Rolling Rotations for Recognizing Human Actions from 3D Skeletal Data

Raviteja Vemulapalli, Rama Chellappa
Center for Automation Research
UMIACS, University of Maryland, College Park.

Abstract

Recently, skeleton-based human action recognition has been receiving significant attention from various research communities due to the availability of depth sensors and real-time depth-based 3D skeleton estimation algorithms. In this work, we use rolling maps for recognizing human actions from 3D skeletal data. The rolling map is a well-defined mathematical concept that has not been explored much by the vision community. First, we represent each skeleton using the relative 3D rotations between various body parts. Since 3D rotations are members of the special orthogonal group SO_3, our skeletal representation becomes a point in the Lie group $SO_3 \times \ldots \times SO_3$, which is also a Riemannian manifold. Then, using this representation, we model human actions as curves in this Lie group. Since classification of curves in this non-Euclidean space is a difficult task, we unwrap the action curves onto the Lie algebra $\mathfrak{so}_3 \times \ldots \times \mathfrak{so}_3$ (which is a vector space) by combining the logarithm map with rolling maps, and perform classification in the Lie algebra. Experimental results on three action datasets show that the proposed approach performs equally well or better when compared to state-of-the-art.

1. Introduction

Human action recognition has been an active area of research for the past several decades due to its wide range of applications. Though a significant amount of work has been done over the past few decades, recognizing human actions from RGB videos still remains as a challenging problem due to various nuisance factors like illumination changes, variations in view-point, occlusions and background clutter.

In the recent past, there has been an increased interest in skeleton-based human action recognition approaches due to the availability of cost-effective depth sensors and real-time depth-based skeleton estimation algorithms [28]. These approaches consider the human body as an articulated system of connected rigid segments, and describe human motion using the temporal evolution of the spatial configuration of these segments.
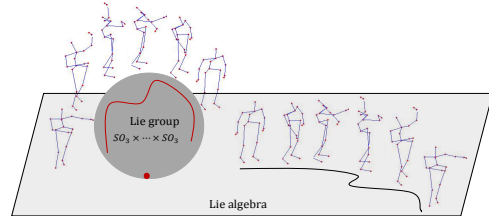

Figure 1: Unwrapping an action sequence onto the Lie algebra by rolling the Lie group $SO_3 \times \ldots \times SO_3$.

Existing skeleton-based action recognition approaches can be broadly divided into two main categories: joint-based approaches and body part-based approaches. Joint-based approaches consider human skeleton as a set of points, and represent it using features like joint positions [5, 8, 9, 13, 16, 21, 26], joint orientations in a fixed coordinate system [24, 37, 32, 22], pairwise relative joint positions [34, 35, 36, 39], etc. On the other hand, part-based approaches consider human skeleton as a set of connected rigid segments, and represent it with features like joint angles [18, 19, 31], bio-inspired 3D features [6], individual part locations [38], relative 3D geometry between parts [33], etc.

Noting that for human action recognition, the relative 3D geometry between various body parts provides a more meaningful description than their absolute locations, [33] used the relative 3D geometry between all pairs of body parts to represent the human skeleton. Specifically, the relative 3D geometry between each pair of body parts was represented as a point in the special Euclidean group SE_3 using the full 3D rigid body transformation required to take one body part to the position and orientation of the other. Using this representation, human actions were modeled as curves in the Lie group $SE_3 \times \ldots \times SE_3$, where $\times$ denotes the direct product between Lie groups. Since this Lie group is a non-Euclidean manifold, action curves were mapped from the Lie group to its Lie algebra using the logarithm map, and action recognition was performed in the Lie algebra. Instead of mapping to the Lie algebra, [1] obtained a lower-dimensional representation for curves in $SE_3 \times \ldots \times SE_3$, by first representing them using the recently-proposed transported square-root vector field [30], and then performing manifold functional principal component analysis.

