Efficient Temporal Sequence Comparison and Classification using Gram Matrix Embeddings On a Riemannian Manifold*

Xikang Zhang, Yin Wang, Mengran Gou, Mario Sznaier, Octavia Camps
Electrical and Computer Engineering
Northeastern University, Boston, MA 02115, US
zhangxk@ece.neu.edu, wang.yin@husky.neu.edu, {mengran,msznaier,camps}@coe.neu.edu

Abstract

In this paper we propose a new framework to compare and classify temporal sequences. The proposed approach captures the underlying dynamics of the data while avoiding expensive estimation procedures, making it suitable to process large numbers of sequences. The main idea is to first embed the sequences into a Riemannian manifold by using positive definite regularized Gram matrices *of their Hankelets. The advantages of the this approach are: 1) it allows for using non-Euclidean similarity functions on the Positive Definite matrix manifold, which capture better the underlying geometry than directly comparing the sequences or their Hankel matrices; and 2) Gram matrices inherit desirable properties from the underlying Hankel matrices: their rank measure the complexity of the underlying dynamics, and the order and coefficients of the associated regressive models are invariant to affine transformations and varying initial conditions. The benefits of this approach are illustrated with extensive experiments in 3D action recognition using 3D joints sequences. In spite of its simplicity, the performance of this approach is competitive or better than using state-of-art approaches for this problem. Further, these results hold across a variety of metrics, supporting the idea that the improvement stems from the embedding itself, rather than from using one of these metrics.*

1. Introduction

Comparison and classification of temporal sequences is a key problem in action recognition (Figure 1), event detection and abnormal activity detection. Approaches to this problem may be divided into two categories: model-based and data-driven. Model-based methods assume that each sequence is generated by an implicit dynamic model or Hidden Markov Model (HMM) with some added measurement noise. Query sequences are classified based on whether they fit this model or not. The main disadvantages of these approaches are that the model parameters need to be estimated (or learned) from training data and that the learning and inference are usually time consuming, especially for high order models.

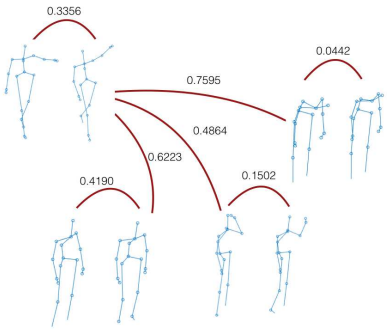

Figure 1: 3D action recognition is an example of the class of problems requiring efficient, robust comparison and classification of temporal sequences.

On the other hand, data-driven methods, rather than assuming an implicit model, postulate that data close to each other, using a suitable metric, should be in the same class. Then, query data is classified with its nearest neighbor class label. While data-driven methods enjoy efficiency and do not require parameter estimation, their success critically hinges on choosing a metric that reflects well the data structure, a task that is far from trivial.

It is well known that the Euclidean distance is unsuitable for comparing temporal sequences: sequences may have large Euclidean distances within class but small distances between classes. Several methods attempt to circumvent this issue by projecting the sequences to Euclidean space. Examples include Dynamic Time Warping (DTW) [24], specialized kernels [39], Fourier hierarchical pyramid [39],

*This work was supported in part by NSF grants IIS–1318145 and ECCS–1404163; AFOSR grant FA9550-15-1-0392; and the Alert DHS Center of Excellence under Award Number 2013-ST-061-ED0001.

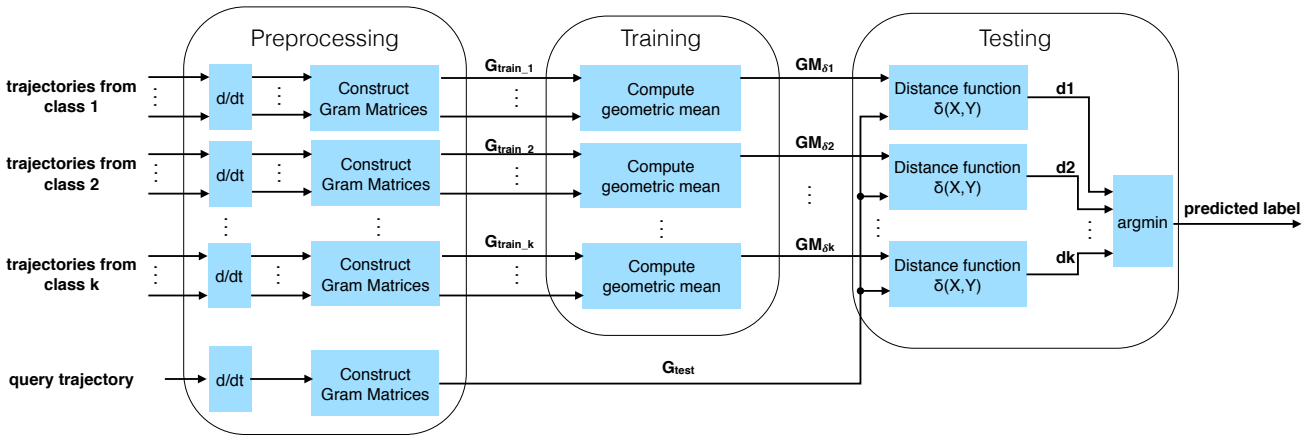

Figure 2: Diagram of the proposed method

and covariance features [17] etc. While these approaches align or transform the data before using an Euclidean metric, none of them takes into account the implicit dynamics of the sequences, which may lead to poor accuracy in some scenarios.

[20] showed that dynamic information can be encapsulated in Hankelets (Hankel matrices associated with short portions of the data). In principle, comparing Hankelets requires computing subspace angles [5], which is non-trivial in the case of noisy data, since it entails rank estimation. Alternatively, [21] proposed a surrogate that does not entail estimating rank. However, since this surrogate is not a metric, it is hard to assess the properties of the induced geometry.

From the discussion above it follows that it is desirable to develop a distance-like function that combines the best features of the methods above, that is, it should a) reflect the similarity of the dynamics underlying the sequences; b) be amenable to efficient computations, robust to noise; and c) either be a metric or share most of its properties, so that the associated geometry can be easily analyzed.

