Coding Kendall's Shape Trajectories for 3D Action Recognition

Amor Ben Tanfous, Hassen Drira, Boulbaba Ben Amor

Coding Kendall's Shape Trajectories for 3D Action Recognition

Amor Ben Tanfous Hassen Drira Boulbaba Ben Amor
IMT Lille Douai, Univ. Lille, CNRS, UMR 9189 – CRIStAL –
Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille, France
omar.bentanfous@imt-lille-douai.fr

Abstract

*Suitable shape representations as well as their temporal evolution, termed trajectories, often lie to non-linear manifolds. This puts an additional constraint (*i.e.*, non-linearity) in using conventional machine learning techniques for the purpose of classification, event detection, prediction, etc. This paper accommodates the well-known Sparse Coding and Dictionary Learning to the Kendall's shape space and illustrates effective coding of 3D skeletal sequences for action recognition. Grounding on the Riemannian geometry of the shape space, an intrinsic sparse coding and dictionary learning formulation is proposed for static skeletal shapes to overcome the inherent non-linearity of the manifold. As a main result, initial trajectories give rise to sparse code functions with suitable computational properties, including sparsity and vector space representation. To achieve action recognition, two different classification schemes were adopted. A bi-directional LSTM is directly performed on sparse code functions, while a linear SVM is applied after representing sparse code functions using Fourier temporal pyramid. Experiments conducted on three publicly available datasets show the superiority of the proposed approach compared to existing Riemannian representations and its competitiveness with respect to other recently-proposed approaches. When the benefits of invariance are maintained from the Kendall's shape representation, our approach not only overcomes the problem of non-linearity but also yields to discriminative sparse code functions.*

1. Introduction

The availability of cost-effective and real-time human body skeletal data estimation solutions [33] has pushed researchers to study their shape as well as their temporal evolution. In particular, the problem of action recognition from 3D skeletons has received a particular attention with the availability of several datasets and end-users applications as gaming, Human Machine Interaction, and physical performance assessment, to cite a few. However, human actions observed from visual sensors are often subject to view variations. Considering this problem, an efficient way for analyzing 3D skeleton motions takes into account view-invariance properties, giving rise to shape representations often lying to non-linear shape spaces [2, 3, 25]. For instance, Kendall [25] defines the shape as the geometric information that remains when location, scale, and rotational effects are filtered out from an object. Accordingly, we represent 3D skeletons as points in the Kendall's shape space, considering skeletal sequences as *trajectories* [2]. However, inferencing such a shape representation remains a challenging problem due to the *non-linearity* of the manifold of interest. For example, the use of standard data coding (*e.g.*, sparse coding, PCA, etc.) and machine learning techniques (*e.g.*, dictionary learning, SVM, deep learning, etc.) is not straightforward. The problem is even more acute with the introduction of the temporal dimension, *i.e.*, analyzing trajectories in Kendall's shape space. In the literature, two alternatives have been proposed to overcome these problems for different Riemannian manifolds – they are either *Extrinsic* [17, 20, 23, 27] or *Intrinsic* [4, 5, 21, 22]. When the first family is based on the embedding into high dimensional Hilbert spaces, the second maps points on the manifold to a fixed tangent space attached to the manifold at a *reference point*. In the second family, the main issue is that distortions are introduced when points are not close to the reference point [1, 2, 35]. In this work, we propose an intrinsic solution to overcome the problem of non-linearity of the Kendall's shape space while avoiding mapping points to a fixed tangent space at a reference point.

Motivated by the success of sparse representations in several recognition tasks [6, 11, 17, 21], we propose to represent human actions using an intrinsic formulation of sparse coding and dictionary learning of skeletal shapes in the Kendall's shape space. Specifically, a 3D skeletal shape is coded on its attached tangent space where the dictionary of shapes is mapped. Hence, for each trajectory, this representation gives rise to a function of sparse codes lying in Euclidean space. By doing so, we handle the problem of non-linearity of the manifold without mapping points to a

reference tangent space. Furthermore, we propose to learn a dictionary grounding on the Riemannian geometry of the Kendall's shape space, with a novel initialization step allowing an automatic inference on the number of atoms. In the context of action recognition, our approach brings two main advantages: (1) Sparse coding of skeletal shapes is performed with respect to a Riemannian dictionary. Hence, the resulting sparse code functions are expected to be more discriminative than the data themselves [21]; (2) Using sequences of sparse codes as discriminative features allows us to perform classification in vector space, avoiding the more difficult task of classification on the manifold. The **contributions of this work** are: 1) A novel human actions representation based on an intrinsic sparse coding of skeletal shape trajectories on the Kendall's shape space. This allows to map skeletal trajectories from a non-linear space to sparse time-series in Euclidean space. 2) The dictionary of shapes is learned with respect to the geometry of the manifold and is preceded by a novel initialization step based on Bayesian clustering of shapes and principal geodesic analysis, to automatically infer on the number of atoms. 3) Classification of the sparse time-series using two different classification schemes. Experiments are conducted on three commonly-used datasets to show the competitiveness of the proposed approach in the context of 3D action recognition.

The rest of the paper is organized as follows. In section 2, we briefly review existing solutions of sparse coding and dictionary learning in non-linear manifolds, in addition to recent achievements in 3D action recognition using skeletal data, with a particular focus on Riemannian approaches. Section 3 introduces the sparse coding and dictionary learning method with a review of the geometric properties of the Kendall's shape space. In section 4, we describe the adopted temporal modeling and classification pipelines. Experimental results and discussions are reported in section 5, and section 6 concludes the paper.

2. Prior Work

In this section, we firstly focus on the extension of sparse coding and dictionary learning (SCDL) to non-linear Riemannian manifolds. Then, we briefly review some recent works in action recognition using 3D skeletal data.