Since the scale/size of the skeleton varies from subject to subject, it is very important to normalize the skeletal data. In [33], the authors chose one of the skeletons from the training set as reference and normalized all the other skeletons (without changing the joint angles) such that their body part lengths are equal to the corresponding part lengths in the reference skeleton. Interestingly, while the translations between various body parts change with this normalization, the 3D rotations do not change. Hence, instead of explicitly normalizing the skeletal data to handle scale variations, in this work, we obtain a scale-invariant skeletal representation by using only the rotations to describe the relative 3D geometry between body parts. Apart from making the skeletal representation scale-invariant, using only the rotations also reduces the feature dimensionality by half (compared to [33]) thereby speeding up the action recognition pipeline. As shown later in the experiments section, this rotation-based representation performs equally well when compared to the full rigid body transformation-based representation of [33].

Since 3D rotations are members of the special orthogonal group SO_3, our representation becomes a point in the Lie group $SO_3 \times \ldots \times SO_3$, which is also a Riemannian manifold. A similar SO_3-based representation was also used in [22, 32] to represent human skeletons. However, while [22, 32] used only the joint orientations, our skeletal representation includes the 3D rotations between all pairs of body parts. The special orthogonal group was also used earlier for video-based action recognition in [15], where a video sequence was considered as a 3D tensor and the orthogonal matrices obtained by using high-order singular value decomposition were considered as points in $SO(3)$.

Classification of curves in the Lie group $SO_3 \times \ldots \times SO_3$ is a non-trivial task due to the non-Euclidean nature of the underlying space. Similar to [33], we can overcome this difficulty by mapping the action curves from the Lie group $SO_3 \times \ldots \times SO_3$ to its Lie algebra $\mathfrak{so}_3 \times \ldots \times \mathfrak{so}_3$, which is the tangent space at the identity element, using the logarithm map. [1] But, flattening the Lie group using the logarithm map at a single point P introduces distortions due to which curves that are nearby in the Lie group can move away from each other in the Lie algebra (especially when they are not close to the point P). Figure 2 (left) illustrates this pictorially using the example of a sphere. Here, the longitudinal curves moves away from each other when mapped to the tangent space at P using the logarithm map. Note that though we use a sphere for illustration in Figure 2, the manifold of interest here is $SO_3 \times \ldots \times SO_3$.

To reduce the distortions introduced by flattening the Lie group using the logarithm map at a single point, we combine the logarithm map with rolling maps [10, 12, 25] in this
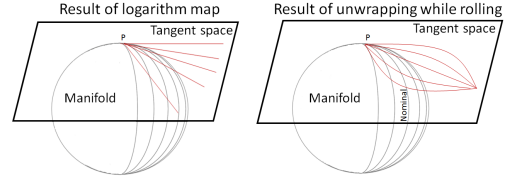
---

[1]Instead of identity element and Lie algebra, one can use Karcher mean of training data and the tangent space at the Karcher mean.

Figure 2: Left: Logarithm map at point P, Right: Unwrapping (via the logarithm map) while rolling along the nominal curve.

work. Rolling maps can be used to flatten the Lie group $SO_3 \times \ldots \times SO_3$ by unwrapping the action curves onto its Lie algebra using the logarithm map while rolling. Figure 2 (right) illustrates the effect of unwrapping (via the logarithm map) while rolling using the example of a sphere. When rolled along the middle longitudinal curve, referred to as the nominal curve in the figure, the other curves that are close to the nominal curve on the sphere remain close to it even after unwrapping onto the tangent space at P.

Though rolling map is a mathematically well-defined concept, it has not been explored much by the computer vision community. Recently, Caseiro *et al*. [4] introduced the rolling map to the vision community by using it for classification of manifold features. In [4], the Grassmann manifold was first rolled as a rigid body over the tangent space at identity, and the data samples were unwrapped onto this tangent space. Then, classification was performed in this tangent space. Rolling maps have also been used for interpolation on SO_3 [11, 27] and Grassmann manifold [3].