Our main result shows that the objectives above can be achieved by embedding the sequences into the Positive Definite (PD) manifold via suitably regularized Gram matrices. We show that while these matrices contain the same dynamic information as their Hankel counterparts, the embedding allows for blackucing the problem to computing distances between matrices on the PD manifold, which is a well studied problem that can be solved easily. The benefits of the proposed approach are illustrated with 3D action recognition. In this context, sequences with the same dynamic model are very close, while those corresponding to different dynamics are far apart. The experiments show that the proposed approach provides substantial robustness against noise and improves performance over the state-of-art 3D action recognition, both in terms of accuracy and computational efficiency. Most notably, these results hold regardless of the specific metric used, which supports the idea that the performance of the method is due to the embedding itself, rather than the properties of the metric used.

2. Related work

Recent advances in computing distances on the PD manifold include [30] [4] [2] [23]. These methods work well when applied to covariance features in tracking, face recognition and texture classification[8]. However, while there is a standard definition of covariance feature for images, no similar definition is available for dynamic systems.

[17] built a temporal hierarchy of covariance descriptors on 3D joints to classify human actions. [16] extended this feature to infinite dimensional covariances in a Hilbert space, and used metrics in the Riemannian manifold. [13] extended VLAD feature to Riemannian manifolds. These approaches share with the proposed method the fact that temporal sequences are embedded in the PD manifold, rather than handled directly. However, they do not exploit the dynamic information (e.g. model order and invariants) implicit in the data.

Finally, as mentioned earlier, [20, 21] showed that dynamic information can be encapsulated in Hankel matrices and proposed comparing sequences using the Hankelet subspaces angle. Our method shares the idea of exploiting dynamic information through the properties of the subspaces of suitable matrices (Gram rather than Hankel, in our case). However, rather than comparing these subspaces directly, it uses a manifold metric to compare matrices, leading to better performance in the presence of noise.

3. Preliminaries

In this section, we recall the relationship between autoregressive (AR) models and Gram matrices, as well as several distance-like functions on Riemannian manifolds, which will be used to compare Gram matrices and thus the embedded temporal sequences.

3.1. Notation

$\mathbb{R}$	set of real numbers
$\mathcal{S}^n$	set of symmetric matrices in $\mathbb{R}^{n\times n}$
$\mathcal{S}^n_+(\mathcal{S}^n_{++})$	set of positive-semidefinite (-definite) matrices in $\mathbb{R}^{n\times n}$
$\mathbf{x}(\mathbf{X})$	a vector (matrix) in $\mathbb{R}$
$\mathbf{X}(\succeq)\ \succ 0$	$\mathbf{X}$ is positive-(semi)definite
$\mathcal{N}(\mathbf{X})$	null space of $\mathbf{X}$

3.2. Hankel and Gram matrices Representations

Over short horizons, the output $\mathbf{t}_k$ generated by a dynamic system from some initial conditions can be approximated by the output of an AR model of the form [6]:

$$\mathbf{t}_k = \sum_{i=1}^{n} a_i \mathbf{t}_{k-i} \tag{1}$$

In addition, to each sequence $\mathbf{t}_k$ one can associate a block Hankel matrix of the form

$$\mathbf{H}_{\mathbf{t}}^{r,s} = \begin{bmatrix} \mathbf{t}_1 & \mathbf{t}_2 & \mathbf{t}_3 & \cdots & \mathbf{t}_s \\ \mathbf{t}_2 & \mathbf{t}_3 & \mathbf{t}_4 & \cdots & \mathbf{t}_{s+1} \\ \vdots & \vdots & \vdots & \ddots & \cdots \\ \mathbf{t}_r & \mathbf{t}_{r+1} & \mathbf{t}_{r+2} & \cdots & \mathbf{t}_{r+s-1} \end{bmatrix} \tag{2}$$

where r and s determine the shape of the Hankel matrix. For example, if the observations are a set of points in 3D and there are n tracked points in each frame, the block for frame i is given by

$$\mathbf{t}_i = [x_1^i, y_1^i, z_1^i, x_2^i, y_2^i, z_2^i, \cdots, x_n^i, y_n^i, z_n^i]^\top. \tag{3}$$

As pointed out in [21], Hankel matrices carry useful invariant properties. Specifically, if r and s are selected such that $\rho = \text{rank}(\mathbf{H}_{\mathbf{t}}^{r,s}) < \min\{r, s\}$, then ρ measures the complexity of the underlying dynamics in the sense that there exists an AR model of order at most ρ that can generate the observed data. Further, the subspace spanned by the columns of $\mathbf{H_t}$ completely characterizes these dynamics and is invariant to both initial condition and affine viewpoint changes. Thus, in principle Hankel matrices could be compablack by simply comparing the angles between the subspaces spanned by the respective columns. However, this comparison is difficult in the presence of noise. This is due to the fact that, in the case of noisy measurements, $\mathbf{H}$ tends to be full rank, and hence the angle between subspaces becomes zero. Thus, in order to apply these ideas, one needs to first estimate the rank ρ of the underlying clean matrix, a difficult task akin to model order estimation, and take into consideration only ρ principal components when computing the angle. To circumvent this difficulty, in this paper we propose to work with Gram matrices defined as:

$$\hat{\mathbf{G}} = \frac{\mathbf{H}\mathbf{H}^\top}{\|\mathbf{H}\mathbf{H}^\top\|_F} \tag{4}$$

where we follow [21] by using a Frobenius normalization. It can be easily shown that Gram matrices inherit the rank and invariance properties of the associated Hankel matrices. However, in contrast to the later, Gramm matrices are confined to the Positive Semi Definite (PSD) manifold, a fact that is key to the techniques here.

3.3. Distance-like functions in the PD Manifold

In this section, we review some commonly used distance-like functions on the PD Riemannian manifold, which are listed, together with their associated mean formulas, in Table 1. The most widely used among these functions is the Affine Invariant Riemannien Metric (AIRM) [30] [4], which is defined as:

$$\delta_R(\mathbf{X}, \mathbf{Y}) = \|\log(\mathbf{X}^{-1/2}\mathbf{Y}\mathbf{X}^{-1/2})\|_F \tag{5}$$

It can be shown that this is indeed a metric that defines geodesics on the manifold. However, its computational cost is relatively high. Despite its computational inefficiency, it is the most robust metric.