2.1. SCDL on Riemannian manifolds

Sparse representations have proved to be successful in various computer vision tasks [11] which explains the significant interest in the last decade [6, 17, 21]. Based on a learned dictionary, each data point can be represented as a *linear combination* of a few dictionary elements (atoms), so that a squared Euclidean loss is minimized. This assumes that the data points as well as the dictionary atoms are defined in vector space (to allow speaking on linear combination). However, most suitable image features often lie to non-linear manifolds [30]. Thus, to sparsely code these data while exploiting the invariance properties of Riemannian manifolds, the classical problem of SCDL needs to be extended to its non-linear counterpart. Previous works addressed this problem [6, 16, 17, 18, 21, 27, 46]. For instance, a straightforward solution was proposed in [16, 43] by embedding the manifolds of interest into Euclidean space via a fixed tangent space at a reference point. However, this solution does not take advantage of the entire Riemannian structure as in this tangent space, only distances to the reference point are equal to true geodesic distances. To overcome this problem, Ho *et al*. [21] proposed a general framework for SCDL in Riemannian manifolds by working on the tangent bundle. Here, each point is coded on its attached tangent space into which the atoms are mapped. By doing so, only distances to the tangent point are needed. Their proposed dictionary learning method includes an iterative update of the atoms using a gradient descent approach along geodesics. This general solution essentially relies on mappings to tangent spaces using the logarithm map operator. Although it is well defined for several manifolds, analytic formulation of the logarithm map is not available or difficult to compute for others. Therefore, some studies [17, 18, 20, 27] proposed to embed the Riemannian manifold into a Reproducing Kernel Hilbert Space (RKHS). These are Euclidean spaces where linear SCDL becomes possible. Recently, Harandi *et al*. [17] proposed to map the Grassmann manifolds into the space of symmetric matrices to overcome the latter problem and preserve several properties of the Grassmann structure. They also proposed kernelized versions of the SCDL algorithms to handle the non-linearity of the data, similarly proposed in [19] for Symmetric Positive Definite matrices.

2.2. Action recognition from 3D skeletal sequences

Several recent approaches include the use of temporal state-space model to classify action sequences without any manifold assumptions on the data representation. Considering a human action as transitions between body poses over time, G. Hernando *et al*. [13] proposed a forest-based classifier called transition forests to discriminate both static pose information and temporal transitions between pairs of two independent frames. Another work [40] modeled a human action as a set of semantic parts called *motionlets* obtained by tracking then segmenting the trajectory of each joint. By combining the *motionlets* and their spatio-temporal correlations, they proposed an undirected complete labeled graph to represent a video, and a subgraph-pattern graph kernel to measure the similarity between graphs, then to classify videos. More recently, two kernel-based tensor representations named sequence compatibility kernel (SCK) and dynamics compatibility kernel (DCK) were introduced in [26]. These can capture the higher-order relationships between

the joints. The first captures the spatio-temporal compatibility of joints between two sequences, while the second models a sequence dynamics as the spatio-temporal co-occurrences of the joints. Tensors are then formed from these kernels to train SVM. On the other hand, recurrent neural networks (RNNs) have showed promising performance when applied to 3D action recognition. For instance, HBRNN-L [10] applied bidirectional RNNs hierarchically by dividing a skeleton into five parts of neighboring joints. Then, each is separately fed into a bidirectional RNN before fusing their outputs to form the upper-body and the lower-body. Similarly, these latter were fed into different RNNs and their outputs fusion form the global body representation. More recently, the spatio-temporal LSTM (ST-LSTM) [29] extended LSTM to spatio-temporal domains. To this end, the analysis of a 3D skeleton joint considers spatial information from neighboring joints and temporal information from previous frames. In addition, a tree-structure based method allows to better describe the adjacency properties among the joints. This method is further improved by a gating mechanism to handle noise and occlusion.

Other approaches exploited some basics of the Riemannian geometry to analyze skeletal sequences. In [35], the authors proposed to represent skeletal motions as trajectories in the Special Euclidean (Lie) group $SE(3)^n$ (respectively $SO(3)^n$). These representations are then mapped into the correspondent Lie algebra $\mathfrak{se}(3)^n$ (respectively $\mathfrak{so}(3)^n$) which is a vector space, the tangent space attached to the Lie group at the identity, where they are processed and classified. Exploiting the same representation on Lie Groups, Anirudh *et al*. [1] used the framework of Transported Square-Root Velocity Fields (TSRVF) [34] to encode trajectories lying on Lie groups. They extended existing coding methods such as PCA, KSVD, and Label Consistent KSVD to these Riemannian trajectories. Another approach [2] proposed a different solution by extending the Kendall's shape theory to trajectories. Accordingly, translation, rotation, and global scaling are first filtered out from each skeleton to quantify the shape. Then based on the TSRVF, they defined an elastic metric to jointly align and compare trajectories. Here, trajectories are transported to a reference tangent space attached to the Kendall's shape space at a fixed point. A common major drawback of these approaches is mapping trajectories to a reference tangent space which may introduce distortions. Conscious of this limitation, the authors in [36] proposed a mapping of trajectories on Lie groups combining the usual logarithm map with a rolling map that guarantees a better flattening of trajectories on Lie groups. In our work, we represent the motion of skeletal shapes as trajectories in the Kendall's shape space, as in [2]. To overcome the problem of non-linearity of the manifold, we propose to code trajectories using an intrinsic formulation of SCDL that avoids distortions caused by tangent space approximations.

3. Coding Kendall's skeletal shapes

We propose to adapt a general intrinsic formulation of Riemannian SCDL to the case of Kendall's shape space. This allows to represent a 3D skeletal shape lying on Kendall's space as a sparse vector encoded with respect to a dictionary of shapes. In what follows, we start by briefly reviewing the geometry of the manifold of interest. Then, we describe the SCDL framework.