In this work, we first compute a nominal curve for each action category in $SO_3 \times \ldots \times SO_3$, and warp all the action curves to these nominal curves using dynamic time warping (DTW). This helps us to handle the rate variations. Then, we roll $SO_3 \times \ldots \times SO_3$ (by rolling each SO_3 individually) over its Lie algebra $\mathfrak{so}_3 \times \ldots \times \mathfrak{so}_3$ along the nominal curves, and unwrap all the action curves (via the logarithm map) onto the Lie algebra while rolling. The main advantage of unwrapping while rolling is that the distances between the action curves and the nominal curves are preserved while mapping the curves from the Lie group to the Lie algebra. Finally, we perform classification in the Lie algebra using a support vector machine (SVM). Our experimental results show that flattening by unwrapping while rolling improves the recognition performance when compared to flattening by using the logarithm map at a single point.

In most of the prior works that used rolling maps, the rolling curve was chosen as a geodesic curve [4, 11, 27]. But, in this work, we are interested in rolling SO_3 along the nominal action curves, which are usually non-geodesic. While [4, 11, 12, 27] provide closed form expressions for the rolling map when the rolling curve is a geodesic, they do not explain how to compute the rolling map in closed form when the rolling curve is non-geodesic. In this work, we show how to obtain a piecewise smooth rolling map for a given (discrete) non-geodesic rolling curve in SO_3.

Contributions:

- We combine the logarithm and rolling maps to flatten the special orthogonal group SO_3 for performing human action recognition from 3D skeletal data. The rolling map is a mathematically well-defined concept that has not been explored much by the vision community. To the best of our knowledge, it was never used in the context of human action recognition.

- Most existing works on rolling maps use a geodesic curve as the rolling curve. They do not provide closed form expressions for the rolling map in the case of a non-geodesic rolling curve. In this work, we show how to compute a piecewise smooth rolling map corresponding to a given (discrete) non-geodesic rolling curve in SO_3.

- We reduce the dimensionality of skeletal representation by half compared to the $\mathfrak{se}_3$-based representation of [33] by using only 3D rotations to describe the relative geometry between various body parts. We show that this scale-invariant rotation-only representation performs equally well when compared to the full rigid body transformation-based representation of [33].

Organization: Section 2 provides the relevant background information on various groups used in this paper and section 3 introduces the rolling map. Section 4 presents the rolling and unwrapping operations for SO_3 and section 5 presents the proposed human action recognition approach. Section 6 presents the experimental results and section 7 concludes the paper.

Notations: We use $\mathcal{R}$ to denote the set of real numbers and I_n to denote the $n \times n$ identity matrix. The determinant, trace, transpose, inverse and Frobenius norm of a matrix A are denoted by $|A|$, $\text{trace}(A)$, $A^\top$, A^{-1} and $\|A\|_{Fr}$ respectively. The tangent space to a manifold $\mathcal{M}$ at a point p is denoted using $T_p\mathcal{M}$ and its orthogonal complement is denoted using $(T_p\mathcal{M})^\perp$. We use $\times$ to represent the direct product between Lie groups.

2. Relevant Background - Groups

In this section, we briefly discuss the groups SO_n, SE_n, SO_n^2 and $SO_n^2 \mathcal{R}^{n^2}$, which will be used in later sections.

SO_n: The special orthogonal group SO_n is a matrix Lie group formed by the set of all $n \times n$ matrices R satisfying the following constraints: $R^\top R = RR^\top = I_n$, $|R| = 1$. The elements of SO_n act on points in $\mathcal{R}^n$ via matrix-vector multiplication:

$$SO_n \circ \mathcal{R}^n \to \mathcal{R}^n, \ \ R \circ p = Rp. \quad (1)$$

The tangent space $T_{R_0}SO_n$ at $R_0 \in SO_n$ is the vector space spanned by the set of all $n \times n$ matrices A such that

$A = \Omega R_0$ for some skew-symmetric matrix Ω. The tangent space at $I_n \in SO_n$ is called the Lie algebra of SO_n and is denoted by $\mathfrak{so}_n$. The special orthogonal group forms a Riemannian manifold with the inner product in each tangent space given by the Frobenius inner product:

$$\langle A_1, A_2 \rangle_{R_0} = \text{trace}(A_1^\top A_2), \ A_1, A_2 \in T_{R_0}SO_n. \quad (2)$$