The Log-Euclidean Riemannian Metric (LERM) [2] is defined as:

$$\delta_{le}(\mathbf{X}, \mathbf{Y}) = \|\log(\mathbf{X}) - \log(\mathbf{Y})\|_F \tag{6}$$

Note that in this metric $\mathbf{X}$ and $\mathbf{Y}$ are decoupled, so that each term can be precomputed. While the LERM is computationally more efficient than the AIRM, in many applications it performs worse.

Another two popular distance measures on the PD manifold originate from Bregman divergences. The Jensen-Bregman Log-det Divergence (JBLD) [8], which is also known as Stein Divergence, is defined as

$$\delta^2_{ld}(\mathbf{X}, \mathbf{Y}) = \log\left|\frac{\mathbf{X}+\mathbf{Y}}{2}\right| - \frac{1}{2}\log|\mathbf{X}\mathbf{Y}| \tag{7}$$

Although JBLD itself is not a metric, Sra [34] proved that its square root δ_{ld} is. Computing δ^2_{ld} is very efficient. However, computing its mean is relatively slow since it does not admit a closed form solution. The KL-Divergence metric (KLDM) [23], which is also known as Jeffrey Divergence, is defined as

$$\delta^2_{jkl}(\mathbf{X}, \mathbf{Y}) = \frac{1}{2}\text{Tr}(\mathbf{X}^{-1}\mathbf{Y} + \mathbf{Y}^{-1}\mathbf{X} - 2\mathbf{I}) \tag{8}$$

Table 1: Popular distance measures on Riemannian manifold and their associated means

Distance	Definition	Associated geometric mean
AIRM	$\delta_R(\mathbf{X},\mathbf{Y}) = \|\log(\mathbf{X}^{-1/2}\mathbf{Y}\mathbf{X}^{-1/2})\|_F$	iteration: $\mathbf{X}_{(k+1)} = \mathbf{X}_{(k)}^{\frac{1}{2}} \exp\left(\sum_{i=1}^n \log(\mathbf{X}_{(k)}^{-\frac{1}{2}}\mathbf{X}_i\mathbf{X}_{(k)}^{-\frac{1}{2}})\right)\mathbf{X}_{(k)}^{\frac{1}{2}}$
LERM	$\delta_{le}(\mathbf{X},\mathbf{Y}) = \|\log(\mathbf{X}) - \log(\mathbf{Y})\|_F$	$\mathbf{X}_{le}^* = \exp\left(\frac{1}{n}\sum_{i=1}^n \log(\mathbf{X}_i)\right)$
JBLD	$\delta_{ld}^2(\mathbf{X},\mathbf{Y}) = \log\left\|\frac{\mathbf{X}+\mathbf{Y}}{2}\right\| - \frac{1}{2}\log\|\mathbf{X}\mathbf{Y}\|$	iteration: $\mathbf{X}_{(k+1)} = \left(\sum_{i=1}^n \frac{\mathbf{X}_{(k)}+\mathbf{X}_i}{2}\right)$
KLDM	$\delta_{jkl}^2(\mathbf{X},\mathbf{Y}) = \frac{1}{2}\mathrm{Tr}(\mathbf{X}^{-1}\mathbf{Y} + \mathbf{Y}^{-1}\mathbf{X} - 2\mathbf{I})$	$\mathbf{X}_{jkl}^* = \mathbf{P}^{-\frac{1}{2}}(\mathbf{P}^{\frac{1}{2}}\mathbf{Q}\mathbf{P}^{\frac{1}{2}})^{\frac{1}{2}}\mathbf{P}^{-\frac{1}{2}}$ where $\mathbf{P} = \sum_{i=1}^n \mathbf{X}_i^{-1}$ and $\mathbf{Q} = \sum_{i=1}^n \mathbf{X}_i$

Contrary to the functions above, the KLDM is not a metric, since it does not satisfy the triangular property. It is less efficient than the JBLD but its mean has a closed form, which can be efficiently computed.

Finally, given a metric $\delta_\bullet$ and a set of PD matrices $\{\mathbf{X}_i | i = 1,\cdots,n, \mathbf{X}_i \in \mathbb{R}^{d\times d}, \mathbf{X}_i \succ 0\}$, the associated mean of this set is defined as

$$\mathbf{X}_\bullet^* = \arg\min_{\bar{\mathbf{X}}\succ 0} \sum_{i=1}^n \delta_\bullet^2(\bar{\mathbf{X}}, \mathbf{X}_i) \tag{9}$$

While LERM and KLDM have closed form mean, AIRM and JBLD do not. However, they can be found iteratively.

4. Comparing Temporal Sequences

Based on the discussion above, we propose comparing temporal sequences by using a distance-like function on the PD manifold to compare their associated Gram matrices. This idea is motivated by the following result.

Consider n measurements $\mathbf{z}_k = \mathbf{t}_k + \boldsymbol{\eta}_k$ of $\mathbf{t}_k$, corrupted by $\boldsymbol{\eta}_k$, where $k = 1, 2, \cdots, n$. The corresponding Gram matrix: $\mathbf{H}_\mathbf{z}\mathbf{H}_\mathbf{z}^\top = \mathbf{H}_\mathbf{t}\mathbf{H}_\mathbf{t}^\top + \mathbf{H}_\mathbf{t}\mathbf{H}_{\boldsymbol{\eta}}^\top + \mathbf{H}_{\boldsymbol{\eta}}\mathbf{H}_\mathbf{t}^\top + \mathbf{H}_{\boldsymbol{\eta}}\mathbf{H}_{\boldsymbol{\eta}}^\top$ is generically full rank due to the noise. However, if the noise $\boldsymbol{\eta}$ is zero mean and uncorrelated with $\mathbf{t}$, we have $\mathbf{H}_\mathbf{z}\mathbf{H}_\mathbf{z}^\top \approx \mathbf{H}_\mathbf{t}\mathbf{H}_\mathbf{t}^\top + \mathbf{H}_{\boldsymbol{\eta}}\mathbf{H}_{\boldsymbol{\eta}}^\top = \mathbf{H}_\mathbf{t}\mathbf{H}_\mathbf{t}^\top + n_r\epsilon^2\mathbf{I}$, where n_r is the number of rows of $\mathbf{H}_{\boldsymbol{\eta}}$ and ϵ^2 is the variance of the noise $\boldsymbol{\eta}$.

