

Fitting, Comparison, and Alignment of Trajectories on Positive Semi-Definite Matrices with Application to Action Recognition

Benjamin Szczapa¹, Mohamed Daoudi¹, Stefano Berretti², Alberto Del Bimbo², Pietro Pala²
and Estelle Massart³

¹ IMT Lille Douai, Univ. Lille, CNRS, UMR 9189 CRIStAL, F-59000 Lille, France

² Department of Information Engineering, University of Florence, Italy

³ ICTEAM Institute, UCLouvain, Avenue Georges Lemaître 4 bte L4.05.01, 1348 Louvain-la-Neuve, Belgium

Abstract

In this paper, we tackle the problem of action recognition using body skeletons extracted from video sequences. Our approach lies in the continuity of recent works representing video frames by Gramian matrices that describe a trajectory on the Riemannian manifold of positive-semidefinite matrices of fixed rank. Compared to previous work, the manifold of fixed-rank positive-semidefinite matrices is endowed with a different metric, and we resort to different algorithms for the curve fitting and temporal alignment steps. We evaluated our approach on three publicly available datasets (UTKinect-Action3D, KTH-Action and UAV-Gesture). The results of the proposed approach are competitive with respect to state-of-the-art methods, while only involving body skeletons.

1. Introduction

In the last decades, automatic analysis of human motion has been an active research topic, with applications that have been exploited in a number of different contexts, including video surveillance, semantic annotation of videos, entertainment, human computer interaction and home care rehabilitation, to say a few. Differences in body proportion (size, height, corpulence), body stiffness and training, influence the way different people perform an action. Even one same person is not able to perform the same action twice, exactly replicating the same sequence of body poses in space and time. This variability makes the task of human motion analysis very challenging.

For years, the approaches could be distinguished in two main classes: those operating on pixel values extracted from the RGB stream (either stacking groups of consecutive frames or extracting motion vectors) and those building upon the higher level representation of body skeletons. These latter approaches were supported by the dif-

fusion of low-cost RGB-D cameras (such as the Microsoft Kinect) that can operate in real-time, while reliably extracting the 3D coordinates of body joints. More recently, deep CNN architectures have demonstrated real-time and accurate extraction of the coordinates of body joints from RGB streams (7). These advances make it possible to use a skeleton-based body representation in a much broader range of domains and operative contexts than before, being not limited by the short operative range of RGB-D sensors that typically operate indoor and in the range of a few meters. The design of the recognition/classification module on top of the body skeleton representation makes it possible to describe an action as a sequence of body poses, each one corresponding to a point in a feature space, whose dimension is proportional to the number of body joints. By exploiting the geometric properties of the manifold where these pose descriptors lie, it is possible to define a similarity metric that is invariant under translation, scaling, rotation and also under variations of the speed of execution of the action. Furthermore, the explicit representation of an action as a trajectory, *i.e.*, a sequence of poses, on the manifold makes it possible to extract statistical summaries, such as mean and deviation from the mean, from a group of actions. Through these summaries, one action can be better characterized for the purpose of detecting outliers corresponding to the anomalous execution of an action, that can be of particular relevance for action prediction. In fact, when analyzing the skeleton sequences, there are four main aspects to challenge: (1) A shape representation invariant to undesirable transformations; (2) A temporal modeling of landmark sequences; (3) A suitable rate-invariant distance between arbitrary sequences, and (4) A solution for temporal sequence classification.

This paper lies in the continuity of recent works that model the comparison and classification of temporal sequences of landmarks on the Riemannian manifold of positive-semidefinite matrices. Building on the work (18), our approach involves four different steps: 1) We build a

trajectory on the Riemannian manifold from the body skeletons; 2) We apply a curve fitting algorithm on the trajectories to denoise the data points; 3) We perform a temporal alignment using a Global Alignment Kernel, defining a positive-semidefinite kernel; 4) Finally, we use this kernel with a classic SVM to classify the actions. An overview of the full approach is given in Fig. 1.

The novelties with respect to (18) are:

- The manifold of positive-semidefinite matrices is here endowed with a different metric;
- A recent curve fitting method is used to smooth trajectories on the manifold;
- We use Global Alignment Kernel for temporal alignment, instead of Dynamic Time Warping.

The rest of the paper is organized as follows: in Section 2, we summarize the closest works in the literature. In Section 3, we explain how we represent our data on the manifold of fixed-rank positive-semidefinite matrices, and present the Riemannian metric that we will use in this paper. In Section 4, we describe the curve fitting algorithm that we will use to denoise the trajectories. The curve alignment method and the classifier are presented in Section 5, while results and discussions are reported in Section 6. Finally, in Section 7, we conclude and draw some perspectives of the work.

2. Related Work

A detailed review of the many approaches to human action recognition and classification is out of the scope of this paper. The interested reader can refer to (21) for a detailed and updated survey. In the following, we focus on approaches that use body skeletons as inputs to the recognition/classification module.

One of the first approaches to perform action recognition by the analysis of trajectories of tracked body interest points was presented in Matikainen *et al.* (31). Despite the promising results obtained, the authors did not take into account the geometric information of the trajectories. More recently, in the case of human skeleton in RGB-D images, Devanne *et al.* (9) proposed to formulate the action recognition task as the problem of computing a distance between trajectories generated by the joints moving during the action. An action is then interpreted as a normalized parameterized curve in \mathbb{R}^N . However, this approach does not take into account the relationship between the joints. In the same direction, Su *et al.* (38) proposed a metric that considers time-warping on a Riemannian manifold, thus allowing the registration of trajectories and the computation of statistics on the trajectories. Su *et al.* (39) applied this framework to the problem of visual speech recognition. Similar ideas have been developed by Ben Amor *et al.* (3) on the Kendall’s shape space

with application to action recognition using rate-invariant analysis of skeletal shape trajectories.

Anirudh *et al.* (2) started from the framework of Transported Square-Root Velocity Fields (TSRVF), which has desirable properties including a rate-invariant metric and vector space representation. Based on this framework, they proposed to learn an embedding such that each action trajectory is mapped to a single point in a low-dimensional Euclidean space, and the trajectories that differ only in the temporal rate map to the same point. The TSRVF representation and accompanying statistical summaries of Riemannian trajectories are used to extend existing coding methods such as PCA, KSVD, and Label Consistent KSVD to Riemannian trajectories. In the experiments, it is demonstrated that such coding efficiently captures distinguishing features of the trajectories, enabling action recognition, stroke rehabilitation, visual speech recognition, clustering, and diverse sequence sampling.