3.1. Geometry of the Kendall's shape space

A skeleton is represented using a finite number of salient points or landmarks (points in $\mathbb{R}^3$). To quantify skeletal shapes, Kendall [25] proposed to establish equivalences with respect to shape invariant transformations that are translations, rotations, and global scaling of configurations. Let $Z \in \mathbb{R}^{n\times 3}$ represent a skeleton, *i.e*., a configuration of n landmarks in $\mathbb{R}^3$. To remove the translation variability, we follow [8] and introduce the notion of Helmert sub-matrix, a $(n-1)\times n$ sub-matrix of a commonly used Helmert matrix, to perform centering of configurations. For any $Z \in \mathbb{R}^{n\times 3}$, the product $HZ \in \mathbb{R}^{(n-1)\times 3}$ represents the Euclidean coordinates of the centered configuration. Let $\mathcal{C}_0$ be the set of all such centered configurations of n landmarks in $\mathbb{R}^3$, *i.e*., $\mathcal{C}_0 = \{HZ \in \mathbb{R}^{(n-1)\times 3} | Z \in \mathbb{R}^{n\times 3}\}$. $\mathcal{C}_0$ is a $3(n-1)$ dimensional vector space and can be identified with $\mathbb{R}^{3(n-1)}$. To remove the scale variability, we define the pre-shape space to be: $\mathcal{C} = \{Z \in \mathcal{C}_0 | \|Z\|_F = 1\}$; $\mathcal{C}$ is a unit sphere in $\mathbb{R}^{3(n-1)}$ and, thus, is $(3n-4)$ dimensional. The tangent space at any pre-shape Z is given by: $T_Z(\mathcal{C}) = \{V \in \mathcal{C}_0 | \text{trace}(V^T Z) = 0\}$. To remove the rotation variability, for any $Z \in \mathcal{C}$, we define an equivalence class: $\bar{Z} = \{ZO | O \in SO(3)\}$ that represents all rotations of a configuration Z. The set of all such equivalence classes, $\mathcal{S} = \{\bar{Z} | Z \in \mathcal{C}\} = \mathcal{C}/SO(3)$ is called the *shape space* of skeletons. The tangent space at any shape $\bar{Z}$ is $T_{\bar{Z}}(\mathcal{S}) = \{V \in \mathcal{C}_0 | \text{trace}(V^T Z) = 0, \ \ \text{trace}(V^T ZU) = 0\}$, where U is any 3×3 skew-symmetric matrix. The first condition makes V tangent to $\mathcal{C}$ and the second makes V perpendicular to the rotation orbit. Together, they force V to be tangent to the shape space $\mathcal{S}$. Assuming standard Riemannian metric on $\mathcal{S}$, the geodesic between two points $\bar{Z}_1, \bar{Z}_2 \in \mathcal{S}$ is defined as:

$$\alpha(t) = \frac{1}{\sin(\theta)}(\sin((1-t)\theta)Z_1 + \sin(t\theta)Z_2 O^*), \qquad (1)$$

where $\theta = \cos^{-1}(\langle Z, Z_2 O^* \rangle)$ and O^* is the optimal rotation that aligns Z_2 with Z_1: $O^* = argmin_{O\in SO(3)} \|Z_1 - Z_2 O\|_F^2$. This θ is also the geodesic distance between $\bar{Z}_1$ and $\bar{Z}_2$ in the shape space $\mathcal{S}$, representing the amount of the optimal deformation of $\bar{Z}_1$ into $\bar{Z}_2$. For $t = 0$, $\alpha(0) = \bar{Z}_1$ and for $t = 1$ we have $\alpha(1) = \bar{Z}_2$. Note that Kendall's

shape space is a complete Riemannian manifold such that $\log_{\bar{Z}}$ is defined for all $\bar{Z} \in \mathcal{S}$. As a consequence, the geodesic distance between two configurations $\bar{Z}_1$ and $\bar{Z}_2$ can be computed as $\mathbf{d}_{\mathcal{S}}(\bar{Z}_1, \bar{Z}_2) = \| \log_{\bar{Z}_1}(\bar{Z}_2) \|_{\bar{Z}_1}$, where $\|.\|_{\bar{Z}_1}$ denotes the norm induced by the Riemannian metric at $T_{\bar{Z}_1}(\mathcal{S})$. In view of the spherical structure of $\mathcal{C}$, analytic expressions of the exponential and logarithm maps are well defined [8, 25] and can be easily adapted to $\mathcal{S}$. In summary, we have analytical expressions for computing exponential map, logarithm map, and intrinsic mean [24] of shapes on $\mathcal{S}$. We refer the reader to [2] for definitions.

3.2. Sparse coding of skeletal shapes

In this section, we propose to adapt the Riemannian formulation of sparse coding proposed in [21] to the case of Kendall's shape space. To this end, we start by studying the formulation of the problem in Euclidean space.

In Euclidean space, let $\mathcal{D} = \{d_1, d_2, ..., d_N\}$ be a set of vectors in $\mathbb{R}^k$ denoting a given dictionary of N elements or atoms, and $z \in \mathbb{R}^k$ a query data point. The problem of sparse coding z with respect to $\mathcal{D}$ can be expressed as

$$l_E(z, \mathcal{D}) = \min_w \|z - \sum_{i=1}^{N} [w]_i \, d_i\|_2^2 + \lambda f(w), \qquad (2)$$

where $w \in \mathbb{R}^N$ denotes the vector of codes comprised of $\{[w]_i\}_{i=1}^N$, $f : \mathbb{R}^N \to \mathbb{R}$ is the sparsity inducing function defined as the ℓ_1 norm, and λ is the sparsity regularization parameter. Eq. (2) seeks to optimally approximate z (by $\hat{z}$) as a linear combination of atoms, *i.e.*, $\hat{z} = \sum_{i=1}^N [w]_i \, d_i$, while tacking into account a particular sparsity constraint on the codes, $f(w) = \|w\|_1$. This sparsity function has the role of forcing z to be represented as only a small number of atoms.

Moving to the case of Kendall's shape space, $\mathcal{D} = \{\bar{d}_1, \bar{d}_2, ..., \bar{d}_N\}$ is now a dictionary on $\mathcal{S}$, and similarly the query $\bar{Z}$ is a point on $\mathcal{S}$. Accordingly, the problem of sparse coding involves the geodesic distance defined on $\mathcal{S}$ and, thus, becomes

$$l_{\mathcal{S}}(\bar{Z}, \mathcal{D}) = \min_w (d_{\mathcal{S}}(\bar{Z}, C(\mathcal{D}, w))^2 + \lambda f(w)). \qquad (3)$$

Here, $C : \mathcal{S}^N \times \mathbb{R}^N \to \mathcal{S}$ denotes an encoding function that generates the approximated point $\hat{\bar{Z}}$ on $\mathcal{S}$ by combining atoms with codes. Note that in the special case of Euclidean space, $C(\mathcal{D}, w)$ would be a linear combination of atoms. However, in the Riemannian manifold $\mathcal{S}$, we have forsaken the structure of vector space which makes the linear combination of atoms lying on $\mathcal{S}$ no longer applicable, as the approximated $\hat{\bar{Z}}$ may lie out of the manifold. An interesting alternative is the intrinsic formulation of Eq. (3), when considering that $\mathcal{S}$ is a complete Riemannian manifold, thus, the geodesic distance $d_{\mathcal{S}}(\bar{Z}, \bar{d}) = \| \log_{\bar{Z}}(\bar{d}) \|_{\bar{Z}}$

Algorithm 1 Kendall Sparse Coding

Input: Dictionary $\mathcal{D} = \{\bar{d}_i\}_{i=1}^N$, $\bar{d}_i \in \mathcal{S}$; $\bar{Z} \in \mathcal{S}$ (query)
Output: Sparse codes vector $w^*$of the query $\bar{Z}$.

