbf{M}(k^{-1}\mathbf{P}) + \epsilon\mathbf{I}_d = \mathbf{P}^{-1}\mathbf{M}\mathbf{P} + \epsilon\mathbf{I}_d = \mathbf{D}_M + \epsilon\mathbf{I}_d \tag{6}$$

We conclude that:

$$(\lambda_1)_i = (\lambda)_i + \epsilon \tag{7}$$

$(\lambda)_i$ and $(\lambda_1)_i$ for $i = 1...d$ respectively represent the ordered eigenvalues of $\mathbf{M}$ and $\mathbf{M}_1$. With the analysis of this result, it can be noted that if ϵ is very small compared to $(\lambda)_i, \forall i \in [\![1, d]\!]$, the approximation is accurate enough. More details and practical experimentation concerning this parameter will be given in Section 7.4.4.

5. RBF-Kernel methods Symmetric Positive semi-Definite space

In this section, the RBF kernel based methods are extended to the space SPsD Sym_d^+ using the MLE distance. As the distance d_{MLE} between two matrices $\mathbf{A}, \mathbf{B}$ which are symmetric positive semi-definite leads to compare the distance d_{LE} between two symmetric positive definite matrices, it is easy to show that d_{MLE} is negative definite as shown for the Log-Euclidean distance in (Jayasumana et al., 2013).

To respect Mercer's theorem, the kernel has to be positive definite. Consequently, for an RBF kernel, the distance used should be negative definite (Schoenberg, 1938). Thus, Theorem 3.1 induces the following Corollary 5.1.

Corollary 5.1. *Let $K_G^{SPsD} : Sym_d^+ \times Sym_d^+ \rightarrow \mathbb{R}$ be a kernel such as $K_G^{SPsD}(\mathbf{x}_i, \mathbf{x}_j) = exp(-\frac{d_{MLE}^2(\mathbf{x}_i,\mathbf{x}_j)}{2\sigma^2})$ and $d_{MLE}(\mathbf{x}_i, \mathbf{x}_j) = ||Log(\mathbf{x}_i + \epsilon\mathbf{I}_d) - Log(\mathbf{x}_j + \epsilon\mathbf{I}_d)||_F$ for $\mathbf{x}_i, \mathbf{x}_j \in Sym_d^+$. Then, K_G^{SPsD} is a positive definite kernel $\forall\sigma \in \mathbb{R}$ and $\forall\epsilon > 0$.*

[Proof of Corollary 5.1] Since $\mathbf{x}_i, \mathbf{x}_j \in Sym_d^+$ and $\epsilon > 0$, $\mathbf{x}_i + \epsilon\mathbf{I}_d$ and $\mathbf{x}_j + \epsilon\mathbf{I}_d \in Sym_d^{+*}$. Let suppose that $\mathbf{X}_i = \mathbf{x}_i + \epsilon\mathbf{I}_d$ and $\mathbf{X}_j = \mathbf{x}_j + \epsilon\mathbf{I}_d$. Therefore, $d_{MLE}(\mathbf{x}_i, \mathbf{x}_j) = ||Log(\mathbf{x}_i + \epsilon\mathbf{I}_d) - Log(\mathbf{x}_j + \epsilon\mathbf{I}_d)||_F = ||Log(\mathbf{X}_i) - Log(\mathbf{X}_j)||_F = d_{LE}(\mathbf{X}_i, \mathbf{X}_j)$ with $\mathbf{X}_i, \mathbf{X}_j \in Sym_d^{+*}$. Thus, $K_G^{SPsD}(\mathbf{x}_i, \mathbf{x}_j) = exp(-\frac{d_{LE}^2(\mathbf{X}_i,\mathbf{X}_j)}{2\sigma^2})$ and based on the Corollary 3.1, the kernel K_G^{SPsD} is positive definite.

6. Using Symmtric Positive semi-Definite matrices as descriptors for action recognition

As mentioned in Section 1, this study aims to make use of covariance matrices as descriptors for action recognition because of their various advantages. In this section, we start by presenting the low-level features composed of Kinematic Features (KF) extracted from normalized skeleton joints. Then, two novel approaches for action recognition are proposed, making use of covariance matrices (which are elements of the space of SPsD matrices), containing Kinematic Features. The first one is static since it does not contain the temporal information, while the second one is dynamic. To classify actions, the extension of kernel methods for Sym_d^+ is used.

6.1. Kinematic Features as low-level features

To ensure the invariance to anthropometric variation, skeletons are first normalized. After that, Kinematic Features representing low-level features are calculated with the use of normalized skeletons.

6.1.1. Skeleton normalization

At each instant t_k, a skeleton pose $\mathbf{P}(t_k)$ corresponds to the 3D position of a set of n joints, as described in Equation (8).

$$\mathbf{P}(t_k) = [\mathbf{p_1}(t_k), ..., \mathbf{p_j}(t_k), ..., \mathbf{p_n}(t_k)] \tag{8}$$

The position of each joint i is denoted by $\mathbf{p_i}(t_k) = [\mathbf{x_i}(t_k), \mathbf{y_i}(t_k), \mathbf{z_i}(t_k)]$.

The initial position of the human hip joint $\mathbf{p_{hip}}$ is assumed to be the origin (Equation (9)).

$$\mathbf{P}(t_k) = [\mathbf{p}_1(t_k) - \mathbf{p}_{\text{hip}}, ..., \mathbf{p}_{\text{j}}(t_k) - \mathbf{p}_{\text{hip}}, ..., \mathbf{p}_{\text{n}}(t_k) - \mathbf{p}_{\text{hip}}] \tag{9}$$

To overcome the anthropometric variability, we propose to follow the same protocol presented in (Ghorbel et al., 2018). All joint positions are normalized except the hip joint position which is assumed to be the root and is therefore unchanged ($\mathbf{p_{hip}^{norm}} = \mathbf{p_{hip}}$). The length of the joint positions are normalized starting with the segments connected with the root of the skeleton (hip joint) and moving gradually to the neighbour segments. Algorithm 1 describes in details the skeleton normalization method.

```
Algorithm 1: Skeleton normalization at an instant $t_k$
Input  : $(\mathbf{p_{a_i}}(t_k), \mathbf{p_{b_i}}(t_k))_{1\leq i\leq C}$ represents the segment
         extremities ordered and $C$ represents the
         number of connections with $a_i$ the root
         extremity and $b_i$ the other extremity of the
         segment $i$
Output: $(\mathbf{p_{a_i}^{norm}(t_k)}, \mathbf{p_{b_i}^{norm}(t_k)})_{1\leq i\leq C}$ with
         $a_i, b_i \in [\![1, n]\!]$
$\mathbf{p_{a_1}^{norm}}(t_k) := \mathbf{p_{a_1}}(t_k)(\mathbf{p_{a_1}}(t_k) = p_{hip}(t_k)$ represents the
  position of the hip joint)
for $i \leftarrow 1$ to $C$ (C:Number of segments) do
    $\mathbf{S_i} := \mathbf{p_{a_i}}(t_k) - \mathbf{p_{b_i}}(t_k)$
    $\mathbf{s_i'} := \frac{\mathbf{S_i}}{||\mathbf{p_{a_i}}(t_k) - \mathbf{p_{b_i}}(t_k)||_2}$
    $\mathbf{p_{b_i}^{norm}}(t_k) := \mathbf{s_i'} + \mathbf{p_{a_i}^{norm}}(t_k)$
end
```