Theorem 1. *Given $\mathbf{X}, \mathbf{Y} \in \mathcal{S}_n^+$, define the regularized matrices $\mathbf{X}_\sigma = \mathbf{X} + \sigma\mathbf{I}$, $\mathbf{Y}_\sigma = \mathbf{Y} + \sigma\mathbf{I}$, where $\sigma > 0$. Then*

$$\lim_{\sigma\to 0} \delta_\bullet(\mathbf{X}_\sigma, \mathbf{Y}_\sigma) \neq \infty \iff \mathcal{N}(\mathbf{X}) = \mathcal{N}(\mathbf{Y}) \tag{10}$$

where $\delta_\bullet(.,.)$ denotes any of the functions in section 3.3.

Proof. To prove sufficiency, note that since $\mathcal{N}(\mathbf{X}) = \mathcal{N}(\mathbf{Y})$ there exists some unitary matrix $\mathbf{U}$ such that

$$\begin{aligned}\mathbf{U}\mathbf{X}\mathbf{U}^\top + \sigma\mathbf{I} &= \begin{bmatrix}\mathbf{M}_\mathbf{X} + \sigma\mathbf{I}_r & \mathbf{0}\\ \mathbf{0} & \sigma\mathbf{I}_{n-r}\end{bmatrix}\\ \mathbf{U}\mathbf{Y}\mathbf{U}^\top + \sigma\mathbf{I} &= \begin{bmatrix}\mathbf{M}_\mathbf{Y} + \sigma\mathbf{I}_r & \mathbf{0}\\ \mathbf{0} & \sigma\mathbf{I}_{n-r}\end{bmatrix}\end{aligned}$$

where $r \doteq \mathrm{rank}(\mathbf{X})$. Note that the AIRM, JBLD and KLDM are affine invariant [15, 16], while simple computations show that, if $\mathbf{U}$ is unitary, then $\delta_{le}(\mathbf{X},\mathbf{Y}) = \delta_{le}(\mathbf{U}\mathbf{X}\mathbf{U}^\top, \mathbf{U}\mathbf{Y}\mathbf{U}^\top)$ Thus:

$$\begin{aligned}&\delta_\bullet(\mathbf{X}_\sigma, \mathbf{Y}_\sigma) =\\ &\delta_\bullet\left(\begin{bmatrix}\mathbf{M}_\mathbf{X} + \sigma\mathbf{I}_r & \mathbf{0}\\ \mathbf{0} & \sigma\mathbf{I}_{n-r}\end{bmatrix}, \begin{bmatrix}\mathbf{M}_\mathbf{Y} + \sigma\mathbf{I}_r & \mathbf{0}\\ \mathbf{0} & \sigma\mathbf{I}_{n-r}\end{bmatrix}\right) =\\ &\delta_\bullet(\mathbf{M}_\mathbf{X} + \sigma\mathbf{I}_r, \mathbf{M}_\mathbf{Y} + \sigma\mathbf{I}_r)\end{aligned} \tag{11}$$

where the last line follows from computing $\delta_\bullet$ explicitly in each case. The desiblack result follows now by taking limits as $\sigma \to 0$.

To prove necessity, assume that $\mathcal{N}(\mathbf{X}) \neq \mathcal{N}(\mathbf{Y})$. Then, there exists a vector $\mathbf{u}$ such that $\mathbf{X}\mathbf{u} = 0$ and $\mathbf{u}'\mathbf{Y}\mathbf{u} = y > 0$. Define a unitary matrix $\mathbf{U} = \begin{bmatrix}\mathbf{M} & \mathbf{u}\end{bmatrix}$ where $\mathbf{M}$ is an arbitrary matrix such that $\mathbf{M}^\top\mathbf{M} = \mathbf{I}$ and $\mathbf{M}^\top\mathbf{u} = 0$. By construction

$$\begin{aligned}\mathbf{U}^\top(\mathbf{X} + \sigma\mathbf{I})\mathbf{U} &= \begin{bmatrix}\mathbf{M}_X & \mathbf{0}\\ \mathbf{0} & \sigma\end{bmatrix}\\ \mathbf{U}^\top(\mathbf{Y} + \sigma\mathbf{I})\mathbf{U} &= \begin{bmatrix}\mathbf{M}_Y & \mathbf{0}\\ \mathbf{0} & \sigma + y\end{bmatrix}\end{aligned} \tag{12}$$

It can be easily seen that for diagonal matrices,

$$\delta_\bullet^2\left(\begin{bmatrix}\mathbf{D}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{D}_2\end{bmatrix}, \begin{bmatrix}\mathbf{D}_3 & \mathbf{0}\\ \mathbf{0} & \mathbf{D}_4\end{bmatrix}\right) = \delta_\bullet^2(\mathbf{D}_1,\mathbf{D}_3) + \delta_\bullet^2(\mathbf{D}_2,\mathbf{D}_4)$$

Combining this observation with (12) and the fact that $\delta_\bullet(.,.)$ are invariant to unitary affine transformations yields

$$\begin{aligned}\delta_\bullet^*(\mathbf{X}_\sigma, \mathbf{Y}_\sigma) &= \delta_\bullet^2(\mathbf{M}_X, \mathbf{M}_Y) + \delta_\bullet^2(\sigma, y+\sigma)\\ &\geq \delta_\bullet^2(\sigma, y+\sigma)\end{aligned} \tag{13}$$

The proof follows from the fact that since $y > 0$, $\delta_\bullet^2(\sigma, y+\sigma) \to \infty$ as $\sigma \to 0$. □

Corollary 1. *Consider the $n \times n$ Gram matrices $\mathbf{G}^{(1)}$ and $\mathbf{G}^{(2)}$ from two sequences $\{\mathbf{y}_t^{(1)}\}_{t=1}^{2n-1}$ and $\{\mathbf{y}_t^{(2)}\}_{t=1}^{2n-1}$, each generated by an (n-1)th order AR model defined by parameter vectors $\mathbf{r}^{(1)} = [a_1^{(1)}..a_n^{(1)}]$ and $\mathbf{r}^{(2)} = [a_1^{(2)}..a_n^{(2)}]$. Then the two dynamics are the same (e.g. $\mathbf{r}^{(1)} = \mathbf{r}^{(2)}$) if and only if $\lim_{\sigma\to 0}\delta_\bullet(\mathbf{G}_\sigma^{(1)}, \mathbf{G}_\sigma^{(2)}) < \infty$*