In (44), Vemulapalli *et al.* proposed a Lie group trajectory representation of the skeletal data on a product space of special Euclidean (SE) groups. For each frame, this representation is obtained by computing the Euclidean transformation matrices encoding rotations and translations between different joint pairs. The temporal evolution of these matrices is seen as a trajectory on $SE(3) \times \dots \times SE(3)$ and mapped to the tangent space of a reference point. A one-versus-all SVM, combined with Dynamic Time Warping and Fourier Temporal Pyramid (FTP) is used for classification. One limitation of this method is that mapping trajectories to a common tangent space using the logarithm map could result in significant approximation errors. Aware of this limitation, in (45) the same authors proposed a mapping combining the usual logarithm map with a rolling map that guarantees a better flattening of trajectories on Lie groups.

More recently, Kacem *et al.* (18) proposed a geometric approach for modeling and classifying dynamic 2D and 3D landmark sequences based on Gramian matrices derived from the static landmarks. This results in an affine-invariant representation of the data. Since Gramian matrices are positive-semidefinite, the authors relies on the geometry of the manifold of fixed-rank positive-semidefinite matrices, and more specifically, to the metric investigated in (4). However, this metric is parametrized, and the parameter should ideally be learned from the data. In addition, this paper adopts Dynamic Time Warping for sequence alignment. The resulting distance does not generally lead to a positive-definite kernel for classification.

All the approaches described above rely on the use of *hand-crafted features* enabling representation of the action as a trajectory or point in some suitable manifold. Differently from these approaches, many neural network models have been proposed that rely on training for the extraction of *deep learned features*. Recurrent neural networks

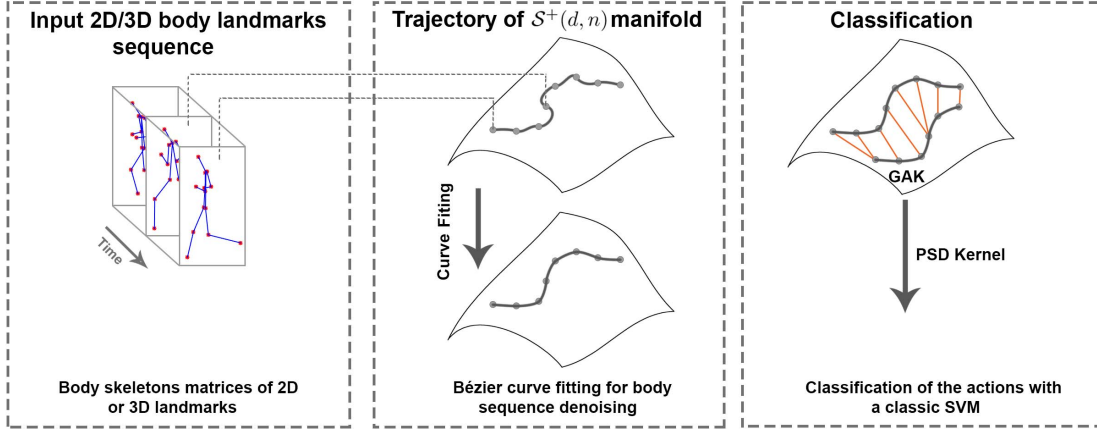


Figure 1. Overview of the proposed approach - After automatic body skeletons detection for each frame of a sequence, the Gram matrices are computed to build the trajectory on the $\mathcal{S}^+(d, n)$ manifold. We apply a curve fitting algorithm on the trajectory to smooth the curve and reduce noise. Global Alignment Kernel (GAK) is then used to align the trajectories on the manifold. Finally, we use the kernel generated from GAK with SVM to classify the actions.

(RNNs) and particularly Long-Short-Term Memory Networks (LSTMs) have been used to perform action recognition by the analysis of sequences of skeleton poses (53). However, these methods typically lose structural information when converting the skeleton data and joint connectivity into the vector-shaped input of the neural network. As an alternative, some approaches introduce Convolutional Neural Networks (CNNs) (24) and Graph Convolutional Networks (GCNs) (49; 23) in the overall architecture so as to retain the structural information among joints of the skeleton. Although these approaches result in state-of-the-art performance (11) on public action recognition benchmarks, it is not possible to define a formal mathematical framework to compute a valid metric on the internal, learned feature representation so as to perform a statistical analysis of the learned actions.

3. Our Approach

3.1. Shape Representation

To represent body movement dynamics, we rely on the time series made of the coordinates of the n tracked body points (*i.e.*, $p_1 = (x_1, y_1), \dots, p_n = (x_n, y_n)$ in 2D, or $p_1 = (x_1, y_1, z_1), \dots, p_n = (x_n, y_n, z_n)$ in 3D), during each video sequence. Each video sequence is thus characterized by a set of landmark configurations $\{Z_0, \dots, Z_\tau\}$, where τ is the number of frames of the video sequence, and where each configuration matrix Z_i ($1 \leq i \leq \tau$) $\in \mathbb{R}^{n \times d}$ encodes the position of the n landmarks in d dimensions (with $d = 2$ or $d = 3$). We aim to measure the dynamic changes of the curves made of the landmark configurations, remaining invariant to rotation and translation. Similarly as in (18), this goal is achieved through a Gram matrix repre-

sentation, where we compute the Gram matrices as:

$$G = ZZ^T. \quad (1)$$

These Gram matrices are $n \times n$ positive-semidefinite matrices, of rank smaller than or equal to d (always equal to d in the datasets considered). Conveniently for us, the Riemannian geometry of the space $\mathcal{S}^+(d, n)$ of $n \times n$ positive-semidefinite matrices of rank d has been studied in (4; 17; 41; 42; 28; 30), and used in, *e.g.*, (10; 32; 14; 29).

A classical approach in the design of algorithms on manifolds consists in resorting to first order local approximations on the manifold, called tangent spaces. This requires two tools: the Riemannian exponential (that allows us to map tangent vectors from the tangent space to the manifold), and the Riemannian logarithm (mapping points from the manifold to the tangent space).