After applying Algorithm 1, we obtain the normalized skeleton $\mathbf{P^{norm}}$ at each instant(t_k), as described by Equation (10).

$$\mathbf{P}^{\text{norm}}(t_k) = [\mathbf{p}_1^{\text{norm}}(t_k), \mathbf{p}_2^{\text{norm}}(t_k), ..., \mathbf{p}_{\text{n}}^{\text{norm}}(t_k)] \tag{10}$$

6.1.2. Kinematic features

As in (Ghorbel et al., 2016), Kinematic Features $\mathbf{KF}(t_k)$ composed of normalized joint positions $\mathbf{P}^{\mathbf{norm}}(t_k)$, joint velocities $\mathbf{V}(t_k)$ as well as joint accelerations $\mathbf{A}(t_k)$ are used as low level-features.

$$\mathbf{KF}(t_k) = [\mathbf{P}^{\mathbf{norm}}(t_k), \mathbf{V}(t_k), \mathbf{A}(t_k)] \tag{11}$$

The velocity and the acceleration at an instant t_k are also calculated as in (Zanfir et al., 2013; Ghorbel et al., 2016).

$$\mathbf{V}(t_k) = \mathbf{P}^{\mathbf{norm}}(k+1) - \mathbf{P}^{\mathbf{norm}}(k-1) \tag{12}$$

$$\mathbf{A}(t_k) = \mathbf{P}^{\mathbf{norm}}(k+2) + \mathbf{P}^{\mathbf{norm}}(k-2) - 2 \times \mathbf{P}^{\mathbf{norm}}(t_k) \tag{13}$$

Thus, the dimension of $\mathbf{KF}(t_k)$, at an instant t_k is equal to $d_1 = 9 \times n$.

6.2. A static approach

In this section, we present the static approach which can be divided into two parts which are the introduction of a novel Kinematic Covariance (KC) descriptor followed by the extension of the RBF-based SVM classification to SPsD matrices. Figure 2 describes these different steps.

6.2.1. Kinematic Covariance Descriptor

As mentioned before, we propose to introduce a novel descriptor making use of statistical and kinematic tools that we call Kinematic Covariance (KC). So, the Kinematic Features (KF) are integrated in a covariance matrix. We suppose that N is the number of frames of the segmented sequence. Thus, the novel Kinematic Covariance (KC) descriptor is computed as described by Equation (14),

$$\mathbf{KC} = \frac{1}{N} \sum_{k=1}^{N} (\mathbf{KF}(t_k) - \nu)(\mathbf{KF}(t_k) - \nu)^T \tag{14}$$

where $\nu = \sum_{k=1}^{N} \mathbf{KF}(t_k)$ is the mean of Kinematic Features. We can deduce that the Kinematic Covariance descriptor $\mathbf{KC}$ represents a square matrix of dimension $d_1 \times d_1$. Since the number of joints is in general equal to $n = 20$ and the number of frames in the majority of datasets is inferior to 100 frames, we can conclude that $d_1 = 180$ is generally superior to the number of frames (which represents the number of samples used to calculate the matrix of covariance). Therefore, in the presence of this condition, covariance matrices are singular and belongs to the space Sym_d^+ and not to the space Sym_d^{+*}. For this reason, the use of a classifier suitable to the classification of SPsD matrices is needed.

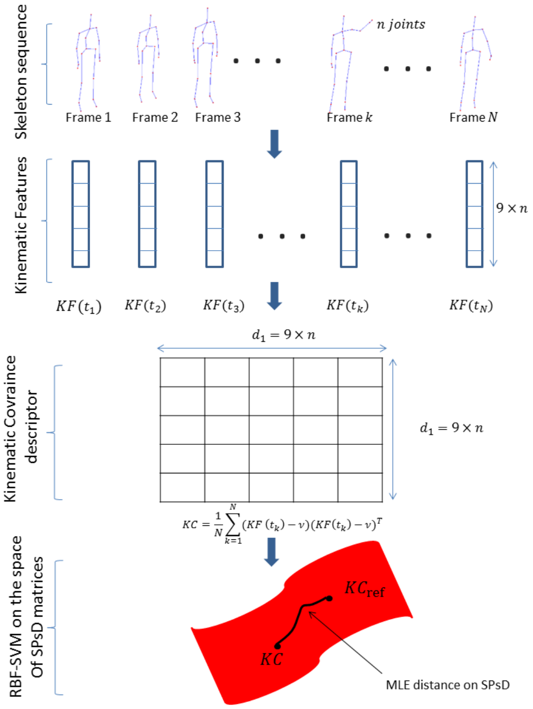

Figure 2: The proposed static approach: For every instant t_i corresponding to the time acquisition of the frame i, Kinematic Features ($KF(t_i)$) are first calculated. Then, the Kinematic Covariance descriptor is computed by integrating the KF as features in the covariance matrix. Finally, an SVM based on RBF kernel is carried out to classify actions in the space of SPsD matrices by using the MLE distance.

6.2.2. Action recognition recognition via the extension of SVM for SPsD matrices

After extracting descriptors, a supervised classification step is necessary to perform the recognition of actions. We propose to use a multi-class Support Vector Machines (SVM) as classifier which is a very popular kernel-based method.