Proof. Follows from Theorem 1 by noting that $\mathcal{N}(\mathbf{G}^{(i)}) = \text{span}(\begin{bmatrix}(\mathbf{r}^{(i)})^\top & -1\end{bmatrix}^\top)$. □

Remark 1. *Corollary 1 justifies using the functions $\delta_\bullet(.,.)$ to compare PSD (rather than PD) Gram matrices and to use the corresponding mean:*

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \sum_{i=1}^{N} \delta_\bullet^2(\mathbf{X}, \mathbf{X}_i)$$

to define the "center" of a cluster of matrices from the same dynamics. However, from a practical standpoint, taking the limit as $\sigma \to 0$, can lead to numerical problems. Thus, in the sequel, we will use finite, but non-zero values of σ. Nevertheless, as long as this regularization value is smaller than the smallest non-zero singular value of $\mathbf{G}^{(i)}$, a reasoning similar to the one in the Theorem above shows that $\delta_\bullet(.,.)$ will yield very large values for matrices belonging to different dynamics.

The reasoning above allows for recasting the problem of comparing temporal sequences into the problem of comparing Gram matrices on the PD Riemannian manifold. The full procedure is illustrated in Figure 2.

5. 3D Action Recognition Application

Johansson [18] showed that a temporal sequence of the skeleton joints is good enough to capture an amazing diversity of human actions. Inspiblack by these results, the increasing availability of equipment such as the Kinect to capture 3D data, and the capability of estimating body parts from depth maps [33], researchers have proposed working with 3D skeleton joints data to recognize human actions.

Next, we present experiments to evaluate the effectiveness and efficiency of the proposed approach when comparing sequences to recognize human actions from (noisy) temporal sequences of their 3D joints. As described in detail below, we used Gram matrix embeddings with the JBLD (G-J), LERM (G-L), AIRM (G-A), and KLDM (G-K) metrics on three standard datasets: MSR-Action3D, MHAD and UTKinect, and compablack the performance against the 3D action recognition state-of-art methods.

For all datasets which will be introduced below, we performed basic pre-processing on joint locations as in [36], i.e., projecting the joints from the world coordinate system to a person centric coordinate system which puts its origin at the hip joint, scaling and rotating skeletons to make each of them scale and view invariant. We did not do sequence interpolation and Dynamic Time Warping (DTW) as [36] did. There are two reasons for this: one is that these two procedures are computationally expensive; the other is that our method inherently deals with sequences of different length. That is why running the code of [36] took many hours while the code of the proposed method ran for only several minutes.

Since hip joints were normalized as the origins, their coordinates are always zero and were discarded. Instead of using the absolute joint positions directly, we used the velocities. This is because the bias of the temporal sequences has a negative effect on the distance measurements. For each frame, we concatenated the $K-1$ joints into a $3(K-1)$-dimensional column vector. Then, a Hankel matrix was built for each sequence according to Equation 2.

There are two parameters: regularization number σ and Hankel block row size r. In principle σ should be as small as possible. However, too small σ may cause numeric problem and lead to lower accuracy. From our experience, a range from 10^{-4} to 10^{-2} is good enough for all metrics consideblack in this paper. We have observed better performance with larger block row size r. However, it is constrained by higher computation cost and the length of the shortest sequence. For example, the shortest sequence in UTKinect dataset has just 4 frames, so the largest r we can have is 4.

During training, based on a selected metric, we computed the geometric mean of all the Gram matrices for each class in the training set. During testing, a query sequence was first converted to a Gram matrix. Then, distances between the query Gram matrix and the trained class means were computed using the same metric. Finally, the query sequence is labeled with the class label of its closest mean.

5.1. MSR Action3D dataset Experiments

The MSR Action3D dataset [22] contains skeleton joints 3D coordinates, which are used to classify actions. It has 20 actions. Each action was performed by 10 subjects. For each subject, 20 joint locations were recorded. The main challenges of this data set include noisiness of the trajectories and similarity across actions. The original dataset has a total of 567 sequences. However, following [39], 10 of these sequences were discarded due to having too much noise. There are three different experimental protocols associated with this dataset.

The first protocol [22] divides the data into three subsets: AS1, AS2 and AS3. Each subset has eight actions which are similar to each other in some sense. The classification is performed on the three subsets separately, and the averaged accuracy is the final accuracy on the whole dataset. As in [22], we divided the 10 subjects into half training set and half testing set and run our classification algorithm 10 times with different splittings. Table 2 shows experiment results of our methods with different flavors. All distance-like functions performed well closely. To some extent, larger r brought higher accuracy at the cost of higher computation. When $r = 9$, the accuracy of G-J is higher than the state-of-the-art method by 2.13%. The run time of our methods

Table 2: Recognition accuracy (%) on MSR-Action3D dataset with $\sigma = 0.01$. r stands for Hankel matrix block row number. AS1, AS2 and AS3 stands for Action Set 1,2 and 3, respectly; Avg stands for average of the three sets results; Prep., Train and Test columns show the pre-processing, training and testing time for the whole dataset, respectively.