In (18), the manifold $\mathcal{S}^+(d, n)$ is identified to the quotient manifold $(\text{St}(d, n) \times \mathcal{P}_d) / \mathcal{O}_d$, where $\text{St}(d, n) := \{Y \in \mathbb{R}^{n \times d} | Y^T Y = I_d\}$ is the Stiefel manifold, \mathcal{P}_d is the manifold of $d \times d$ positive-definite matrices, and \mathcal{O}_d is the orthogonal group in dimension d . We consider here another representation of the manifold $\mathcal{S}^+(d, n)$, that will result in different expressions for the distance between two points, the Riemannian exponential and logarithm.

3.2. The Quotient Manifold $\mathbb{R}_*^{n \times d} / \mathcal{O}_d$

We consider here the identification of $\mathcal{S}^+(d, n)$ to the quotient manifold $\mathbb{R}_*^{n \times d} / \mathcal{O}_d$, where $\mathbb{R}_*^{n \times d}$ is the set of full-rank $n \times d$ matrices. This geometry has been studied in (17; 28; 30).

The identification of $\mathcal{S}^+(d, n)$ with the quotient $\mathbb{R}_*^{n \times d} / \mathcal{O}_d$ comes from the following observation. Any PSD matrix $G \in \mathcal{S}^+(d, n)$ can be factorized as $G = ZZ^T$, with

$Z \in \mathbb{R}_*^{n \times d}$. However, this factorization is not unique, as any matrix $\tilde{Z} := ZQ$, with $Q \in \mathcal{O}_d$, satisfies $\tilde{Z}\tilde{Z}^T = ZQQ^T Z^T = G$. The two points Z and \tilde{Z} are thus *equivalent* with respect to this factorization, and the set of equivalent points

$$ZO_d := \{ZQ | Q \in \mathcal{O}_d\},$$

is called the equivalence class associated to G . The quotient manifold $\mathbb{R}_*^{n \times d} / \mathcal{O}_d$ is defined as the set of equivalence classes. The mapping $\pi : \mathbb{R}_*^{n \times d} \rightarrow \mathbb{R}_*^{n \times d} / \mathcal{O}_d$, mapping points to their equivalence class, induces a Riemannian metric on the quotient manifold from the Euclidean metric in $\mathbb{R}_*^{n \times d}$. This metric results in the following distance between PSD matrices:

$$\delta(G_i, G_j) = \left[\text{tr}(G_i) + \text{tr}(G_j) - 2\text{tr} \left(\left(G_i^{\frac{1}{2}} G_j G_i^{\frac{1}{2}} \right)^{\frac{1}{2}} \right) \right]^{\frac{1}{2}}. \quad (2)$$

This distance can be expressed in terms of the landmarks variables $Z_i, Z_j \in \mathbb{R}_*^{n \times d}$ as follows:

$$d(G_i, G_j) = \min_{Q \in \mathcal{O}_d} \|Z_j Q - Z_i\|_F. \quad (3)$$

The optimal solution is $Q^* := VU^\top$, where $Z_i^\top Z_j = U\Sigma V^\top$ is a singular value decomposition.

As stated by the next theorem, when $d = 2$, the distance can also be formulated as follows:

Theorem. *Let $G_i, G_j \in \mathcal{S}^+(2, n)$ be two Gram matrices, obtained from landmark matrices $Z_i, Z_j \in \mathbb{R}^{n \times 2}$. The Riemannian distance (2) can be expressed as:*

$$d(G_i, G_j) = \text{tr}(G_i) - 2\sqrt{(a+d)^2 + (c-b)^2} + \text{tr}(G_j), \quad (4)$$

$$\text{where } Z_j^\top Z_i = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

Proof. See Appendix. ■

Expressions for the Riemannian exponential and logarithm are given in (28). We used the implementations provided in the Manopt toolbox (5).

4. Trajectory Modeling

The dynamic changes of body joints movement are characterized by trajectories on the Riemannian manifold of positive-semidefinite matrices of fixed rank (see Fig. 1). More specifically, we fit a curve β_G to a sequence of landmark configurations $\{Z_0, \dots, Z_\tau\}$ represented by their corresponding Gram matrices $\{G_0, \dots, G_\tau\}$ in $\mathcal{S}^+(d, n)$. This curve will enable us to model the spatio-temporal evolution of the elements on $\mathcal{S}^+(d, n)$.

Modeling a sequence of landmarks as a piecewise-geodesic curve on $\mathcal{S}^+(d, n)$ showed very promising results

when the data are well acquired, *i.e.*, without tracking errors or missing data, see (18; 19; 34). To account for both missing data and tracking errors, we rely on a more recent curve fitting algorithm: fitting by composite cubic blended curves, proposed in (13, §5). Specifically, given a set of points $G_0, \dots, G_\tau \in \mathcal{S}^+(d, n)$ associated to times t_0, \dots, t_τ , with $t_i := i$, the curve β_G , defined on the interval $[0, \tau]$, is defined as:

$$\beta_G(t) := \gamma_i(t - i), \quad t \in [i, i + 1],$$

where each curve γ_i is obtained by blending together fitting cubic Bézier curves computed on the tangent spaces based on the data points d_i and d_{i+1} .

These fitting cubic Bézier curves depend on a parameter λ , allowing us to balance two objectives: proximity to the data points at the associated time instants, and regularity of the curve (measured in terms of mean square acceleration). A high value of λ results in a curve with possibly high acceleration, but that will almost interpolate the data, while taking $\lambda \rightarrow 0$ will result in a geodesic. The interested reader can refer to (13, §5) for more information about the curve fitting procedure.

5. Classification

Now that we have defined how to represent a sequence and how to compare two distinctive landmark configurations, we present in this section how we compare two landmark sequences and how to classify the actions performed in these same sequences.