Since covariance matrices are elements of SPsD and classical kernels can not be applied directly, we propose to use the extension of RBF-kernel for SPsD matrices proposed in Section 5. Using the extended kernel K_G^{SPsD}, presented in Section 5, the dual problem becomes:

$$max_\alpha \sum_{i=1}^{N_t} \alpha_i - \frac{1}{2} \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} \alpha_i \alpha_j y_i y_j K_G^{SPsD}(\mathbf{KC}_i, \mathbf{KC}_j) \qquad (15)$$

$$with \sum_{i=1}^{N_t} \alpha_i y_i; \alpha_i \geq 0;$$

$\mathbf{KC}_i$ and $\mathbf{KC}_j$ represents respectively the descriptor of the instance i and the instance j, with y_i, y_j their associated labels (which represents the identifier of the action) , N_t the number of instances used for the training and $\alpha = (\alpha_1, ..., \alpha_i, ..., \alpha_{N_t})$ the Lagrange multipliers.

6.3. A dynamic approach

The limitation of KC descriptor and more generally of covariance descriptors in action recognition is mainly due to the fact that it does not contain the temporal information. Indeed, this kind of representation does not inform about the dynamical evolution over time. Thus, we propose a descriptor called Hierarchical Kinematic Covariance (HKC) descriptor combined with an RBF-SVM based Multiple Kernel Learning approach. More details will be given below.

6.3.1. Hierarchical Kinematic Covariance descriptor

Hussein et al. (2013) proposed to use a covariance matrix, using only the joint position information, as a human action descriptor. Also, they noticed the lack of the temporal evolution. To overcome this limitation, they proposed a hierarchical covariance descriptor. This descriptor contains the concatenation of covariance descriptors calculated on 3 sub-ranges and on the whole range of the sequence.

Inspired by this idea, we propose to follow it by applying it to the KC descriptor.

As in (Hussein et al., 2013), we extract 4 Kinematic covariance matrices $(\mathbf{KC}_i)_{1 \leqslant i \leqslant 4}$ from a skeleton sequence using respectively 3 sub-ranges and the whole range of the skeleton sequence. We call the set of these kinematic covariance matrices, the Hierarchical Kinematic Covariance (HKC) descriptor, as depicted by Equation (17).

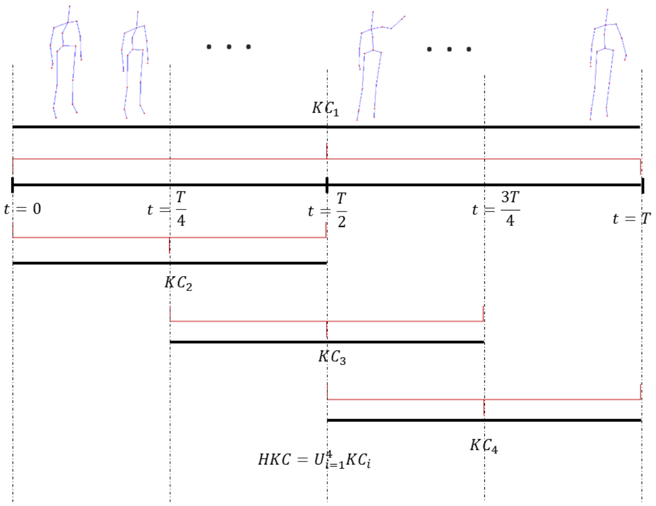

Figure 3: Computation of the Hierarchical Kinematic Covariance (HKC) descriptor as in (Hussein et al., 2013): T corresponds to the skeleton sequence length and KC_1, KC_2, KC_3 and KC_4 are the covariance matrices calculated from each corresponding range.

$$\mathbf{HKC} = \cup_{i=1}^{4} \mathbf{KC}_i \qquad (16)$$

Figure 3 illustrates how are extracted the different ranges from a skeleton sequence. More precisely, the HKC descriptor represents the combination of four KC descriptors. The question that appears now, is how to classify actions using these four matrices simultaneously.

6.3.2. Classification using a Multiple Kernel Learning (MKL) strategy

To realize the classification with the use of the HKC descriptor, we propose to make use of a Multiple Kernel Learning (MKL) strategy. Figure 4 illustrates the proposed approach. The idea of Multiple Kernel Learning is to use more than one kernel for learning as reflected by its name. In this work, we use a linear combination of four different kernels. For each $\mathbf{KC}_i$, an RBF-kernel based on the distance MLE is computed, that we denote by $K_G^{SPsD}(\mathbf{KC}_i)$. Thus, the final kernel used for the learning is computed as presented in Equation (17), where the value μ_i represents the weight attributed to the kernel $K_{\mathbf{KC}_i}^{SPsD}$.

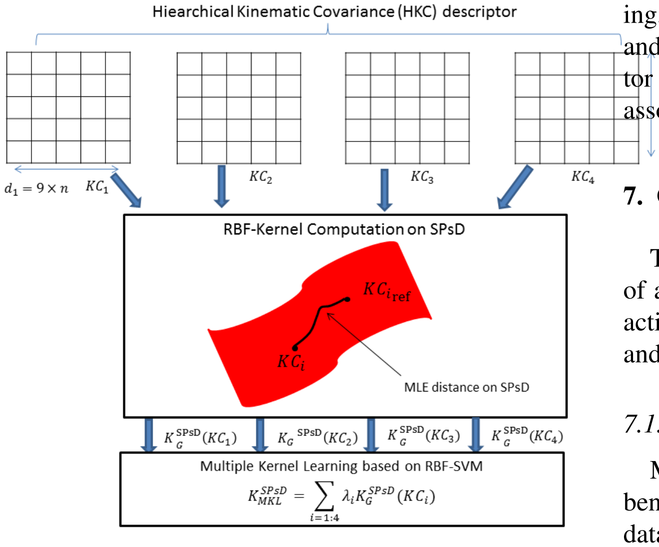
Hiearchical Kinematic Covariance (HKC) descriptor
$d_1 = 9 \times n$
KC_1
KC_2
KC_3
KC_4
$d_1 9 \times n$
RBF-Kernel Computation on SPsD
$KC_{i_{\text{ref}}}$
KC_i
MLE distance on SPsD
$K_G^{\text{SPsD}}(KC_1)$
$K_G{}^{\text{SPsD}}(KC_2)$
$K_G^{\text{SPsD}}(KC_3)$
$K_G^{\text{SPsD}}(KC_4)$
Multiple Kernel Learning based on RBF-SVM
$K_{MKL}^{SPsD} = \sum_{i=1:4} \lambda_i K_G^{SPsD}(KC_i)$

Figure 4: The dynamic proposed approach: For every sub-range i extracted from the whole sequence, a Kinematic Covariance $\mathbf{KC}_i$ is computed. Then, based on every $\mathbf{KC}_i$, an RBF-kernel $K_G^{SPsD}(\mathbf{KC_i})$ using the MLE distance is calculated. To fuse the information of the different kernels, a Multiple Kernel Learning approach is followed using a linear combination given by the kernel denoted by K_{MKL}^{SPsD}. Finally, an SVM model is learned using K_{MKL}^{SPsD} to perform the classification.

$$K_{MKL}^{SPsD} = \sum_{i=1}^{4} \mu_i K_G^{SPsD}(\mathbf{KC}_i) \tag{17}$$

In the last chapter, we have shown that the kernel K_G^{SPsD} is positive definite. Since the linear combination of positive definite matrices is a positive definite matrix, we can conclude that K_{MKL}^{SPsD} is positive definite.