Method	AS1 (%)	AS2 (%)	AS3 (%)	Avg (%)	Prep.	Train	Test	Total Time
Subspace Angle [5]	56.86	57.59	75.63	72.21	5.0s	13.2h	3063.7s	14.1h
Covariance [17]	88.04	89.29	86.96	90.53	120.7s	114.9s	13.2s	248.8s
RF [42]	-	-	-	90.90	-	-	-	-
HOD [14]	92.39	90.18	91.43	91.26	213.0s	8.0s	4.4s	225.4s
Lie group [36]	95.29	83.87	98.22	92.46	-	-	-	$>$ 6h
DHMM-SL [32]	90.29	95.15	93.29	92.91	-	-	-	-
RF+depth [42]	94.88	87.00	100	94.30	-	-	-	-
HBRNN-L [11]	93.33	94.64	95.50	94.49	-	-	-	-
JAS+HOG2 [28]	-	-	-	94.84	-	-	-	-
Hankelet [21] ($r = 1$)	80.73	64.25	89.92	78.30	1.4s	9.0s	0.6s	11.1s
Hankelet [21] ($r = 3$)	83.74	72.14	94.38	83.42	1.5s	63.9s	4.3s	69.8s
Hankelet [21] ($r = 5$)	82.45	79.81	92.50	84.92	1.6s	214.4s	14.5s	230.6s
Hankelet [21] ($r = 9$)	82.43	80.94	92.24	85.20	4.2s	716.5s	48.5s	769.3s
Hankelet [21] ($r = 12$)	79.14	80.39	90.19	83.24	6.2s	1333.2s	90.3s	1429.7s
G-A ($r = 1$)	94.54	77.68	96.17	89.47	1.4s	76.5s	7.3s	85.3s
G-A ($r = 3$)	97.89	88.81	97.60	94.77	1.6s	615.4s	50.3s	667.4s
G-A ($r = 5$)	98.75	92.44	97.78	96.32	2.2s	1813.7s	130.7s	1946.7s
G-A ($r = 9$)	98.74	93.94	97.95	96.88	5.7s	1.9h	460.8s	2.0h
G-A ($r = 12$)	98.44	94.56	97.86	**96.96**	8.0s	3.3h	735.8s	3.5h
G-J ($r = 1$)	94.45	77.95	96.79	89.73	1.4s	13.1s	2.4s	16.5s
G-J ($r = 3$)	97.79	90.06	97.60	95.15	1.4s	105.1s	11.4s	117.9s
G-J ($r = 5$)	98.17	92.43	97.87	96.16	1.9s	269.7s	31.1s	302.8s
G-J ($r = 9$)	98.66	94.11	98.13	**96.97**	4.8s	1009.8s	117.5s	1132.1s
G-J ($r = 12$)	98.46	94.20	98.13	96.93	8.3s	1965.9s	247.5s	2221.9s
G-L ($r = 1$)	94.53	75.99	95.99	88.84	1.4s	1.3s	19.4s	22.2s
G-L ($r = 3$)	97.90	88.53	97.60	94.67	1.5s	7.7s	135.4s	144.6s
G-L ($r = 5$)	98.46	92.00	97.60	96.02	1.7s	22.5s	390.6s	414.9s
G-L ($r = 9$)	98.36	93.42	97.77	**96.52**	5.1s	93.7s	1443.3s	1542.2s
G-L ($r = 12$)	97.87	93.94	97.59	96.47	8.2s	183.1s	2676.2s	2867.8s
G-K ($r = 1$)	93.86	72.64	96.08	87.53	1.4s	0.7s	4.1s	6.2s
G-K ($r = 3$)	97.61	87.76	97.69	94.35	1.5s	4.8s	26.1s	32.5s
G-K ($r = 5$)	97.59	91.50	97.78	95.62	1.7s	13.6s	89.8s	105.3s
G-K ($r = 9$)	96.92	93.69	97.33	**95.98**	5.7s	49.4s	365.4s	420.6s
G-K ($r = 12$)	95.96	94.66	96.90	95.80	8.9s	89.3s	741.0s	839.4s

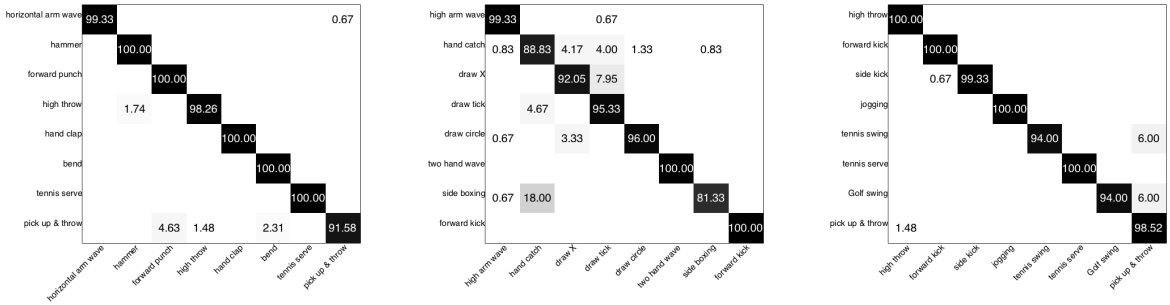

Figure 3: Recognition confusion matrices of MSR Action3D dataset. (Left: AS1; Ctr.: AS2; Right: AS3)

Table 3: Recognition accuracy on MSR-Action3D dataset following protocol of [36], $r = 12, \sigma = 0.01$. Prep., Train and Test columns show the pre-processing, training and testing time for the whole dataset, respectively.

Method	Acc. (%)	Prep.	Train	Test	Total Time
Occupancy[38]	86.50	-	-	-	-
Actionlets[39]	88.20	-	-	-	-
HON4D[29]	88.89	-	-	-	-
H-HMM[31]	89.01	-	-	-	-
DHMM-SL[32]	89.23	0.8s	10.5h	8.6s	10.5h
Lie[36]	89.48	-	-	-	>6h
Pose1[37]	90.22	-	-	-	-
Pose2[12]	91.50	-	-	-	-
Pose3[41]	91.07	-	-	-	-
Traj.Shape[10]	92.10	-	-	-	-
G-K	**93.68**	7.3s	**69s**	1452s	**1529s**
G-L	**94.38**	7.6s	154s	1.6h	1.7h
G-J	**94.71**	7.3s	1513s	**506s**	2026s
G-A	**94.74**	7.2s	2.7h	1523s	3.1h

Table 4: Recognition accuracy on MSR-Action3D dataset following protocol of [29], $r = 9, \sigma = 0.01$; Prep., Train and Test columns show the pre-processing, training and testing time for the whole dataset, respectively.

Method	Acc. (%)	Prep.	Train	Test	Total time
HON4D[29]	82.15±4.18	-	-	-	-
elastic[1]	85.16±3.13	-	-	-	-
mot.Traj.[10]	87.28±2.99	-	-	-	-
G-K	**88.80±2.75**	4.9s	962s	5.2h	5.4h
G-L	**90.07±2.51**	5.9s	2777s	30.7h	31.4h
G-J	**90.16±2.89**	5.8s	13.3h	2.7h	15.9h
G-A	**90.35±2.66**	5.8s	23.8h	4.9h	28.7h

is also shown. We observed that G-K is fastest in total time and in training, and G-J is the fastest in testing.