1: **for** $i = 1$ to N **do**
2: $\quad \mathcal{V}_i \leftarrow \log_{\bar{Z}}(\bar{d}_i)$ //Projection of $\mathcal{D}$ into $T_{\bar{Z}}(\mathcal{S})$
3: **end for**
4: $w^* = \text{argmin}_w \| \sum_{i=1}^N [w]_i \, \mathcal{V}_i \|_2^2 + \lambda f(w)$

(as explained in section 3.1). As a consequence, the cost function in (3) can be written as

$$l_{\mathcal{S}}(\bar{Z}, \mathcal{D}) = \min_w \| \sum_{i=1}^{N} [w]_i \log_{\bar{Z}}(\bar{d}_i) \|_{\bar{Z}}^2 + \lambda f(w), \qquad (4)$$

where $\log_{\bar{Z}}$ denotes the logarithm map operator that maps each atom $\bar{d} \in \mathcal{S}$ to the tangent space $T_{\bar{Z}}(\mathcal{S})$ at the point $\bar{Z}$ being coded, and $\|.\|_{\bar{Z}}$ is the norm induced by the Riemannian metric at $T_{\bar{Z}}(\mathcal{S})$. Mathematically, this allows to partially compensate the lack of vector space structure on $\mathcal{S}$, as illustrated in Fig. 1. To avoid the solution $w = 0$, we imposed in Eq. (4) an important additional affine constraint defined as $\sum_{i=1}^N [w]_i = 1$. In algorithm 1, we provide a summary of the SC approach on Kendall's shape space.

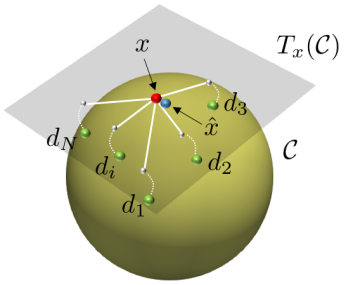

Figure 1. Pictorial of the sparse coding approach on the pre-shape space $\mathcal{C}$. The approximation of $x \in \mathcal{C}$ could be viewed as a weighted Karcher mean of the atoms of a dictionary $\mathcal{D} = \{d_i\}_{i=1}^N$.

3.3. Dictionary Learning on Kendall's Space

Learning a discriminative dictionary $\mathcal{D}$ typically yields accurate reconstruction of training samples and produces discriminative codes with the desired structure, *e.g.*, sparsity. In this section, we propose to learn $\mathcal{D}$ using the geometry of $\mathcal{S}$. Before describing our approach, it is important to note that the performance of $\mathcal{D}$ is sensitive to the number of atoms N. To the best of our knowledge, all previous methods opted for an empiric choice of N, which tends to be highly time consuming, especially when it comes to large datasets. As a solution, we propose an elegant initialization step enabling a fully automatic inference on N. Moreover, it remarkably accelerates the convergence of the dictionary learning algorithm, as illustrated in the right panel of Fig. 2.

Dictionary initialization – Given m training skeletons on $\mathcal{S}$, the idea is to select N relevant atoms to initialize the dictionary. This is done in two main steps: (1) Clustering of skeletal shapes; (2) Generating atoms from each cluster such that they well describe the intra-cluster variability. In the first step, we adapted the Bayesian clustering of shapes of curves method proposed in [45] to cluster skeletal shapes in the Kendall's shape space. Following [45], an inner product matrix is calculated from the data, *i.e.*, skeletal shapes in our case. Then, it is modeled using a Wishart distribution. To allow for an automatic inference on the number of clusters, prior distributions are assigned to the parameters of the Wishart distribution. Then, posterior is sampled using a Markov chain Monte Carlo procedure. For more details about the clustering algorithm, we refer the reader to [45]. At this point, we suppose having h clusters on $\mathcal{S}$. The next step is to process each cluster independently to generate initial atoms.

For each cluster, an immediate atom candidate would be the Karcher mean shape $\bar{\mu} \in \mathcal{S}$. However, $\bar{\mu}$ is not sufficient to summarize the intra-cluster variability. Thus, we propose to perform principal geodesic analysis (PGA), first proposed by [12], to generate the most representative atoms of the cluster. More specifically, we map all cluster elements to the tangent space of the Karcher mean shape $T_{\bar{\mu}}(\mathcal{S})$ using logarithm map, overcoming the lack of vector structure on $\mathcal{S}$. Then, we perform principal component analysis (PCA) in this vector space. Finally, the resulting vectors are mapped to the manifold $\mathcal{S}$ using exponential map to become shapes on $\mathcal{S}$ and constitute initial atoms of $\mathcal{D}$. Atoms generated from all clusters are then gathered together to define the initial dictionary $\mathcal{D}$. Note an important advantage of performing PGA in each cluster and not in the whole training set is that in a cluster, elements on $\mathcal{S}$ are relatively close to each others, *i.e.*, pairwise geodesic distances between them are relatively small. Therefore, when mapping them to $T_{\bar{\mu}}(\mathcal{S})$, we avoid the problematic case of having points that are in the cut locus of $\bar{\mu}$.

Dictionary optimization – We present a dictionary learning algorithm based on the sparse coding framework described above. First, we recall the formulation of the problem in Euclidean space. Given a finite set of training observations $\{z_1, z_2, ..., z_m\}$ in $\mathbb{R}^k$, learning Euclidean dictionary is defined as to jointly minimize the coding cost over all choices of atoms and codes according to:

$$l_E(\mathcal{D}) = \min_{\mathcal{D},w} \sum_{i=1}^{m} \left\| z_i - \sum_{j=1}^{N} [w_i]_j d_j \right\|_2^2 + \lambda f(w_i). \quad (5)$$

To solve this non-convex problem, a common approach alternates between the two sets of variables, $\mathcal{D}$ and w, such that: (1) Minimizing over w while $\mathcal{D}$ is fixed is a convex problem (*i.e.*, sparse coding). (2) Minimizing Eq. (5) over $\mathcal{D}$ while w is fixed is similarly a convex problem.