Therefore, an SVM approach can be used for the learning based on the optimization of the following dual problem:

$$max_\alpha \sum_{i=1}^{N_t} \alpha_i - \frac{1}{2} \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} \alpha_i \alpha_j y_i y_j K_{MKL}^{SPsD}(\mathbf{HKC}_i, \mathbf{HKC}_j)$$
$$with \sum_{i=1}^{N_t} \alpha_i y_i; \alpha_i \geq 0; \tag{18}$$

N_t represents the number of instances used for the training, $\alpha = (\alpha_1, ..., \alpha_i, ..., \alpha_{N_t})$ are the Lagrangian multipliers and $\mathbf{HKC}_i$ and $\mathbf{HKC}_j$ represents respectively the descriptor of the instance i and the instance j, with y_i, y_j their associated labels.

7. Offline experiments

To validate our approach, we propose to test it (in terms of accuracy and latency) on three benchmarks of human actions, namely MSRAction3D dataset, UTkinect dataset and Multiview3Ddataset.

7.1. Datasets

MSRAction3D dataset (Li et al., 2010) is a well-known benchmark used in the field of 3D action recognition. This dataset is composed of two modalities provided by RGB-D cameras: depth maps and skeleton sequences. There are 20 types of actions where subjects are facing the camera (there is no viewpoint variation). Each action is performed by 20 different actors 2 or 3 times. This dataset is very challenging because of the similarity of many of its actions.

UTKinect dataset represents a well-know benchmark for RGB-D based action recognition. It contains three modalities: RGB images, depth maps and skeleton sequences. This dataset contains 10 actions realized by 10 different subjects 2 times.

Multiview3D dataset is a recently introduced dataset in (Hammouche et al., 2016) for 3D action recognition. The particularity of this dataset is that it contains actions performed with different body orientations. Multiview3D dataset contains 12 actions performed by 30 different subjects in three different orientations ($-30\,°$, $0\,°$, $30\,°$). Thus, this dataset is very challenging because of its important body orientation variability.

7.2. Criteria of evaluation

Generally, only the accuracy of recognition is reported. This evaluation criterion is informative, but not sufficient. As presented in (Ghorbel et al., 2015), we also report the Mean Execution Time (MET) per descriptor. The MET is computed by averaging the execution time necessary to extract a descriptor from an instance. In other words, the execution time required to calculate descriptors on a whole dataset is divided by the number of instances contained in this dataset. To obtain a meaningful comparison, this criterion is measured for all descriptors with the respect of the same experimental conditions.

7.3. Experimental settings

Because the experimental parameters and settings vary from a paper to another and because the MET is rarely reported, we recover available codes of recent RGB-D methods provided by their authors and run them with the respect of the same settings to realize a fair comparison. In total, we test on our machines 7 different descriptors which can be divided into 2 groups: depth-based descriptors and skeleton-based descriptors. The depth-based descriptor group contains the following descriptors: Square Histogram of Oriented Gradient (HOG2) (Ohn-Bar and Trivedi, 2013), Histogram of 4D normals (HON4D)(Oreifej and Liu, 2013) and Super Normal Vectors (SNV) (Yang and Tian, 2014). On the other hand, the group of skeleton-based descriptors group gathers Joint Positions (JP) (Vemulapalli et al., 2014), Relative Joint Positions (RJP) (Vemulapalli et al., 2014), Joint Angles (JA) (Vemulapalli et al., 2014) and finally Lie Algebra Relative Pairwise representation (Vemulapalli et al., 2014).

All the tested codes are run on the same laptop, a Dell Inspiron N5010 laptop computer with Intel Core i5 processor, Windows 7 operating system and 4GB RAM.

For MSRAction3D dataset, we follow the procedure proposed in (Li et al., 2010) where MSRAction3D dataset is divided in three sets according to action labels: AS1, AS2 and AS3 (Li et al., 2010). The training and prediction steps are separately done in each set and an accuracy score is obtained for each one. The final accuracy recognition score represents the average percentage of recognition on the three sets. As in (Yang and Tian, 2014), data generated by subjects 1,3,5,7,9 are used for training and data generated by subjects 2,4,6,8,10 are used for testing.

For UTKinect dataset, we follow the same experimental protocol proposed in (Vemulapalli et al., 2014), where the data collected, thanks to the subjects 1,3,5,7,9 are used for the training while the rest of the data is used for testing. This dataset is very challenging because of the important intra-class variability.

For Multiview3D dataset, the same experimental protocol proposed in (Hammouche et al., 2016) is followed. A cross-splitting is done to separate the data in two sets: training data and testing data. Then, the training is done by using data with a specific orientation and the testing is carried out by considering data with another orientation. This process is repeated several times for all possible combinations. Finally, we calculate two values: 1) the average accuracy when the orientation of training and testing data is the same. This test is called Same View (SV) test. 2) the average accuracy when the orientation of training and testing data is different. We call this test Different View test (DV). These two values are a way to analyze how much methods are affected by view-point variation.

Skeleton alignment for Multiview 3D dataset: Finally, since Multiview3D dataset contains skeletons with variable orientations, we propose to add in the pre-processing step, a simple skeleton alignment algorithm. We align the data as follows: we consider that for each action the first skeleton of the sequence is in the rest state. To do that, we assume that we work in a specific scenario where the actions are already segmented and where each first skeleton is in the rest state. We choose one of the first skeletons as a reference and we optimize the transformation matrix between the first pose of each sequence and the reference skeleton using a least square optimization. Hence, we apply the obtained transformation matrix to the rest of the sequence.

7.4. Results and discussion

7.4.1. A trade-off between computational latency and recognition accuracy

Table 1, Table 2 and Table 3 report respectively the results of both proposed methods (KC+SVM-MLE and HKC+MKL-MLE) by comparing them to state-of-the-art methods in terms of MET per descriptor and accuracy on MSRAction3D, UTKinect and Multiview3D datsets. According to the experimentation conducted on these three datasets, we can conclude that our descriptor (HKC+SVM-MLE) is accurate (with an accuracy of 92.35% on MSRAction3D, 94.94% on UTKinect and 96.17%(SV)-93.40%(DV) on Multiview3D) and is also fast to compute (with an MET of 0.044 per descriptor on MSRAction3D, 0.033 per descriptor on UTKinect and 0.035 per descriptor on Multiview3D).