In the second protocol [36, 39], subjects are randomly divided into halves for 10 times. Each time 5 subjects are used for training and the rest subjects for testing. The experiment results using this protocol are shown in Table 3. The confusion matrix is shown in Figure 4. The proposed method using all four metrics performed better than the state-of-the-art methods and the best improvement is 2.64%. The last protocol [29] also uses half of the subjects for training and the rest for testing. Instead of picking randomly, the authors experimented on all possible 252 splits and reported the average accuracy and the standard deviation. The results following this protocol are shown in Table 4. Again, the proposed method performed the best.

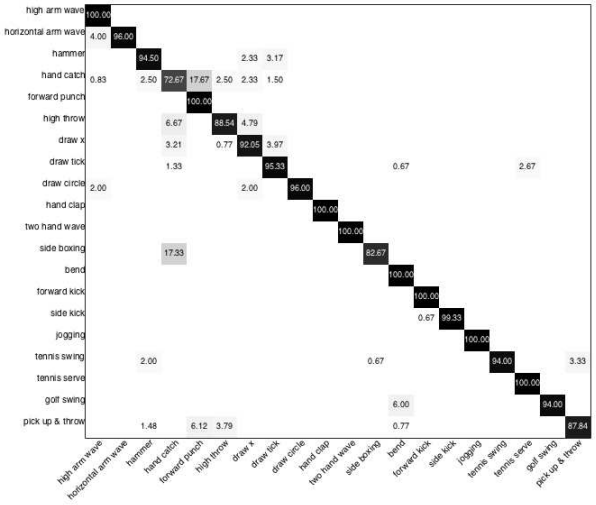

Figure 4: Recognition confusion matrix of MSR-Action3D dataset following protocol of [36]

5.2. MHAD dataset Experiments

The Berkley MHAD dataset contains 11 actions in 659 sequences performed by 12 subjects. Each skeleton has 35 joints and the sequences are 480 frames per second. Following the protocol of [26], we used the first seven subjects for training and the last 5 for testing. Table 5 compablack recognition accuracy and running time for the proposed method against the state-of-the-art methods. The parameters were set as $r = 5, \sigma = 0.0001$ for all methods. With G-K, the performance is 100% accuracy in only 52 seconds. The results show that the proposed method is both effective and efficient.

Table 5: Recognition accuracy on MHAD dataset, $r = 5, \sigma = 0.0001$; Prep., Train and Test columns show the pre-processing, training and testing time for the whole dataset.

Method	Acc. (%)	Pre	Train	Test	Total time
SMIJ [27]	95.37	-	-	-	-
RBF Net [35]	97.58	-	-	-	-
Dynemes [19]	98.18	-	-	-	-
Bio-LDS [7]	**100**	-	-	-	-
HBRNN-L [11]	**100**	-	-	-	-
G-L	97.45	5.1s	10.2s	169.1s	185.5s
G-J	97.45	5.1s	134.1s	**13.1s**	153.8s
G-A	98.18	5.1s	1044.9s	49.6s	1099.8s
G-K	**100**	5.1s	**3.9s**	41.3s	**51.8s**

Table 6: Classwise recognition accuracy (%) and running time (in seconds) for the UTKinect dataset. Performance is compablack against the Hankelets subspace angles [21], Hankelet-based HMM [31], 3D joints [40], and Space-time pose [9] methods. Prep., Train and Test columns show the pre-processing, training and testing time for the whole dataset, respectively.

	Walk	S.Dwn	S.Up	P.Up	Carry	Throw	Push	Pull	Wave	Clap	Avg	Prep.	Train	Test	Total Time
[21]	60	40	75	80	75	70	80	85	70	85	71.9	0.4s	280s	0.7s	281s
[31]	63.2	100	100	100	83.3	61.1	90	100	85	85	86.8	-	-	-	-
[40]	96.5	91.5	93.5	97.5	97.5	59.0	81.5	92.5	100	100	90.9	-	-	-	-
[9]	90	100	100	100	68.4	95	90	100	100	80	91.5	-	-	-	-
G-J	100	85	100	100	100	100	100	100	100	100	98.5	0.5s	192s	1.7s	194s
G-L	100	85	100	100	100	100	100	100	100	100	98.5	0.5s	14.7s	17.1s	32s
G-A	100	100	100	100	100	100	100	100	100	100	**100**	0.5s	1365s	5.9s	1371s
G-K	100	100	100	100	100	100	100	100	100	100	**100**	0.5s	8.0s	4.4s	**13s**

5.3. UTKinect dataset Experiments

The UTKinect-Action dataset [40] is an action classification dataset based on 3D skeleton joints positions. It contains 10 actions: walk, sit down, stand up, pick up, carry, throw, push, pull, wave hands and clap hands. Each action was performed by 10 subjects, for two instances. In total, it has 199 sequences (one instance of a subject is missing).

To evaluate our results we followed the leave-one-out-cross-validation (LOOCV) protocol which was proposed by the original paper which provided the dataset [40]. Table 6 shows the accuracy performance and the time taken using the methods from [21, 31, 40, 9] and the proposed approach.

We can see that the proposed approach achieves the best performance. In particular, using KLDM achieved 100% accuracy in only 13 seconds. Note that [42] and [36] also reported high accuracy with this dataset. However, they used a different protocol, so we did not include them in Table 6.

5.4. HDM05 dataset Experiments

The HDM05 dataset [25] is a motion capture dataset which contains 3D locations of 31 skeleton joints of human subjects. We applied our methods on this dataset using the protocol in [16], which included 14 different actions and used 4 joints corresponding to arms and legs. In the classification setup, 4 out of the 5 subjects were used for training and the remaining one for testing. Again, all proposed methods outperform existing ones.

Table 7: Experiment on HDM05 dataset using the protocol in [14]. In the proposed methods we used the parameters $r = 5, \sigma = 0.01$.