5.1. Global Alignment

As we described in Section 4, we represent a sequence as a trajectory of Gram matrices in $\mathcal{S}^+(d, n)$. The sequences represented in this manifold can be of different length as the execution rate of the actions can vary from one person to another, meaning that we can not effectively compare them. A common method to do so is to use Dynamic Time Warping (DTW) as proposed in several works (3; 18; 15). However, DTW does not define a proper metric and can not be used to derive a valid positive-definite kernel for the classification phase. To address the problem of non positive definiteness of the kernel defined by DTW, Cuturi *et al.* (8) proposed the Global Alignment Kernel (GAK), which allows us to derive a valid positive-definite kernel when aligning two time series. More recently Otterdout *et al.* (33) have proposed to classify deep trajectories in SPD manifold using GAK. The generated kernel can be used directly with Support Vector Machine (SVM) for the classification phase, whereas it is not the case with kernels generated with DTW. In fact, the kernels built with DTW do not show favorable positive definiteness properties as they rely on the computation of an optimum rather than the construction of a feature map. Note

that the computation of the kernels with GAK can be done in quadratic complexity, similarly to naive implementation of DTW. The next paragraph describes how to compute the similarity score between two sequences, using this Global Alignment Kernel.

Let us now consider $Z^i = \{Z_0^1, \dots, Z_{\tau_1}^1\}$, where $Z_{0 \leq i \leq \tau_1}^1$ and $Z^j = \{Z_0^2, \dots, Z_{\tau_2}^2\}$, where $Z_{0 \leq j \leq \tau_2}^2$ two sequences. Given a metric to compute the distance between two elements of each sequence, we propose to compute the matrix D of size $\tau_1 \times \tau_2$, where each $D(i, j)$ is the distance between two elements of the sequences, with $1 \leq i \leq \tau_1$ and $1 \leq j \leq \tau_2$.

$$D(i, j) = d(Z_i^1, Z_j^2). \quad (5)$$

The kernel \tilde{k} can now be computed using the halved Gaussian Kernel on this same matrix D . Therefore, the kernel \tilde{k} can be defined as:

$$\tilde{k}(i, j) = \frac{1}{2} * \exp\left(-\frac{D(i, j)}{\sigma^2}\right). \quad (6)$$

This kernel \tilde{k} is by construction positive semi-definite but in certain cases it may not be. As reported in (8), we can redefine our kernel such as:

$$k(i, j) = \frac{\tilde{k}(i, j)}{(1 - \tilde{k}(i, j))}. \quad (7)$$

This strategy assures us that the kernel is positive semi-definite and can be used in its own. Finally, we can compute the similarity score between the two sequences Z^i and Z^j . Remember that this computation is performed in quadratic complexity, like DTW. To do so, we define a new matrix M that will contain the path to the similarity between our two sequences. We define M as a zeros matrix of size $(\tau_1 + 1) \times (\tau_2 + 1)$ and $M_{0,0} = 1$. Computing the terms of M is done using Theorem 2 in (8, §2.3):

$$M_{i,j} = (M_{i,j-1} + M_{i-1,j-1} + M_{i-1,j}) * k(i, j). \quad (8)$$

The similarity score we seek is the value at $M_{(\tau_1+1),(\tau_2+1)}$. Algorithm 1 describes all the steps to get the similarity score.

Finally, we build a new matrix K of size $n_{seq} \times n_{seq}$, where n_{seq} is the number of sequences in the dataset we test. This matrix is symmetric and contains all the similarity scores between all the sequences of the dataset and it is used as the kernel for the classification phase with SVM. As this matrix is built with values computed from positive semi-definite kernel, it is a positive semi-definite matrix itself.

5.2. Classification with SVM

Our trajectory representation reduces the problem of landmark sequence classification to that of trajectory classification in $\mathcal{S}^+(d, n)$. Given that GAK provides a valid PSD

input : Two sequences of landmark configurations

$Z^1 = \{Z_0^1, \dots, Z_{\tau_1}^1\}$, where $Z_{0 \leq i \leq \tau_1}^1$ and

$Z^2 = \{Z_0^2, \dots, Z_{\tau_2}^2\}$, where $Z_{0 \leq j \leq \tau_2}^2$.

output: The similarity score between two sequences Z^1, Z^2

$\tilde{k} \leftarrow \frac{1}{2} * \exp\left(-\frac{D(Z^1, Z^2)}{\sigma^2}\right)$ Equations (5) and (6)

for $i \leftarrow 0$ **to** τ_1 **do**

for $j \leftarrow 0$ **to** τ_2 **do**

$k(i, j) \leftarrow \frac{\tilde{k}(i, j)}{(1 - \tilde{k}(i, j))}$ Equation (7)

end

end

$M \leftarrow \text{zeros}(\tau_1 + 1, \tau_2 + 1)$

$M_{0,0} \leftarrow 1$

for $i \leftarrow 1$ **to** $\tau_1 + 1$ **do**

for $j \leftarrow 1$ **to** $\tau_2 + 1$ **do**

$M_{i,j} \leftarrow$

$(M_{i,j-1} + M_{i-1,j-1} + M_{i-1,j}) * k(i, j)$ See Equation (8)

end

end

$\text{similarity} \leftarrow M_{\tau_1+1, \tau_2+1}$

return similarity, the similarity score between Z^1 and Z^2

Algorithm 1: Computing the similarity score between two sequences using Global Alignment Kernel (8)

kernel as demonstrated by Cuturi *et al.* (8), and given that our local kernel K satisfies this condition as discussed before, we use the standard SVM with the K kernel that represents the matrix containing the similarity scores between all the sequences of a dataset to classify the aligned trajectories with global alignment on $\mathcal{S}^+(d, n)$.

By contrast, DTW cannot define a positive definite kernel. Hence, we adopt the pairwise proximity function SVM (ppfSVM), which assumes that instead of a valid kernel function, all that is available is a proximity function without any restriction. That is, let us consider $\mathcal{T} = \{\beta_G : [0, 1] \rightarrow \mathcal{S}^+(d, n)\}$, the set of time-parameterized trajectories of the underlying manifold. Like in (18, §4.1), we define a matrix D_{dtw} containing the similarity measure between two trajectories aligned with DTW. In that case, given m trajectories in \mathcal{T} , the proximity function $\mathcal{P} : \mathcal{T} \times \mathcal{T} \rightarrow \mathbb{R}^+$ between two trajectories Z^1 and Z^2 is defined by,

$$\mathcal{P}(Z^1, Z^2) = D_{dtw}(Z^1, Z^2). \quad (9)$$

Using this proximity function, the main idea of ppfSVM is to represent each training example Z with a vector $[\mathcal{P}(Z, Z^1), \dots, \mathcal{P}(Z, Z^m)]^T$. The set of trajectories can be represented by a $m \times m$ matrix P , where $P(i, j) = \mathcal{P}(Z^1, Z^2)$, with $1 \leq i, j \leq m$. From this matrix P we can use a classical linear SVM.