The majority of descriptors which exceeds in terms of accuracy our descriptor such as LARP on UTKinect and

Descriptor	AS1(%)	AS2(%)	AS3(%)	Overall(%)	M.E.T(s)
HOG2 (Ohn-Bar and Trivedi, 2013)	90.47	84.82	**98.20**	91.16	**6.44**
HON4D (Oreifej and Liu, 2013)	94.28	91.71	**98.20**	94.47	27.33
SNV (Yang and Tian, 2014)	**95.25**	**94.69**	96.43	**95.46**	146.57
JP (Vemulapalli et al., 2014)	82.86	68.75	83.73	78.44	0.58
RJP (Vemulapalli et al., 2014)	81.90	71.43	88.29	80.53	2.15
Q (Vemulapalli et al., 2014)	66.67	59.82	71.48	67.99	1.33
LARP (Vemulapalli et al., 2014)	83.81	84.82	92.73	87.14	17.61
KSC (Ghorbel et al., 2016)	83.81	87.5	97.3	89.54	0.092
KC+SVM-MLE (ours)	88.57	83.04	92.79	88.133	**0.043**
HKC+MKL-MLE (ours)	**91.42**	**92.85**	**92.79**	**92.35**	0.044

Table 1: Accuracy of recognition (%) and Mean Execution Time per descriptor (MET) in seconds on MSRAction3D: AS1, AS2 and AS3 represent the three groups proposed in the protocol experimentation of (Li et al., 2010)

Descriptor	Accuracy (%)	M.E.T (s)
HOG2 (Ohn-Bar and Trivedi, 2013)	74.15	5.025
SNV (Yang and Tian, 2014)	79.80	1365.33
HON4D (Oreifej and Liu, 2013)	90.92	25.33
Random Forest* (Zhu et al., 2013)	87.90	-
LARP (Vemulapalli et al., 2014)	**97.08**	42.00
KSC (Ghorbel et al., 2016)	96.00	0.082
KC+SVM-MLE (ours)	90.91	**0.032**
HKC+MKL-MLE (ours)	94.95	**0.033**

Table 2: Accuracy of recognition (%) and M.E.T (s) on UTKinect dataset. *The results of Random Forest have been recovered from (Zhu et al., 2013) because the code is not available.

SNV on MSRAction3D requires a more important computational time. For example, LARP needs an MET of 17.61s per descriptor on MSRAction3D dataset, 42.00 s per descriptor on UTKinect dataset and 10.51s per descriptor on Multiview3D dataset. This is maybe due to the high number of approximation and calculation required by this method.

The first descriptors HOG2 (Ohn-Bar and Trivedi, 2013), HON4D (Oreifej and Liu, 2013) and SNV (Yang and Tian, 2014) which represent depth-based descriptors are very accurate according to the recognition accuracy results on MSRAction3D. However, as remarked in (Ghorbel et al., 2015), they are greedy in terms of computational time since the dimension of depth images is more important on the three datasets. Furthermore, HOG2 (Ohn-Bar and Trivedi, 2013) and SNV (Yang and Tian, 2014) gives very low accuracy on UTKinect dataset. That could be explained by the fact that UTKinect dataset contains some videos with a very small number of frames.

It can be noted that compared to skeleton representations, the Hierarchical Kinematic Covariance (HKC) descriptor combined with an MLE-distance based MKL presents the most accurate recognition on MSRAction3D and Multiview3D dataset . Moreover, it presents one of the lowest mean execution time per descriptor with 0.044s per descriptor (after Kinematic Covariance descriptor with 0.043s per descriptor). Even if our method does not present the best results in terms of accuracy on UTKinect dataset with 94.95 % (against 97.08% for LARP and 95% for KSC), it remains accurate and its low computational latency with a mean execution time per descriptor which is equal to 0.033s represents a very motivating result, since it realizes a trade-off between latency and accuracy. Both KSC and HKC presents interesting results in terms of computational latency and accuracy. Nevertheless, KSC remains hardly applicable to online scenarios, in opposition to HKC, as presented in Section 8. In fact, the KSC descriptor needs the a priori knowledge of the whole video in order to calculate the whole kinetic energy (used to normalize temporally the actions).

To illustrate simultaneously the information of accuracy and MET per descriptor, we propose the representation of the results in Figure 5. Every ball corresponds to a method making use of a specific descriptor. The surface area of the ball represents the MET, while the center of the ball corresponds to the recognition accuracy. It is easy to notice that our method (HKC+MKL-MLE) presents one of the best trade-off between latency and accuracy on MSRAction3D.

Although the use of the skeleton needs pre-processing, skeleton modality remains more suited to fast recognition.

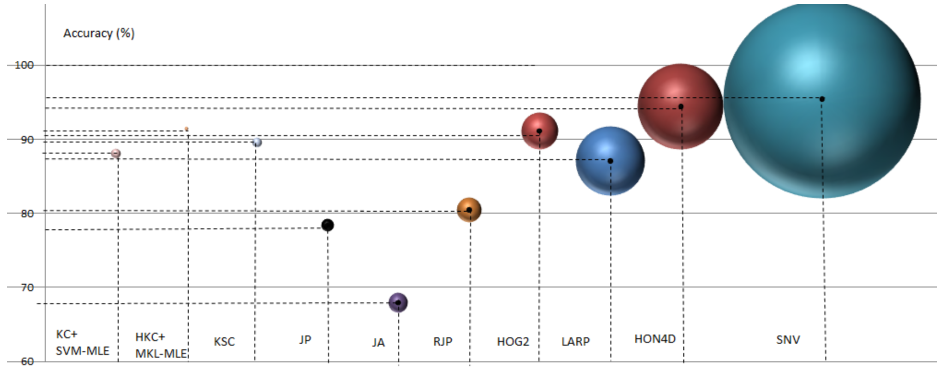

Figure 5: Illustration of the recognition accuracy and the MET of KC , HKC descriptors and state-of-the-art-descriptors: As described in previous chapters, the center of the balls represents the accuracy of every method and the surface area its MET per descriptor