Moving to the case of Kendall's shape space, $\mathcal{D} = \{\bar{d}_1, \bar{d}_2, ..., \bar{d}_N\}$ is now a dictionary on $\mathcal{S}$, and similarly $\{\bar{Z}_1, \bar{Z}_2, ..., \bar{Z}_m\}$ is a set of training samples on $\mathcal{S}$. Similarly to the Kendall sparse coding problem, we introduce in Eq. (5) the geodesic distance defined on $\mathcal{S}$ computed as $d_{\mathcal{S}}(\bar{Z}, \bar{d}) = \| \log_{\bar{Z}}(\bar{d}) \|_{\bar{Z}}$. As a consequence, the problem of dictionary learning on Kendall's shape space is written as

$$\min_{\mathcal{D},w} \sum_{i=1}^{m} \left\| \sum_{j=1}^{N} [w_i]_j \log_{\bar{Z}_i} \bar{d}_j \right\|_{\bar{Z}_i}^2 + \lambda f(w_i), \quad (6)$$

with the important affine constraint $\sum_{j=1}^{N} [w]_j = 1$. Similar to the Euclidean case, the optimization problem can be solved by iteratively performing sparse coding while fixing $\mathcal{D}$, and optimizing $\mathcal{D}$ while fixing the sparse codes. In Algorithm 2, we provide a summary of the dictionary learning approach on Kendall's shape space.

Algorithm 2 Kendall Dictionary Learning.

Input: Training set $\mathcal{Z} = \{\bar{Z}_i\}_{i=1}^m$, where $\bar{Z}_i \in \mathcal{S}$;
nIter: number of iterations
Output: Kendall dictionary $\mathcal{D} = \{\bar{d}_j\}_{j=1}^N, \bar{d}_j \in \mathcal{S}$

1: Dictionary initialization (Clustering - PGA)
2: **for** $k = 1$ to *nIter* **do**
3: Sparse Coding using Algorithm 1 while $\mathcal{D}$ is fixed, $\{w_i^*\}_{i=1}^m$ are the output sparse codes.
4: Updating atoms using line-search algorithm to solve Eq. (6) while $\{w_i^*\}_{i=1}^m$ are fixed.
5: **end for**

4. Temporal modeling and classification

Let $\{\bar{Z}_1, \bar{Z}_2, ..., \bar{Z}_L\}$ be a sequence of skeletons representing a trajectory on $\mathcal{S}$. As described in section 3.2, we code each skeleton $\bar{Z}_i$ into a sparse vector of codes $w_i \in \mathbb{R}^N$ with respect to a dictionary $\mathcal{D}$ ($\mathcal{D}$ is given a particular structure described later on in this section). As a consequence, each trajectory is mapped to an N-dimensional function of sparse codes and the problem of classifying trajectories on $\mathcal{S}$ is turned to classifying N-dimensional sparse codes functions in Euclidean space, where any traditional operation on Euclidean time-series (*e.g.*, standard machine learning techniques) could be directly applied. Several methods in the literature tend to process and classify time series [1, 2, 35, 36]. In our work, we adopt two different classification schemes to perform action classification: (1) A pipeline of dynamic time warping (DTW), Fourier temporal pyramid (FTP), and one-vs-all linear SVM, as in [35]. Thus, we handle rate variability, temporal misalignment and noise, and classify final features, respectively. We refer to [35] for details; (2) Long short-term memory (LSTM) [7],

which is a variant of recurrent neural networks (RNN) that brings the advantage of learning long-term temporal dependencies. Moreover, we explored the use of bidirectional LSTM (Bi-LSTM), an extension of the traditional LSTM that presents each sequence backwards and forwards to two separate recurrent networks, providing context both from the future and past, respectively [14].

Dictionary structure – In the context of classification, one may exploit the important information of data labels to construct more discriminative feature vectors. To this end, we propose to build *class-specific* dictionaries, similarly to [15]. Formally, let S be a set of labeled sequences on $\mathcal{S}$ belonging to q different classes $\{c_1, c_2, ..., c_q\}$, we aim to build q class-specific dictionaries $\{D_1, D_2, ..., D_q\}$ in $\mathcal{S}$ such that each D_j is learned using skeletons belonging to training sequences from the corresponding class c_j. In this scenario, coding a query skeletal shape $\bar{Z} \in \mathcal{S}$ is done with respect to each $D_{j,1\leq j\leq q}$, independently. As a result, q vectors of codes are obtained. These vectors are then concatenated to form a global feature vector W. As discussed in section 5, this yields to more discriminative feature vectors for classification.

5. Experiments

In this section, we evaluate the proposed skeletal representation using three benchmark datasets presenting different challenges: Florence3D-Action [32], UTKinect-Action [42], and MSR-Action 3D [28]. The obtained recognition accuracies are discussed in section 5.2 with respect to Riemannian approaches, other recent approaches that used 3D skeletal data, and to a kernel-based SCDL approach that we implemented. Additional experiments were conducted to evaluate the main properties of our proposed approach.

Florence3D-Action [32] dataset consists of 9 actions performed by 10 subjects. Each subject performed every action two or three times for a total of 215 action sequences. The 3D locations of 15 joints collected using the Kinect sensor are provided. The challenges of this dataset consist of the similarity between some actions and also the high intra-class variations as same action can be performed using left or right hand.

UTKinect-Action [42] dataset consists of 10 actions performed twice by 10 different subjects for a total of 199 action sequences. The 3D locations of 20 different joints captured with a stationary Kinect sensor are provided. The main challenge of this dataset is the variations in the view point.

MSR-Action 3D [28] dataset consists of 20 actions performed by 10 different subjects. Each subject performed every action two or three times for a total of 557 sequences. The 3D locations of 20 different joints captured with a depth sensor similar to Kinect are provided with the dataset. This is a challenging dataset because of the high similarity between many actions (*e.g.*, *hammer* and *hand catch*).