6. Experimental Results

To validate the proposed approach, we have conducted experiments on three publicly available datasets with 3D and 2D actions: UTKinect-Action3D, KTH-Action and UAV-Gesture. Our experiments followed the experimental settings commonly used for these datasets.

6.1. UTKinect-Action3D Dataset

The UTKinect-Action3D dataset (48) is a widely used dataset for 3D action recognition. It contains 199 sequences, consisting of 10 actions, namely *walk*, *sit down*, *stand up*, *pick up*, *carry*, *throw*, *push*, *pull*, *wave hands* and *clap hands* performed by 10 different subjects. The videos and the skeletons were captured with a Microsoft Kinect and the skeletons are composed of 20 body joints. In our approach, we use the available skeletal joint locations, where each body joint is defined with its x , y and z coordinates. Following the same experimental settings of (40; 26; 20), we performed the Leave-One-Out cross validation, meaning that we used one sequence for testing and the rest for training. Our experimental results are summarized in Table 1. In particular, the columns are as follows: *Curve Fitting* indicates if we performed the curve fitting algorithm described in Section 4; *Lambda* indicates the value of the lambda parameter in curve fitting; *Alignment Method* indicates if we used the standard DTW to align sequences or GAK as described in Section 5.1; *Sigma* indicates the value of the sigma parameter for the Gaussian Kernel when using GAK; and *Results* indicates our scores.

Table 1. Our results on the UTKinect-Action3D dataset

Curve Fitting	Lambda	Alignment Method	Sigma	Results
Yes	0.5	DTW	-	97%
No	-	DTW	-	97.49%
Yes	0.5	GAK	0.3	97.49%
No	-	GAK	0.3	97.99%
Yes	0.5	GAK	0.5	97.99%

The best accuracy that we obtained on this dataset is 97.99%. Overall, we can say that the application of curve fitting does not increase our results. Our assumption is that the data in this dataset are very clean, and we can lose some information with the application of smoothing on clean data. Note that we obtained better results when using the Global Alignment Kernel rather than DTW.

Table 2. Comparison of our approach with state-of-the-art results for the UTKinect-Action3D dataset. *: Deep Learning approach

Methods	Protocol	
	H-H	LOOCV
Trajectory on $\mathcal{S}^+(d, n)$ (18) (2019)	-	96.48%
SCK+DCK (22) (2016)	98.2%	-
Bi-LSTM (40) (2018)*	-	98.49%
LM ³ TL (50) (2017)	-	98.8%
GCA-LSTM (26) (2018)*	-	99%
MTCNN (20) (2018)*	-	99%
Hankel & Gram matrices (52) (2016)	-	100%
Ours	-	97.99%

In Table 2, we compare our method with recent state-of-the-art results. Overall, our approach achieves competitive results with respect to most recent approaches. We directly compare our results with (18) as we work on the same geometric space of $\mathcal{S}^+(d, n)$ manifold. The main differences between our method and the method in (18) is the use of a different metric and of the Global Alignment Kernel instead of DTW. Our metric is simpler than the metric in (18), as we do not have to estimate the parameter k used in Eq. (7) in (18) for distance computation. Furthermore, the k parameter in (18) is more of a constraint as they have to determine its best value for each dataset they test. The use of GAK is also an advantage for us as it defines a positive semi-definite kernel, which is not the case for DTW allowing us to use a classic SVM instead of ppfSVM.

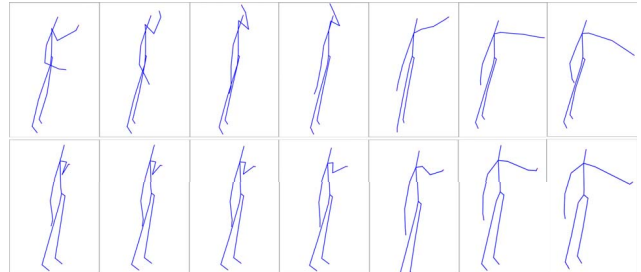


Figure 2. Comparison of two sequences that are confused in UTKinect-Action3D dataset (top: *Throw* action, bottom: *Push* action)

The Figure 2 presents two sequences that are confused, leading to a misclassification for one of them. In that case, the top action (*i.e.* Throw) is misclassified as the bottom action (*i.e.* Push). One of the reasons can be the position of the arm at the end of the action, which is the same in the two sequences. The *Throw* action is the most confused action in the dataset.

6.2. KTH-Action Dataset

The KTH-Action dataset (37) is a 2D action recognition dataset. It consists of six actions, namely *boxing*, *handclapping*, *handwaving*, *jogging*, *running* and *walking* performed by 25 subjects in four different conditions, which are outdoor, outdoor with scale variations, outdoor with different

clothes and indoor. The sequences were acquired with a static camera at a frame rate of 25 fps and a resolution of 160×120 pixels. The dataset contains a total of 599 clips, with 100 clips per actions (1 clip is missing for one action). As the sequences in the dataset are 2D videos, we have to extract the skeletons of the subjects performing the actions. To do so, we used the OpenPose framework (6) to extract the skeletons in the COCO format, with 18 body joints. Note that we clean the landmark sequences by removing the frames where the body joints were not effectively estimated. Keeping all the frames leads to worst results due to misdetected joints, meaning that we do not need all the frames available to recognize an action. Figure 3 shows the configuration of the body joints that we analyzed. For this dataset, we followed the Leave-One-Actor-Out cross validation protocol, meaning that we use one subject for testing and the rest for training. Table 3 summarizes our experimental results on this dataset.

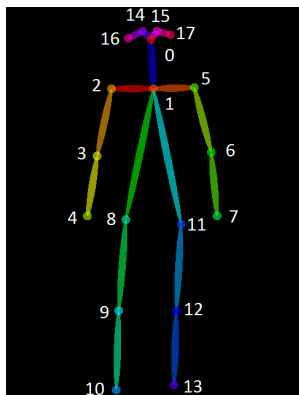


Figure 3. Skeleton with the COCO format.