Descriptor	SV (%)	DV (%)	MET (s)
HOG2 (Ohn-Bar and Trivedi, 2013)	87.8	74.2	9.06
HON4D (Oreifej and Liu, 2013)	89.3	76.6	17.51
SNV (Yang and Tian, 2014)	94.27	76.65	271.3
Actionlet* (Wang et al., 2012a)	87.1	69.7	0.139
LARP (Vemulapalli et al., 2014)	96.00	88.1	10.51
KSC (Ghorbel et al., 2016)	90.45	90.10	0.099
KC+SVM-MLE (ours)	80.72	75.17	**0.033**
HKC+MKL-MLE (ours)	**96.17**	**93.40**	**0.035**

Table 3: Accuracy of recognition using SV and DV tests (%) and M.E.T (s) on Multiview3D. *The results of Actionlet have been recovered from (Hammouche et al., 2016) because the code is not available.

orientation	0°	30°	−30°
0°	98.96	90.63	90.63
30°	94.79	89.58	88.54
-30°	95.83	88.54	92.71

orientation	0°	30°	−30°
0	98.96	96.88	94.79
30	96.88	97.92	96.88
-30°	95.83	90.63	98.96

Table 4: Accuracy of every test on Multiview3D dataset using HKC+MKL-MLE approach: We detail here the accuracy obtained for every test. The table on the left represents the results obtained when the training data are performed by subjects 1,2,3 and 4. The table on the right represents the results obtained when the training data are performed by subjects 5,6,7 and 8. The orientation specified in the columns represents the orientation of the data used for training, while the orientation specified in the lines represents the orientation of the data used for testing

Indeed, according to (Papadopoulos et al., 2014), skeleton extraction process takes around 45ms per frame. For an action of 30 frames (very reasonable length for an action), the necessary time to extract a skeleton sequence is equal to nearly 1,35 s.

7.4.2. Robustness to viewpoint changes

Table 3 which reports the obtained results on Multiview3D dataset shows the robustness of our method (HKC+MKL) to viewpoint changes compared to other methods. Although KSC looks less sensitive to viewpoint variation, it can be noted that the proposed approach presents the best results in terms of accuracy for both Same View (SV) and Different Views (DV) tests. Indeed, HKC combined with MKL-MLE gives 96.17% for data with Same View (SV) and 93.40% for data with Different Views (DV). Moreover, the differences between the accuracies obtained for SV and DV tests are the lowest one after KSC for our method (less than 3% of differences for HKC+MKL).

Table 4 details the different SV and DV tests. It can be noted that even if SV tests present a global better accuracy, the accuracy registred for the different SV and DV tests belong to the same range of values (between 88.54% to 98.96%).

7.4.3. Observational latency

The observational latency is also an important criterion for online action recognition. For this reason, we propose

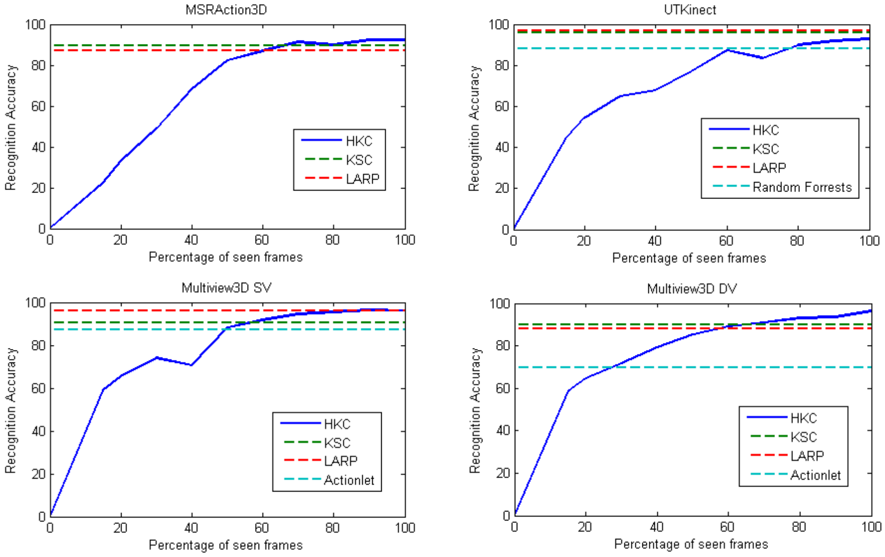

Figure 6: Accuracy of recognition on different datasets according to the percentage of seen frames: The dotted lines mean that the reported accuracy is only done using the whole sequence.

to carry out the following experiments.

The idea is to report the accuracy using only a specific percentage of seen frames as proposed in (Zanfir et al., 2013). Figure 6 illustrates the recognition accuracy according the percentage of observed frames on MSRAction3D, on UTKinect, on Multiview3D dataset for SV tests and Multiview3D dataset for DV tests.

For MSRAction3D dataset, it can be noted that starting from 50% of seen frames, the accuracy exceeds 80% and that starting from 70% of seen frames, our descriptor registers a better score than KSC and LARP. Also, on UTKinect, after 50% of observed frames, the score is superior to 80% and after 60% of seen frames, the accuracy is more or less stable. In the same way, the accuracy is more or less stable starting from 50% of seen frames on Multiview3D dataset for both SV and DV tests.

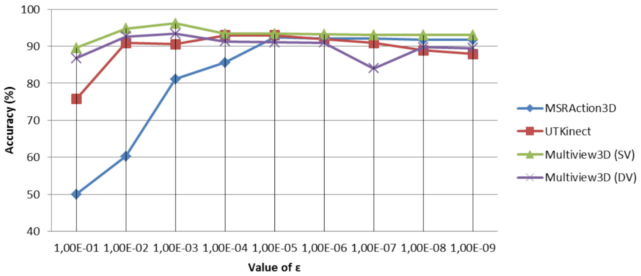

Figure 7: Threshold influence on the accuracy of three benchmarks

7.4.4. Parameter Analysis

The parameter ϵ. As mentioned previously, the parameter ϵ has to be fixed. According to our previous analysis in Section 4.2, it is preferable that ϵ remains smaller than the covariance eigenvalues. In practical, the problem is that if we choose ϵ too small, it is considered null. To study the effect of varying ϵ, we propose the illustration of Figure 7, which reports the accuracy according the chosen value for ϵ. The graph shows that by choosing $\epsilon \leq 10^{-5}$, the accuracy does not vary importantly. However, the highest accuracy has been respectively registered for $\epsilon_{MSR} = 10^{-5}$, $\epsilon_{UT} = 10^{-4}$ and $\epsilon_{UT} = 10^{-3}$ on MSRAction3D, UTKinect and Multiview3D datasets.