5.1. Experiments Settings and Parameters

For all datasets, we followed the cross-subject test setting of [38], in which half of the subjects was used for training and the remaining half was used for testing. Reported results were averaged over ten different combinations of training and test data. For Florence3D-Action and UTKinect-Action datasets, we followed an additional setting for each: Leave-one-actor-out (LOAO) [32, 37] and Leave-one-sequence-out (LOSO) [42], respectively. For MSR-Action3D dataset, we also followed [28] and divided the dataset into three subsets AS1, AS2, and AS3, each consisting of 8 actions, and performed recognition on each subset separately, following the cross-subject test setting of [38]. The subsets AS1 and AS2 were intended to group actions with similar movements, while the subset AS3 was intended to group complex actions together. In all experiments, we performed recognition based on two classification schemes, as explained in section 4, to evaluate the performance of our proposed representation and its independency to a specific classifier. In the first scheme, we used a pipeline of DTW, FTP, and one-vs-all linear SVM as in [35]. In all experiments, we used a six-level Fourier temporal pyramid and fixed the value of SVM parameter C to 1. In the second scheme, we train the network with one Bi-LSTM layer. The minimization is performed using Adam optimizer and the applied probability of dropout is 0.3, for all experiments. Due to variations in terms of the number of joints and sequence length for different datasets, the value of neuron size was chosen based on cross-validation for each dataset.

5.2. Results and discussion

A. Comparison to existing Riemannian representations Table 1 reports recognition accuracies for different Riemannian skeletal representations. Conforming to other methods, we compare results obtained using the evaluation protocol of [38] for Florence3D, UTKinect, and MSR-Action, in addition to the protocol of [28] for MSR-Action. Moreover, as in [2] human actions are also first represented as trajectories in the Kendall's shape space, we report additional results of [2] on Florence3D and UTKinect datasets to give more insights about the strength of our coding approach compared to the method of [2]. In Table 1, it can be seen that we obtain better results than all Riemannian approaches on the three datasets. We recall that one drawback of these methods is to map trajectories on manifolds to a reference tangent space, which may introduce distortions in the case points are not close to the reference point. Our method avoids such a non-trivial problem as coding of each shape is performed locally, on its attached tangent space. First, we discuss our results obtained with the first classification scheme, *i.e.*, FTP repre-

sentation with linear SVM, similarly used in [1, 35, 36]. In the three datasets, it is clearly seen that our approach outperforms existing approaches when using the same classification pipeline, which shows the effectiveness of our skeletal representation. For instance, we highlight an improvement of 1.73% on MSR-Action 3D (following protocol [28]) and 1.45% on Florence3D-Action.

Now, we discuss the results we obtained using Bi-LSTM. Note that although we do not perform any preprocessing on the sequences of codes when using this classifier, our approach still outperforms existing approaches on Florence3D, with 1.64% higher accuracy. However, it performs less well on UTKinect yielding an average accuracy of 96.89% against 97.08% obtained in [35]. In MSR-Action 3D, our approach performs better than the method of [1] using the first protocol. Note that in [1], results were averaged over all 242 possible combinations. However, our average accuracy is lower than other approaches following both protocols on this dataset (around 3.5% in the first and 0.62% in the second). Here, it is important to mention that data provided in MSR-Action 3D are noisy [31]. As a consequence, using Bi-LSTM without any additional processing step to handle the noise (*e.g.*, FTP) could not achieve state-of-the-art results on this dataset.

Table 1. Comparison to Riemannian representations.

Method	MSR3D[1]	Florence	UTK	MSR3D[2]
T-SRVF Lie group [1]	85.16	89.67	94.87	–
T-SRVF on $\mathcal{S}$ [2]	89.9	70.40*	89.82*	–
Lie Group [35]	89.48	90.8	97.08	92.46
Rolling rotations [36]	–	91.4	–	–
Kernel-based SCDL*	–	85.76*	88.94*	–
Ours (FTP-SVM)	**90.01**	**92.85**	**97.39**	**94.19**
Ours (Bi-LSTM)	**86.18**	**93.04**	**96.89**	**91.84**

[1] Average accuracy following protocol of [39].
[2] Average accuracy following protocol of [28].
* Experiments were conducted as part of our work.

B. Comparison to State-of-the-art We discuss our results with respect to recent non Riemannian approaches. In all datasets, our approach achieved competitive results.

Florence3D-Action – On this dataset, our method outperforms other methods using Bi-LSTM in the case of LOAO protocol, as shown in Table 2. However, using the second protocol, it is 2.19% lower than [26]. The authors of [26] combine two kernel representations: sequence compatibility kernel (SCK) and dynamics compatibility kernel (DCK) which separately achieved 92.98% and 92.77%, respectively. The proposed approach achieves good performance for most of the actions. However, the main confusions concern very similar actions, *e.g.*, *Drink from a bottle* and *answer phone*.

UTKinect – Results are reported in table 3. Following the LOSO setting, our approach achieves the best recognition rate with each of the adopted classifiers, yielding to an improvement of 2.49% compared to the method of [29], which is based on an extended version of LSTM. For the second protocol, our best result is competitive to the accuracy of 98.2% obtained in [26]. Considering the main challenge of this dataset, *i.e.*, variations in the view point, our approach confirms the importance of the invariance properties gained by adopting the Kendall's representation of shape, hence, the relevance of the resulting functions of codes generated using the geometry of the manifold.

Table 2. Florence3D: comparison with state-of-the-art.

Method	LOAO	prot. of [38]
Graph-based [40]	91.63	–
T-Forest [13]	94.16	–
SCK+DCK [26]	–	**95.23**
Ours (FTP-SVM)	**92.27**	**92.85**
Ours (Bi-LSTM)	**94.48**	**93.04**

Table 3. UTKinect: comparison with state-of-the-art.

Method	LOSO	prot. of [38]
ST-LSTM [29]	97.0	95.0
JLd+RNN [44]	–	95.96
Graph-based [40]	–	97.44
SCK+DCK [26]	–	**98.2**
Ours (FTP-SVM)	**97.50**	**97.39**
Ours (Bi-LSTM)	**98.49**	**96.89**

MSR-Action 3D – For the experimental setting of [28], our best result is competitive to recent approaches. In particular, on AS3, we report the highest accuracy of 100%. This result shows the efficiency of our approach in recognizing complex actions, as AS3 was intended to group complex actions together. On AS1, we achieved one of the highest accuracies (95.87%). However, our result on AS2 is about 8.9% lower than state-of-the-art best result. This shows that our approach performs less well when recognizing similar actions, as AS2 was intended to group similar actions together. Although our best result is slightly higher than [26], it is lower than the same method when following the experimental setting of [39]. This shows that our approach performs better in recognition problems with less classes.