Table 3. Our experimental results on the KTH-Action dataset

Curve Fitting	Lambda	Alignment Method	Sigma	Results
No	-	DTW	-	94.49%
Yes	10	DTW	-	94.66%
No	-	GAK	0.2	95.16%
Yes	10	GAK	0.2	96.16%

The best accuracy that we obtained on this dataset is 96.16%, using the GAK alignment method and the optimized metric for the 2D case (*i.e.* see Eq. (4)). Here, again, we obtained better results when using the GAK, demonstrating superior performance over DTW. The results reported with DTW are the best accuracy over all the configurations we tested. Unlike the data in UTKinect-Action3D dataset, the data in KTH-Action are 2D and low resolution videos, with presence of noise in the background, leading to noisy skeleton data after extraction. In this regard, the application of the curve fitting algorithm improves our results by 1%. We compare our approach with the state-of-the-art in Table 4. Overall, our method achieves competitive results with recent approaches, while only using skeletal data.

Table 4. Comparison of our approach with state-of-the-art results for the KTH-Action dataset. *: Deep Learning approach

Methods	Input data	Protocol	Accuracy
Schüldt <i>et al.</i> (37) (2004)	RGB	Split	71.7%
Liu <i>et al.</i> (25) (2009)	RGB	LOAO	93.8%
Yoon <i>et al.</i> (51) (2010)	Skeleton	-	89%
Raptis & Soatto (36) (2010)	RGB	LOAO	94.5%
Wang <i>et al.</i> (47) (2011)	RGB	Split	94.2%
Gilbert <i>et al.</i> (12) (2011)	RGB	LOAO	95.7%
Jiang <i>et al.</i> (16) (2012)	RGB	LOAO	95.77%
Vrigkas <i>et al.</i> (46) (2014)	RGB	LOAO	98.3%
Veeriah <i>et al.</i> (43) (2015)*	RGB	Split	93.96%
Liu <i>et al.</i> (27) (2016)	RGB	Split	95%
Almeida <i>et al.</i> (1) (2017)	RGB	LOAO	98%
Our	Skeleton	LOAO	96.16%

6.3. UAV-Gesture Dataset

The UAV-Gesture dataset (35) is a 2D videos dataset, consisting of 13 actions corresponding to UAV (*i.e.*, Unmanned Aerial Vehicles) gesture signals. These actions are *All Clear*, *Have Command*, *Hover*, *Land*, *Landing Direction*, *Move Ahead*, *Move Downward*, *Move To Left*, *Move To Right*, *Move Upward*, *Not Clear*, *Slow Down* and *Wave Off*. The actions are performed by 11 different subjects in an outdoor scenario with slight camera movements. The dataset contains 119 high-quality clips consisting of 37151 frames. As reported in (35), this dataset is not primarily designed for action recognition, but it can be used for this specific task. The skeletons are available with the dataset and the OpenPose framework was also used to extract them in the COCO format. Table 5 compares our results with the baseline experiment reported in (35).

Table 5. Comparison of our approach with the baseline on the UAV-Gesture dataset. *: Deep Learning approach

Method	Curve Fitting	Lambda	Alignment Method	Results
P-CNN (35) (2018)*	-	-	-	91.9%
Ours	No	-	GAK	91.6%
Ours	Yes	10	GAK	92.44%

This is a very recent dataset and its principal interest does not rely on action recognition, meaning a lack of results to compare our results with. However, the authors have tested their dataset for the case of action recognition based on skeletons with Pose-Based Convolutional Neural Network (P-CNN) descriptors, that gives us a baseline to compare our results. The baseline achieves an accuracy of 91.9% with a Deep Learning based approach, whereas our approach achieves an accuracy of 92.44%, outperforming the state-of-the-art results when applying curve fitting and the GAK alignment method.

The Figure 4 presents two very similar actions in the dataset, that is *All Clear* and *Not Clear*. The only big difference between these two actions is the orientation of the hand on the raised arm. This information is not captured when only retrieving the body skeleton from the OpenPose framework, even if it is possible to retrieve the hand skeleton. Note that with our method, the two actions are only

confused three time on a total of 22 sequences, meaning that our method is capable of differentiate minimal changes in the actions.

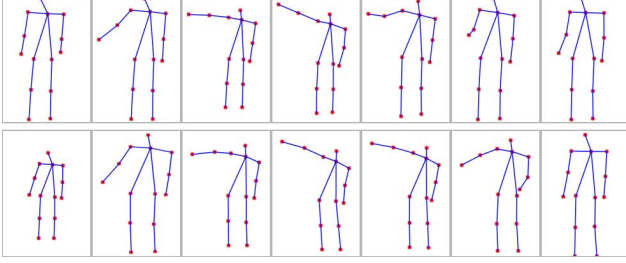


Figure 4. Comparison of two sequences that are confused in UAV-Gesture dataset (top: *All Clear* action, bottom: *Not Clear* action)

6.4. Computation Time Comparison

In this analysis, we have computed the time for each step of our approach. Applying the curve fitting algorithm can be resource demanding as it has to estimate control points to fit the data points of the trajectory. The alignment method can also be resource demanding, regarding the size of the trajectory. With all these parameters in mind, we propose to compute the time that our method takes to compute specific tasks from pose extraction to action classification. The tests were conducted on a laptop equipped with an Intel Core i7-8750H CPU, 16G of RAM and a NVidia Quadro P1000 GPU. Table 6 and Table 7 summarize the execution time of each part of our method for the KTH-Action and UAV-Gesture datasets, respectively. For metric notation, M_1 refers to Eq. (3) and M_2 refers to Eq. (4), in Section 3.2.

Table 6. Execution time (in seconds) obtained on the KTH-Action dataset for the different steps of the method for one sequence.

Pose extraction	Curve Fitting	Alignment method - Metric	Alignment	Classification
147	0.069	DTW - M_1	0.034	0.41
147	0.069	DTW - M_2	0.02	0.41
147	0.069	GAK - M_1	0.04	0.49
147	0.069	GAK - M_2	0.019	0.49

Table 7. Execution time (in seconds) obtained on the UAV-Gesture dataset for the different steps of our method for one sequence.