The parameters μ_i We recall that these parameters weight the contribution of each kernel and weight therefore the contribution of each covariance matrix. In all experiments, we empirically fix the best combination $\mu = (\mu_1, \mu_2, \mu_3, \mu_4)$ providing the best accuracy by performing several tests. Nevertheless, the experiments have shown that even with varying μ randomly the accuracy does not decrease significantly, with a maximum of 5%.

Descriptor	Latency (%)	Miss rate (%)	Error rate (%)
(Hussein et al., 2013)	52	20.7	91.7
(Kviatkovsky et al., 2014)	41.3	15.2	54.1
(Tang et al., 2018)	29	**9.4**	51.6
HKC (ours)	**9.69**	13.44	**45.19**

Table 5: Latency, Miss rate and Error rate on the dataset MSRC12

8. Online experiments

In this part, we propose to extend our descriptor HKC to online action recognition by using a simple sliding window. To detect action, the probabilities P_i^t of belonging to a class i provided by the SVM algorithm at each instant t are used. The standard deviation s of these probabilities are computed and a threshold T_1 is empirically fixed. If $s > T_1$, the action with the higher probability is detected, if not, we assume that the window does not contain any action. As in (Tang et al., 2018), the size of the window is fixed to 30 frames. T_1 is empirically fixed to 0.15. The experiments are realized on the dataset MSRC12 (Fothergill et al., 2012).

8.1. MSRC12 dataset

This dataset is generally used for online skeleton-based action recognition. It contains 594 sequences including 12 different types of gestures performed by 30 subjects. In total, there are 6244 gesture instances. MSRC12 only provides skeleton joints and the action points.

8.2. Experimental settings

To evaluate our method in a online mode, the criteria of evaluation proposed in (Tang et al., 2018) have been reported: the latency, the miss rate as well as the error rate. More details about the calculation of these values can be found in (Tang et al., 2018). For this experimentation, we also follow exactly the same settings proposed in (Tang et al., 2018): the dataset has been divided in 5 parts and a leave-one-out protocol has been used for each part.

8.3. Results and discussion

In Table 5, the results obtained on the dataset MSRC12 are reported. While our method presents a lower latency and error rate than other approaches, it registers a slightly higher miss rate score than the method of (Tang et al., 2018). This means that our method is able to recognize an action by using less information and presents less frame classification errors. However, it also means that compared to (Tang et al., 2018) the actions are sometimes not detected. This could be caused by the detection algorithm that could be improved in the future by integrating an automatic way to fix the threshold T_1.

Compared to (Tang et al., 2018), the superiority of our approach can be explained by three main facts:

a) The use of the hierarchy: In (Tang et al., 2018), the authors are just computing a simple covariance matrix. Therefore, the temporal information is not encoded in the descriptor. This can affect the results due to the confusion of motions with similar spatial distribution. For example, the actions ”standing” and ”sitting” which have similar spatial distribution but a different temporal ordering can be confused. Thanks to the inclusion of the temporal hierarchy, our descriptor (HKC) is able to take into account the temporal evolution.

b) The use of a more sophisticated classifier: Based on the presented analysis of the perturbed Log-Euclidean distance, we were able to extend kernel based classification step which have probably contributed to the improvement of the results by using an MKL approach. Furthermore, in (Tang et al., 2018), authors make use of the information of the distance without integrating it in a classifier. Since we use a machine learning technique, there is no need to compute a distance between the instance to be recognized and all the other instances as in (Tang et al., 2018). This leads to a system able to take a faster decision.

c) A more complete description of the action: As shown in (Zanfir et al., 2013; Ghorbel et al., 2016), the inclusion of kinematic values further to the position such as the velocity and the acceleration of joints boosts the results.

9. Conclusion and perspectives

In this paper, an extension of RBF-kernel methods is proposed in order to apply it to the case of action recognition. To recognize actions, a Hierarchical Covariance descriptor (HKC) is used because of its good properties: this descriptor is accurate, fast to compute and can be extended to online recognition, etc. Since HKC represents a combination of covariance matrices which belong to the space of SPsD matrices, we propose to apply a perturbation to the Log-Euclidean distance. The validity of this perturbation is also studied in this paper. Therefore, the RBF kernel is extended to SPsD matrices by using the MLE distance. We use therefore an MLE-MKL approach to recognize actions described by HKC descriptors. A fair comparative experimentation by recovering available state-of-the-art methods have shown the efficiency of our approach in the case of action recognition. Our descriptor has given good results in terms of accuracy and computational latency on three different datasets. We have also proposed to adopt it for online recognition by making use of a sliding window.

However, some improvements can be done. The parameters ϵ and μ_i have been fixed in an empirical way. Proposing a more formal way to choose these parameters is an interesting track to explore. On the other hand, as shown in (Kviatkovsky et al., 2014), there is an incremental relation between a covariance descriptor at an instant t and an instant $t + 1$. In this way, we propose to exploit the incremental covariance calculation as in (Kviatkovsky et al., 2014; Tang et al., 2018) in a future work. Then, we propose also to extend MLE-SVM and MLE-MKL by making them incremental/decremental in order to include novelty as in (Boukharouba et al., 2009).

References

Amor, B.B., Su, J., Srivastava, A., 2016. Action recognition using rate-invariant analysis of skeletal shape trajectories. IEEE transactions on Pattern Analysis and Machine Intelligence 38, 1–13.

Arsigny, V., Fillard, P., Pennec, X., Ayache, N., 2006. Log-euclidean metrics for fast and simple calculus on diffusion tensors. Magnetic resonance in medicine 56, 411–421.

Arsigny, V., Fillard, P., Pennec, X., Ayache, N., 2007. Geometric means in a novel vector space structure on symmetric positive-definite matrices. SIAM journal on matrix analysis and applications 29, 328–347.

Bonnabel, S., Sepulchre, R., 2009. Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank. SIAM Journal on Matrix Analysis and Applications 31, 1055–1070.

Boukharouba, K., Bako, L., Lecoeuche, S., 2009. Incremental and decremental multi-category classification by support vector machines, in: International Conference on Machine Learning and Applications, pp. 294–300.

Brun, L., Percannella, G., Saggese, A., Vento, M., 2016. Action recognition by using kernels on aclets sequences. Computer Vision and Image Understanding 144, 3–13.

Evangelidis, G., Singh, G., Horaud, R., 2014. Skeletal quads: Human action recognition using joint quadruples, in: IEEE International Conference on Pattern Recognition.