Table 4. MSR-Action 3D: comparison with state-of-the-art.

Method	AS1	AS2	AS3	Avg[1]	Avg[2]
SCK+DCK [26]	–	–	–	93.96	91.45
HBRNN-L [10]	93.33	94.64	95.50	94.49	–
T-Forest [13]	**96.10**	90.54	97.06	94.57	–
ST-NBNN [41]	91.5	**95.6**	97.3	**94.8**	–
Ours (FTP-SVM)	**95.87**	**86.72**	**100**	**94.19**	**90.01**
Ours (Bi-LSTM)	**92.72**	**84.93**	**97.89**	**91.84**	**86.18**

[1] Average accuracy for AS1, AS2, and AS3 following [28].
[2] Average accuracy following protocol of [39].

C. Comparison to an extrinsic SCDL method To further evaluate the strength of the proposed intrinsic approach, we compare it to a kernel-based SCDL method

that we implemented. Several works studied kernels on the 2D Kendall manifold. However, to our knowledge, none of them has proved the existence of valid positive definite (PD) kernels on the 3D Kendall manifold. In [23], for 2D shapes, the authors proved the positive definiteness of the Procrustes Gaussian kernel (PGk) which is based on the full Procrustes distance (fPd). For 3D shapes, we adapted the general kernel-based SCDL formulation of [17] by applying the PGk of [23] in which we also adapted the fPd to 3D shapes as $d_{FP}([z_1],[z_2]) = sin(\theta)$ (see section 4.2.1 of [9]) (θ is the geodesic distance defined in section 3.1). Experimentally, we checked the positive definiteness of the adapted PGk and found out that it is only PD for some values of σ. We empirically chose 0.1 for Florence3D and 0.3 for UTKinect as to have valid PD kernels. Results reported in Table 1 show superiority of our method.

D. Additional Experiments We evaluate some properties of the proposed SCDL approach. In addition, we compare the performance of using Bi-LSTM against a traditional LSTM. These experiments were conducted on the Florence-3D dataset.

Sparsity regularization – In this experiment, we evaluate the effect of the sparsity regularization parameter λ (in Eq. (3) and Eq. (6)) on recognition accuracies obtained using both of the adopted classifiers. To do so, we used half of a training set for learning the dictionary and training the classifiers and the other half for validation. The first graph of Fig. 2 shows the impact of increasing λ from 10^{-4} to 1 at steps of 10^{-2}. Further, we report the average sparsity percentage (*i.e.*, number of non-zero codes divided by the total number of codes) for some values of λ to show the coherence of the obtained codes with the proposed theory. As expected, the sparsity percentage increases when increasing λ. We remark that the accuracy reached a maximum value at $\lambda = 0.01$ (37% of sparsity) and $\lambda = 0.02$ (49% of sparsity) for SVM and Bi-LSTM, respectively. Note that in all previous experiments, λ was chosen empirically so to correspond to these latter percentages of sparsity.

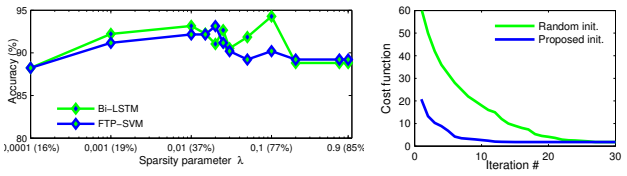

Figure 2. Left: Accuracy when varying the sparsity regularization parameter λ (% values in the x-axis represent the average sparsity). Right: Dictionary learning objective over iterations for: (1) Random initialization; (2) Our proposed initialization based on Bayesian clustering and PGA.

.

Dictionary structure – As described in section 4, we build class-specific dictionaries. To show the relevance of this structure in the context of classification, we compare it to the case of using a global dictionary, *e.g.*, when labels are not taken into account. The obtained recognition accuracies using Bi-LSTM and following the LOAO setting are 94.48% and 91.53% for class-specific and global dictionary, respectively. These results clearly prove that the adopted structure is better in classifying actions.

Dictionary initialization – In this experiment, we evaluate the performance of our proposed initialization step based on Bayesian clustering of shapes and PGA. To this end, we compare it to the case of random initialization, where atoms are randomly selected from the training set. We train a class-specific dictionary (for class *tight lace* in Florence3D dataset) with the same training data in both cases. For the case of random initialization, we set the number of atoms N to 41 to be equal to that of our proposed initialization. Recall that in our approach, N is automatically inferred to avoid its empiric choice, especially as we build class-specific dictionaries. In Fig. 2, on the right graph, we plot the two corresponding dictionary learning objectives over iterations. As it is expected, the proposed initialization shows faster convergence, dividing the overall dictionary learning processing time by approximately two times, when taking into account the execution time of our initialization step.

Performance of Bi-LSTM – We compared average accuracies yielded by Bidirectional LSTM and a traditional LSTM. Following LOAO experimental setting, using Bi-LSTM shows an improvement of around 0.7%, indicating the positive effect of learning both future and past contexts to recognize actions.

6. Conclusion

In this paper, we represented a 3D human skeleton as a point in the Kendall's shape space, hence a human action as a trajectory in this space, to consider important invariance properties for shape analysis. Due to the inherent non-linearity of this manifold, we proposed to sparsely code each skeletal shape on its attached tangent space with respect to a trained dictionary, avoiding the problematic mapping of points to a fixed tangent space attached to the manifold. We initialized the dictionary by clustering skeletal shapes and principal geodesic analysis in the clusters. This step not only accelerated the dictionary learning algorithm but also inferred automatically the number of atoms. We learned the initial dictionary using the geometry of the Kendall's shape space. Our coding scheme yielded to represent trajectories as sparse code functions allowing to directly process and classify them in vector space. This was illustrated on the problem of 3D action recognition using two different classifiers and achieved competitive results with respect to the literature.

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.

[2] B. Ben Amor, J. Su, and A. Srivastava. Action recognition using rate-invariant analysis of skeletal shape trajectories. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(1):1–13, 2016.