Pose extraction	Curve Fitting	Alignment method - Metric	Alignment	Classification
-	0.504	DTW - M_1	0.128	0.038
-	0.504	DTW - M_2	0.072	0.038
-	0.504	GAK - M_1	0.138	0.53
-	0.504	GAK - M_2	0.066	0.053

For the KTH-Action dataset, we consider a sequence of 61 frames and a sequence of 192 frames for UAV-Gesture. First, we can observe that the pose extraction phase takes most of the execution time for the KTH-Action dataset. This is partially due to the fact that our GPU is not powerful enough (we get around 3.5fps with our Quadro P1000). The extraction time is not reported for the UAV-Gesture dataset as the skeletons are available with the dataset. The second thing we can observe is the low difference in computation

time for the alignment part when switching from DTW to GAK. We can also note that when we use M_2 , the computation time can be reduce by a factor of 2 compared to the use of M_1 , demonstrating that the use of an optimized metric can speed up the computation time. If we only consider the execution time for the treatment of the skeletons, it takes around 0.499 seconds to classify an action of the KTH-Action dataset and around 0.614 seconds for an action of the UAV-Gesture dataset in the best case scenario.

7. Conclusion and Future Work

In this paper, we have proposed a method for comparing and classifying temporal sequences of 2D/3D landmarks on the positive semi-definite manifold. Our approach involves three different steps: 1) We build a trajectory on the Riemannian manifold from the body skeletons; 2) we apply a curve fitting algorithm on the trajectories to denoise the data points; 3) we perform a temporal alignment using a Global Alignment Kernel. Our experiments on three publicly available datasets show that the proposed approach gives competitive results with respect to state-of-the-art methods.

8. Acknowledgement

We thank Dr. A. Kacem for fruitful discussions. We also thank Prof. J-C. Alvarez Paiva for fruitful discussions on the formulation of the distance between $n \times 2$ landmarks configurations.

9. Appendix

Proof. of Theorem 3.2. We can reformulate our metric introduced in Eq. (3) with:

$$\begin{aligned} d(G_i, G_j) &= \text{tr} [(Z_j Q - Z_i)(Z_j Q - Z_i)^T] \\ &= \text{tr}(G_i) - 2 \text{tr}(Z_i Q^T Z_j^T) + \text{tr}(G_j). \end{aligned}$$

To minimize our distance, we need to maximize the term $\text{tr}(Z_i Q^T Z_j^T)$. Consider now that $Z_j^T Z_i$ is defined by a 2×2 matrix with four unknown values a, b, c, d and Q is a 2D rotation matrix, we now have to maximize:

$$\max \text{tr} \left[\begin{pmatrix} a \cos \Theta - b \sin \Theta & - \\ - & c \sin \Theta + d \cos \Theta \end{pmatrix} \right]. \quad (10)$$

From Eq. (10) we now have to find the maximum of $(a + d) \cos \Theta + (c - b) \sin \Theta$, meaning that we have to maximize $\sqrt{(a + d)^2 + (c - b)^2} \cos(O - O')$. As we want to maximize this value, O has to be equal to O' , so $\sqrt{(a + d)^2 + (c - b)^2} \cos(O - O') \leq \sqrt{(a + d)^2 + (c - b)^2}$. Therefore we can say that:

$$\max \text{tr}(Z_i Q^T Z_j^T) = \sqrt{(a + d)^2 + (c - b)^2}. \quad (11)$$

■

References

- [1] R. Almeida, Z. K. G. do Patrocínio Jr., and S. J. F. Guimarães. Exploring quantization error to improve human action classification. In *2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017*, pages 1354–1360, 2017. 7
- [2] R. Anirudh, P. K. Turaga, J. Su, and A. Srivastava. Elastic functional coding of riemannian trajectories. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 39(5):922–936, 2017. 2
- [3] B. Ben Amor, J. Su, and A. Srivastava. Action recognition using rate-invariant analysis of skeletal shape trajectories. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 38(1):1–13, 2016. 2, 4
- [4] S. Bonnabel and R. Sepulchre. Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1055–1070, 2009. 2, 3
- [5] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15:1455–1459, 2014. 4
- [6] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *CoRR*, abs/1812.08008, 2018. 7
- [7] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 1
- [8] M. Cuturi, J. Vert, Ø. Birkenes, and T. Matsui. A kernel for time series based on global alignments. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2007, Honolulu, Hawaii, USA, April 15-20, 2007*, pages 413–416, 2007. 4, 5
- [9] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo. 3-D human action recognition by shape analysis of motion trajectories on Riemannian manifold. *IEEE Trans. on Cybernetics*, 45(7):1340–1352, 2015. 2
- [10] M. Faraki, M. T. Harandi, and F. Porikli. Image set classification by symmetric positive semi-definite matrices. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–8. IEEE, 2016. 3
- [11] X. Gao, W. Hu, J. Tang, J. Liu, and Z. Guo. Optimized skeleton-based action recognition via sparsified graph regression. *arXiv:1811.12013*, 2019. 3
- [12] A. Gilbert, J. Illingworth, and R. Bowden. Action recognition using mined hierarchical compound features. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):883–897, 2011. 7
- [13] P.-Y. Gousenbourger, E. Massart, and P.-A. Absil. Data fitting on manifolds with composite Bézier-like curves and blended cubic splines. *Journal of Mathematical Imaging and Vision*, 61(5):645–671, 2018. 4
- [14] P.-Y. Gousenbourger, E. Massart, A. Musolas, P.-A. Absil, L. Jacques, J. M. Hendrickx, and Y. Marzouk. Piecewise-Bézier C^1 smoothing on manifolds with application to wind field estimation. *Proceedings of the 25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 305–310, 2017. 3
- [15] A. Gritai, Y. Sheikh, C. Rao, and M. Shah. Matching trajectories of anatomical landmarks under viewpoint, anthropometric and temporal transforms. *Int. Journal of Computer Vision*, 84(3):325–343, 2009. 4
- [16] Z. Jiang, Z. Lin, and L. S. Davis. Recognizing human actions by learning and matching shape-motion prototype trees. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(3):533–547, 2012. 7
- [17] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010. 3
- [18] A. Kacem, M. Daoudi, B. Ben Amor, S. Berretti, and J. C. Alvarez-Paiva. A novel geometric framework on gram matrix trajectories for human behavior understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 1, 2, 3, 4, 5, 6
- [19] A. Kacem, M. Daoudi, B. Ben Amor, and J. C. Á. Paiva. A novel space-time representation on the positive semidefinite cone for facial expression recognition. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 3199–3208, 2017. 4
- [20] Q. Ke, M. Bennamoun, S. An, F. A. Sohel, and F. Boussaid. Learning clip representations for skeleton-based 3d action recognition. *IEEE Trans. Image Processing*, 27(6):2842–2855, 2018. 6
- [21] Y. Kong and Y. Fu. Human action recognition and prediction: A survey. *arXiv:1806.11230*, 2018. 2
- [22] P. Koniusz, A. Cherian, and F. Porikli. Tensor representations via kernel linearization for action recognition from 3d skeletons. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pages 37–53, 2016. 6
- [23] B. Li, X. Li, Z. Zhang, and F. Wu. Spatio-temporal graph routing for skeleton-based action recognition. In *AAAI Conference on Artificial Intelligence*, 2019. 3
- [24] C. Li, Q. Zhong, D. Xie, and S. Pu. Skeleton-based action recognition with convolutional neural networks. In *IEEE International Conference on Multimedia & Expo Workshops*, page 597–600, 2017. 3
- [25] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 1996–2003, 2009. 7
- [26] J. Liu, G. Wang, L. Duan, K. Abdiyeva, and A. C. Kot. Skeleton-based human action recognition with global context-aware attention LSTM networks. *IEEE Trans. Image Processing*, 27(4):1586–1599, 2018. 6
- [27] L. Liu, L. Shao, X. Li, and K. Lu. Learning spatio-temporal representations for action recognition: A genetic programming approach. *IEEE Trans. Cybernetics*, 46(1):158–170, 2016. 7
- [28] E. Massart and P.-A. Absil. Quotient geometry with simple geodesics for the manifold of fixed-rank positive-semidefinite matrices. Preprint. 3, 4
- [29] E. Massart, P.-Y. Gousenbourger, N. T. Son, T. Stykel, and P.-A. Absil. Interpolation on the manifold of fixed-rank positive-semidefinite matrices for parametric model reduction: preliminary results. In *Proceedings of the 27th Eu-*