Förstner, W., Moonen, B., 2003. A metric for covariance matrices, in: Geodesy-The Challenge of the 3rd Millennium, pp. 299–309.

Fothergill, S., Mentis, H., Kohli, P., Nowozin, S., 2012. Instructing people for training gestural interactive systems, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1737–1746.

Ghorbel, E., Boutteau, R., Boonaert, J., Savatier, X., Lecoeuche, S., 2015. 3D real-time human action recognition using a spline interpolation approach, in: International Conference on Image Processing Theory Tools and Applications, IEEE.

Ghorbel, E., Boutteau, R., Boonaert, J., Savatier, X., Lecoeuche, S., 2016. A fast and accurate motion descriptor for human action recognition applications, in: IEEE International Conference on Pattern Recognition.

Ghorbel, E., Boutteau, R., Boonaert, J., Savatier, X., Lecoeuche, S., 2018. Kinematic spline curves: A temporal invariant descriptor for fast action recognition. Image and Vision Computing 77, 60–71.

Hammouche, M., Ghorbel, E., Fleury, A., Ambellouis, S., 2016. Toward a real time view-invariant 3D action recognition. International Conference on Computer Vision Theory and Applications , 745–754.

Hou, Y., Li, Z., Wang, P., Li, W., 2016. Skeleton optical spectra based action recognition using convolutional neural networks. IEEE Transactions on Circuits and Systems for Video Technology .

Hussein, M.E., Torki, M., Gowayyed, M.A., El-Saban, M., 2013. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations., in: International Joint Conferences on Artificial Intelligence, pp. 2466–2472.

Jayasumana, S., Hartley, R., Salzmann, M., Li, H., Harandi, M., 2013. Kernel methods on the riemannian manifold of symmetric positive definite matrices, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 73–80.

Johansson, G., 1973. Visual perception of biological motion and a model for its analysis. Attention, Perception, & Psychophysics 14, 201–211.

Klingenberg, C.P., 2013. Cranial integration and modularity: insights into evolution and development from morphometric data. Hystrix, the Italian Journal of Mammalogy 24, 43–58.

Kviatkovsky, I., Rivlin, E., Shimshoni, I., 2014. Online action recognition using covariance of shape and motion. Computer Vision and Image Understanding 129, 15–26.

Li, W., Zhang, Z., Liu, Z., 2010. Action recognition based on a bag of 3D points, in: IEEE Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 9–14.

Liu, Z., Zhang, C., Tian, Y., 2016. 3D-based deep convolutional neural network for action recognition with depth sequences. Image and Vision Computing 55, 93–100.

Ohn-Bar, E., Trivedi, M.M., 2013. Joint angles similarities and HOG2 for action recognition, in: IEEE Computer Vision and Pattern Recognition Workshops, pp. 465–470.

Oreifej, O., Liu, Z., 2013. Hon4d: Histogram of oriented 4D normals for activity recognition from depth sequences, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 716–723.

Pang, Y., Yuan, Y., Li, X., 2008. Gabor-based region covariance matrices for face recognition. IEEE Transactions on Circuits and Systems for Video Technology 18, 989–993.

Papadopoulos, G.T., Axenopoulos, A., Daras, P., 2014. Real-time skeleton-tracking-based human action recognition using kinect data, in: MultiMedia Modeling, pp. 473–483.

Poppe, R., 2010. A survey on vision-based human action recognition. Image and vision computing 28, 976–990.

Qiao, R., Liu, L., Shen, C., van den Hengel, A., 2017. Learning discriminative trajectorylet detector sets for accurate skeleton-based action recognition. Pattern Recognition 66, 202–212.

Rahmani, H., Mahmood, A., Huynh, D., Mian, A., 2016. Histogram of oriented principal components for cross-view action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 38, 2430–2443.

Schoenberg, I.J., 1938. Metric spaces and positive definite functions. Transactions of the American Mathematical Society 44, 522–536.

Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R., 2013. Real-time human pose recognition in parts from single depth images. Communications of the ACM 56, 116–124.

Sra, S., 2011. Positive definite matrices and the symmetric Stein divergence. Technical Report.

Tang, C., Li, W., Wang, P., Wang, L., 2018. Online human action recognition based on incremental learning of weighted covariance descriptors. Information Sciences .

Tuzel, O., Porikli, F., Meer, P., 2006. Region covariance: A fast descriptor for detection and classification, in: European Conference on Computer Vision, pp. 589–600.

Vemulapalli, R., Arrate, F., Chellappa, R., 2014. Human action recognition by representing 3D skeletons as points in a Lie group, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 588–595.

Wang, J., Liu, Z., Wu, Y., Yuan, J., 2012a. Mining actionlet ensemble for action recognition with depth cameras, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1290–1297.

Wang, P., Li, W., Gao, Z., Zhang, J., Tang, C., Ogunbona, P.O., 2016. Action recognition from depth maps using deep convolutional neural networks. IEEE Transactions on Human-Machine Systems 46, 498–509.

Wang, R., Guo, H., Davis, L.S., Dai, Q., 2012b. Covariance discriminative learning: A natural and efficient approach to image set classification, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2496–2503.

Wang, R., Guo, H., Davis, L.S., Dai, Q., 2012c. Covariance discriminative learning: A natural and efficient approach to image set classification, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2496–2503.

Weinland, D., Ronfard, R., Boyer, E., 2011. A survey of vision-based methods for action representation, segmentation and recognition. Computer Vision and Image Understanding 115, 224–241.

Xia, L., Chen, C.C., Aggarwal, J., 2012. View invariant human action recognition using histograms of 3D joints, in: IEEE Computer Vision and Pattern Recognition Workshops, pp. 20–27.

Yang, X., Tian, Y., 2012. Eigenjoints-based action recognition using naive-bayes-nearest-neighbor, in: IEEE Computer Vision and Pattern Recognition Workshops, IEEE. pp. 14–19.

Yang, X., Tian, Y., 2014. Super normal vector for activity recognition using depth sequences, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 804–811.

Zanfir, M., Leordeanu, M., Sminchisescu, C., 2013. The moving pose: An efficient 3D kinematics descriptor for low-latency action recognition and detection, in: IEEE International Conference on Computer Vision, pp. 2752–2759.

Zhu, F., Shao, L., Xie, J., Fang, Y., 2016. From handcrafted to learned representations for human action recognition: a survey. Image and Vision Computing 55, 42–52.

Zhu, Y., Chen, W., Guo, G., 2013. Fusing spatiotemporal features and joints for 3D action recognition, in: IEEE Computer Vision and Pattern Recognition Workshops, pp. 486–491.