[3] D. Bryner, E. Klassen, H. Le, and A. Srivastava. 2d affine and projective shape analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):998–1011, 2014.

[4] H. E. Cetingul and R. Vidal. Intrinsic mean shift for clustering on stiefel and grassmann manifolds. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1896–1902. IEEE, 2009.

[5] H. E. Cetingül and R. Vidal. Sparse riemannian manifold clustering for hardi segmentation. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 1750–1753. IEEE, 2011.

[6] A. Cherian and S. Sra. Riemannian dictionary learning and sparse coding for positive definite matrices. *IEEE Transactions on Neural Networks and Learning Systems*, PP(99):1–13, 2017.

[7] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[8] I. Dryden and K. Mardia. *Statistical shape analysis*. Wiley, 1998.

[9] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis: With Applications in R*. John Wiley & Sons, 2016.

[10] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1110–1118, June 2015.

[11] A. Efros and A. Torralba. Guest editorial: Big data. *International Journal of Computer Vision*, 119(1):1–2, 2016.

[12] P. T. Fletcher, C. Lu, S. M. Pizer, and S. Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE transactions on medical imaging*, 23(8):995–1005, 2004.

[13] G. Garcia-Hernando and T.-K. Kim. Transition forests: Learning discriminative temporal transitions for action recognition and detection.

[14] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.

[15] T. Guha and R. K. Ward. Learning sparse representations for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1576–1588, 2012.

[16] K. Guo, P. Ishwar, and J. Konrad. Action recognition from video using feature covariance matrices. *IEEE Transactions on Image Processing*, 22(6):2479–2494, June 2013.

[17] M. Harandi, R. Hartley, C. Shen, B. Lovell, and C. Sanderson. Extrinsic methods for coding and dictionary learning on grassmann manifolds. *International Journal of Computer Vision*, 114(2-3):113–136, 2015.

[18] M. Harandi and M. Salzmann. Riemannian coding and dictionary learning: Kernels to the rescue. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3926–3935, June 2015.

[19] M. T. Harandi, R. Hartley, B. Lovell, and C. Sanderson. Sparse coding on symmetric positive definite manifolds using bregman divergences. *IEEE transactions on neural networks and learning systems*, 27(6):1294–1306, 2016.

[20] M. T. Harandi, C. Sanderson, R. I. Hartley, and B. C. Lovell. Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach. *CoRR*, abs/1304.4344, 2013.

[21] J. Ho, Y. Xie, and B. Vemuri. On a nonlinear generalization of sparse coding and dictionary learning. In *International conference on machine learning*, pages 1480–1488, 2013.

[22] Z. Huang, C. Wan, T. Probst, and L. Van Gool. Deep learning on lie groups for skeleton-based action recognition. *arXiv preprint arXiv:1612.05877*, 2016.

[23] S. Jayasumana, M. Salzmann, H. Li, and M. Harandi. A framework for shape analysis via hilbert space embedding. In *IEEE ICCV*, pages 1249–1256, 2013.

[24] H. Karcher. Riemannian center of mass and mollifier smoothing. *Comm. Pure Appl. Math.*, 30(5):509–541, Sept. 1977.

[25] D. G. Kendall. Shape manifolds, Procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society*, 16(2):81–121, 1984.

[26] P. Koniusz, A. Cherian, and F. Porikli. Tensor representations via kernel linearization for action recognition from 3d skeletons. In *European Conference on Computer Vision*, pages 37–53. Springer, 2016.

[27] P. Li, Q. Wang, W. Zuo, and L. Zhang. Log-euclidean kernels for sparse representation and dictionary learning. In *2013 IEEE International Conference on Computer Vision*, pages 1601–1608, Dec 2013.

[28] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3D points. In *IEEE Inter. Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB)*, page 914, 2010.

[29] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision*, pages 816–833. Springer, 2016.

[30] Y. M. Lui. Advances in matrix manifolds for computer vision. *Image Vision Comput.*, 30(6-7):380–388, June 2012.

[31] L. L. Presti and M. L. Cascia. 3d skeleton-based human action classification: A survey. *Pattern Recognition*, 53(Supplement C):130 – 147, 2016.

[32] L. Seidenari, V. Varano, S. Berretti, A. D. Bimbo, and P. Pala. Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2013, Portland, OR, USA, June 23-28, 2013*, pages 479–485, 2013.

[33] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1304, 2011.

[34] J. Su, S. Kurtek, E. Klassen, and A. Srivastava. Statistical analysis of trajectories on riemannian manifolds: Bird migration, hurricane tracking, and video surveillance. *Annals of Applied Statistics*, 2013.

[35] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3D skeletons as points in a lie group. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[36] R. Vemulapalli and R. Chellapa. Rolling rotations for recognizing human actions from 3d skeletal data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4471–4479, 2016.

[37] C. Wang, Y. Wang, and A. L. Yuille. Mining 3d key-pose-motifs for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2639–2647, 2016.

[38] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3D action recognition with random occupancy patterns. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part II*, pages 872–885, 2012.

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

[40] P. Wang, C. Yuan, W. Hu, B. Li, and Y. Zhang. Graph based skeleton motion representation and similarity measurement for action recognition. In *European Conference on Computer Vision*. Springer, 2016.

[41] J. Weng, C. Weng, and J. Yuan. Spatio-temporal naive-bayes nearest-neighbor (st-nbnn) for skeleton-based action recognition.

[42] 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.

[43] C. Yuan, W. Hu, X. Li, S. Maybank, and G. Luo. *Human Action Recognition under Log-Euclidean Riemannian Metric*, pages 343–353. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[44] S. Zhang, X. Liu, and J. Xiao. On geometric features for skeleton-based action recognition using multilayer lstm networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 148–157, March 2017.

[45] Z. Zhang, D. Pati, and A. Srivastava. Bayesian clustering of shapes of curves. *Journal of Statistical Planning and Inference*, 166:171 – 186, 2015. Special Issue on Bayesian Nonparametrics.

[46] H. E. etingl, M. J. Wright, P. M. Thompson, and R. Vidal. Segmentation of high angular resolution diffusion mri using sparse riemannian manifold clustering. *IEEE Transactions on Medical Imaging*, 33(2):301–317, Feb 2014.