Methods	Acc (%)	Prep.	Train	Test	Total time
S_H-SVM [16]	73.3 ± 11.4	-	-	-	-
R-VLAD [13]	79.1 ± 7.5	-	-	-	-
G-A	$\mathbf{87.0 \pm 4.7}$	5.6s	29.7s	1.1s	36.4s
G-J	$\mathbf{87.3 \pm 4.3}$	5.6s	4.7s	0.3s	10.7s
G-L	$\mathbf{88.0 \pm 6.3}$	5.6s	0.9s	5.5s	12.0s
G-K	$\mathbf{86.3 \pm 5.6}$	5.6s	0.3s	0.6s	6.5s

6. Conclusions

Temporal sequences are ubiquitous in computer vision and are a rich source of information that can be used for a wide range of applications ranging from tracking to action recognition to event detection. However, effectively tapping this information requires having suitable inference tools to compare, cluster and classify temporal sequences. Inspiblack by recent results in activity recognition and advances in computing distance-like function on the Positive Definite manifold, we proposed a new framework to perform temporal inferencing. The main idea of the proposed approach is to first represent the data using regularized Gram matrices derived from their Hankel matrices and then using some metric to compare/classify them on the PD manifold. We illustrated the benefits of this framework by classifying real 3D joint data for human action recognition. Our experiments showed that this simple approach gives competitive or better than state-of-art results for the problem of human action recognition using 3D joints data. Moreover, consistent numerical experience shows that these results are largely independent of the actual metric used, indicating that these advantages stem from embedding the data in the PD manifold and exploiting its structure.

References

[1] R. Anirudh, P. Turaga, J. Su, and A. Srivastava. Elastic functional coding of human actions: From vector-fields to latent variables. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3147–3155, 2015. 7

[2] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Log-euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic resonance in medicine*, 56(2):411–421, 2006. 2, 3

[3] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.

[4] R. Bhatia. *Positive definite matrices*. Princeton University Press, 2009. 2, 3

[5] A. Björck and G. H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of computation*, 27(123):579–594, 1973. 2, 6

[6] L. Breiman. Hinging hyperplanes for regression, classification and function approximation. *IEEE Trans. Inf. Theory*, pages 999–1013, 1993. 3

[7] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal. Bio-inspired dynamic 3d discriminative skeletal features for human action recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 471–478. IEEE, 2013. 7

[8] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos. Jensen-bregman logdet divergence with application to efficient similarity search for covariance matrices. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(9):2161–2174, 2013. 2, 3

[9] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo. Space-time pose representation for 3d human action recognition. In *New Trends in Image Analysis and Processing–ICIAP 2013*, pages 456–464. Springer, 2013. 8

[10] 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. 2014. 7

[11] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1110–1118, 2015. 6, 7

[12] A. Eweiwi, M. S. Cheema, C. Bauckhage, and J. Gall. Efficient pose-based action recognition. In *Computer Vision–ACCV 2014*, pages 428–443. Springer, 2015. 7

[13] M. Faraki, M. T. Harandi, and F. Porikli. More about vlad: A leap from euclidean to riemannian manifolds. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2, 8

[14] M. Gowayyed, M. Torki, M. Hussein, and M. El-Saban. Histogram of oriented displacements (hod): Describing trajectories of human joints for action recognition. In *International Joint Conference on Artificial Intelligence (IJCAI)*, Beijing, China, August 2013. 6

[15] M. Harandi, Salzmann, and R. Hartley. From manifold to manifold: Geometry-aware dimensionality reduction for spd matrices. *arXiv preprint arXiv:1407.1120*, 2014. 4

[16] M. Harandi, M. Salzmann, and F. Porikli. Bregman divergences for infinite dimensional covariance matrices. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1003–1010. IEEE, 2014. 2, 4, 8

[17] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2466–2472. AAAI Press, 2013. 2, 6

[18] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2):201–211, 1973. 5

[19] I. Kapsouras and N. Nikolaidis. Action recognition on motion capture data using a dynemes and forward differences representation. *Journal of Visual Communication and Image Representation*, 25(6):1432–1445, 2014. 7

[20] B. Li, M. Ayazoglu, T. Mao, O. I. Camps, and M. Sznaier. Activity recognition using dynamic subspace angles. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3193–3200. IEEE, 2011. 2

[21] B. Li, O. I. Camps, and M. Sznaier. Cross-view activity recognition using hankelets. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1362–1369. IEEE, 2012. 2, 3, 6, 8

[22] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 9–14. IEEE, 2010. 5

[23] M. Moakher and P. G. Batchelor. Symmetric positive-definite matrices: From geometry to applications and visualization. In *Visualization and Processing of Tensor Fields*, pages 285–298. Springer, 2006. 2, 3

[24] M. Müller. *Information retrieval for music and motion*, volume 2. Springer, 2007. 1

[25] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. Documentation mocap database hdm05. 2007. 8

[26] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Berkeley mhad: A comprehensive multimodal human action database. In *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pages 53–60. IEEE, 2013. 7

[27] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Sequence of the most informative joints (smij): A new representation for human skeletal action recognition. *Journal of Visual Communication and Image Representation*, 25(1):24–38, 2014. 7

[28] E. Ohn-Bar and M. M. Trivedi. Joint angles similarities and hog2 for action recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 465–470. IEEE, 2013. 6

[29] O. Oreifej and Z. Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 716–723. IEEE, 2013. 7

[30] X. Pennec, P. Fillard, and N. Ayache. A riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006. 2, 3

[31] L. L. Presti, M. La Cascia, S. Sclaroff, and O. Camps. Gesture modeling by hanklet-based hidden markov model. In *Computer Vision–ACCV 2014*, pages 529–546. Springer, 2015. 7, 8

[32] L. L. Presti, M. La Cascia, S. Sclaroff, and O. Camps. Hankelet-based dynamical systems modeling for 3d action recognition. *Image and Vision Computing*, 2015. 6, 7

[33] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013. 5

[34] S. Sra. Positive definite matrices and the symmetric stein divergence. Technical report, 2011. 3

[35] S. Vantigodi and V. B. Radhakrishnan. Action recognition from motion capture data using meta-cognitive rbf network classifier. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pages 1–6. IEEE, 2014. 7

[36] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 588–595. IEEE, 2014. 5, 6, 7, 8

[37] C. Wang, Y. Wang, and A. L. Yuille. An approach to pose-based action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 915–922. IEEE, 2013. 7

[38] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3d action recognition with random occupancy patterns. In *Computer vision–ECCV 2012*, pages 872–885. Springer, 2012. 7

[39] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1290–1297. IEEE, 2012. 1, 5, 7

[40] L. Xia, C.-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. 8

[41] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2752–2759. IEEE, 2013. 7

[42] Y. Zhu, W. Chen, and G. Guo. Fusing spatiotemporal features and joints for 3d action recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 486–491. IEEE, 2013. 6, 8