Under this Riemannian metric, the exponential and logarithm maps at $R_0 \in SO_n$ are given by

$$\begin{aligned}
\exp_{SO_n}(R_0, A) &= \mathbf{e}^{AR_0^\top} R_0, \ A \in T_{R_0}SO_n, \\
\log_{SO_n}(R_0, R_1) &= \mathbf{log}(R_1 R_0^\top)R_0, \ R_1 \in SO_n,
\end{aligned} \quad (3)$$

where $\mathbf{e}$ and $\mathbf{log}$ denote the usual matrix exponential and logarithm. The geodesic curve from R_0 to R_1 is given by $\mathbf{e}^{t\,\mathbf{log}(R_1 R_0^\top)}R_0$, $t \in [0,1]$, and the geodesic distance between R_0 and R_1 is given by $\|\log_{SO_n}(R_0, R_1)\|_{Fr}$.

Interpolation on SO_n: Given $R_1, \ldots, R_n \in SO_n$ at time instances $t_1, \ldots, t_n$ respectively, the following curve $\zeta(t)$ defines a piecewise geodesic curve that passes through R_i at time instance t_i.

$$\zeta(t) = \exp_{SO_n}\left(R_i, \frac{t - t_i}{t_{i+1} - t_i} A_i\right) \text{ for } t \in [t_i, t_{i+1}], \quad (4)$$

where $A_i = \log_{SO_n}(R_i, R_{i+1})$ for $i = 1, 2, \ldots, n-1$.

SE_n: The special Euclidean group SE_n is a matrix Lie group formed by the set of all $(n+1) \times (n+1)$ matrices of the form $E(R, \vec{d}) = \begin{bmatrix} R & \vec{d} \\ 0 & 1 \end{bmatrix}$, $R \in SO_n$, $\vec{d} \in \mathcal{R}^n$. The elements of SE_n represent rigid body motions in an n-dimensional Euclidean space. The matrix R represents the rotation and the vector $\vec{d}$ represents the translation. The action of SE_n on $\mathcal{R}^n$ is defined by:

$$SE_n \circ \mathcal{R}^n \to \mathcal{R}^n, \ \ (R, \vec{d}) \circ p = Rp + \vec{d}. \quad (5)$$

The tangent space at $I_n \in SE_n$ is called the Lie algebra of SE_n and is denoted by $\mathfrak{se}_n$. The Lie exponential and logarithm maps between SE_n and $\mathfrak{se}_n$ are given by

$$\begin{aligned}
\text{Lexp}_{SE_n}(B) &= \mathbf{e}^B, \ B \in \mathfrak{se}_n, \\
\text{Llog}_{SE_n}(E) &= \mathbf{log}(E), \ E \in SE_n.
\end{aligned} \quad (6)$$

For both SO_n and SE_n, the group multiplication and inversion are the usual matrix multiplication and inversion. The group identity element is the $n \times n$ identity matrix I_n.

$SO_n^2 = SO_n \times SO_n$: The group SO_n^2 is the direct product of two special orthogonal groups. It is the set of all matrix pairs (U, V), where $U, V \in SO_n$. The group multiplication and inversion operations are defined as

$$\begin{aligned}
(U_2, V_2) \star (U_1, V_1) &= (U_2 U_1, V_2 V_1), \\
(U, V)^{-1} &= (U^\top, V^\top),
\end{aligned} \quad (7)$$

and the group identity element is given by (I_n, I_n). The group SO_n^2 acts on $\mathcal{R}^{n \times n}$ via

$$SO_n^2 \circ \mathcal{R}^{n \times n} \rightarrow \mathcal{R}^{n \times n}, \ (U, V) \circ Z = U Z V^\top. \quad (8)$$