- ropean Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN2019), pages 281–286, 2019. 3
- [30] E. Massart, J. M. Hendrickx, and P.-A. Absil. Curvature of the manifold of fixed-rank positive-semidefinite matrices endowed with the Bures-Wasserstein metric. In *4th conference on Geometric Sciences of Information (GSI 2019, to appear)*, 2019. 3
- [31] P. Matikainen, M. Hebert, and R. Sukthankar. Trajectons: Action recognition through the motion analysis of tracked features. In *Workshop on Video-Oriented Object and Event Classification (ICCV)*, September 2009. 2
- [32] G. Meyer, S. Bonnabel, and R. Sepulchre. Regression on fixed-rank positive semidefinite matrices: a Riemannian approach. *Journal of Machine Learning Research*, 12(Feb):593–625, 2011. 3
- [33] N. Otberdout, A. Kacem, M. Daoudi, L. Ballihi, and S. Berretti. Automatic analysis of facial expressions based on deep covariance trajectories. *CoRR*, abs/1810.11392, 2018. 4
- [34] N. Otberdout, A. Kacem, M. Daoudi, L. Ballihi, and S. Berretti. Deep covariance descriptors for facial expression recognition. In *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, page 159, 2018. 4
- [35] A. G. Perera, Y. W. Law, and J. S. Chahl. UAV-GESTURE: A dataset for UAV control and gesture recognition. In *Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part II*, pages 117–128, 2018. 7
- [36] M. Raptis and S. Soatto. Tracklet descriptors for action modeling and video analysis. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I*, pages 577–590, 2010. 7
- [37] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *17th International Conference on Pattern Recognition, ICPR 2004, Cambridge, UK, August 23-26, 2004.*, pages 32–36, 2004. 6, 7
- [38] J. Su, S. Kurttek, E. Klassen, and A. Srivastava. Statistical analysis of trajectories on riemannian manifolds: Bird migration, hurricane tracking and video surveillance. *Annals of Applied Statistics*, 8(1), 2014. 2
- [39] J. Su, A. Srivastava, F. D. M. de Souza, and S. Sarkar. Rate-invariant analysis of trajectories on riemannian manifolds with application in visual speech recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 2
- [40] A. B. Tanfous, H. Drira, and B. B. Amor. Coding kendall’s shape trajectories for 3d action recognition. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2840–2849, 2018. 6
- [41] B. Vandereycken, P.-A. Absil, and S. Vandewalle. Embedded geometry of the set of symmetric positive semidefinite matrices of fixed rank. In *Statistical Signal Processing, 2009. SSP’09. IEEE/SP 15th Workshop on*, pages 389–392. IEEE, 2009. 3
- [42] B. Vandereycken, P.-A. Absil, and S. Vandewalle. A Riemannian geometry with complete geodesics for the set of positive semidefinite matrices of fixed rank. *IMA Journal of Numerical Analysis*, 33(2):481–514, 2013. 3
- [43] V. Veeriah, N. Zhuang, and G. Qi. Differential recurrent neural networks for action recognition. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 4041–4049, 2015. 7
- [44] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3D skeletons as points in a Lie group. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 588–595, 2014. 2
- [45] R. Vemulapalli and R. Chellapa. Rolling rotations for recognizing human actions from 3d skeletal data. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4471–4479, 2016. 2
- [46] M. Vrigkas, V. Karavasilis, C. Nikou, and I. A. Kakadiaris. Matching mixtures of curves for human action recognition. *Computer Vision and Image Understanding*, 119:27–40, 2014. 7
- [47] H. Wang, A. Kläser, C. Schmid, and C. Liu. Action recognition by dense trajectories. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 3169–3176, 2011. 7
- [48] L. Xia, C. Chen, and J. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 20–27. IEEE, 2012. 6
- [49] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI Conference on Artificial Intelligence*, 2018. 3
- [50] Y. Yang, C. Deng, D. Tao, S. Zhang, W. Liu, and X. Gao. Latent max-margin multitask learning with skelets for 3-d action recognition. *IEEE Trans. Cybernetics*, 47(2):439–448, 2017. 6
- [51] S. M. Yoon and A. Kuijper. Human action recognition using segmented skeletal features. In *20th International Conference on Pattern Recognition, ICPR 2010, Istanbul, Turkey, 23-26 August 2010*, pages 3740–3743, 2010. 7
- [52] X. Zhang, Y. Wang, M. Gou, M. Sznajder, and O. I. Camps. Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4498–4507, 2016. 6
- [53] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *AAAI Conference on Artificial Intelligence*, 2016. 3