$\mathbf{SO_n^2 R^{n^2}}$: The group $SO_n^2 R^{n^2}$ is the set of all matrix triplets (U, V, X), where $U, V \in SO_n$ and $X \in \mathcal{R}^{n \times n}$. The group multiplication and inversion operations are defined as

$$(U_2, V_2, X_2) \star (U_1, V_1, X_1) = (U_2 U_1, V_2 V_1, U_2 X_1 V_2^\top + X_2),$$

$$(U, V, X)^{-1} = (U^\top, V^\top, -U^\top X V), \quad (9)$$

and the group identity element is given by $(I_n, I_n, 0)$. The group $SO_n^2 R^{n^2}$ acts on $\mathcal{R}^{n \times n}$ via

$$SO_n^2 R^{n^2} \circ \mathcal{R}^{n \times n} \rightarrow \mathcal{R}^{n \times n}$$
$$(U, V, X) \circ Z = U Z V^\top + X. \quad (10)$$

3. Rolling Motion

For two m-dimensional Riemannian manifolds $\mathcal{M}$ and $\bar{\mathcal{M}}$, both embedded in the same ambient Euclidean space $\mathcal{R}^n$ ($n \geq m$), the rolling motion describes how $\mathcal{M}$ rolls over $\bar{\mathcal{M}}$ as a rigid body without slip and twist. A classical example of such a motion is the rolling of 2-dimensional sphere over the tangent plane at a point.

The curve $\{\alpha(t) \in \mathcal{M} \subset \mathcal{R}^n : t \in [0, T]\}$ along which the manifold $\mathcal{M}$ rolls is called the rolling curve and the curve $\{\bar{\alpha}(t) \in \bar{\mathcal{M}} \subset \mathcal{R}^n : t \in [0, T]\}$, where the rolling curve touches the manifold $\bar{\mathcal{M}}$ while rolling, is called the development curve of α on $\bar{\mathcal{M}}$.

Definition 1: [12, 25] A rolling map describing how $\mathcal{M}$ rolls over $\bar{\mathcal{M}}$, without slip and twist, along a smooth rolling curve $\alpha : [0, T] \rightarrow \mathcal{M}$, is a smooth map

$$h : [0, T] \rightarrow SE_n, \ t \rightarrow h(t) = (R(t), \vec{d}(t)), \quad (11)$$

satisfying the following conditions:

- Rolling conditions

$$\bar{\alpha}(t) := h(t) \circ \alpha(t) \in \bar{\mathcal{M}},$$
$$T_{h(t) \circ \alpha(t)}(h(t) \circ \mathcal{M}) = T_{\bar{\alpha}(t)} \bar{\mathcal{M}}, \quad (12)$$

- No-slip conditions

$$(\dot{h}(t) \circ h(t)^{-1}) \circ \bar{\alpha}(t) = 0, \quad (13)$$

- No-twist conditions

$$(\dot{h}(t) \circ h(t)^{-1}) \circ T_{\bar{\alpha}(t)} \bar{\mathcal{M}} \subset (T_{\bar{\alpha}(t)} \bar{\mathcal{M}})^\perp,$$
$$(\dot{h}(t) \circ h(t)^{-1}) \circ (T_{\bar{\alpha}(t)} \bar{\mathcal{M}})^\perp \subset T_{\bar{\alpha}(t)} \bar{\mathcal{M}}, \quad (14)$$

where for a point $x \in \mathcal{R}^n$ and a vector $\eta \in \mathcal{R}^n$ (i.e., there exists a curve $y : (-\epsilon, \epsilon) \rightarrow \mathcal{R}^n$ such that $\dot{y}(0) = \eta$), the operations $\dot{h}(t) \circ x$, $(\dot{h}(t) \circ h(t)^{-1}) \circ x$ and $(\dot{h}(t) \circ h(t)^{-1}) \circ \eta$ are defined as

$$\dot{h}(t) \circ x := \frac{d}{ds}(h(s) \circ x)|_{s=t}, \quad (15)$$

$$(\dot{h}(t) \circ h(t)^{-1}) \circ x := \frac{d}{ds}((h(s) \circ h(t)^{-1}) \circ x)|_{s=t},$$

$$(\dot{h}(t) \circ h(t)^{-1}) \circ \eta := \frac{d}{ds}((\dot{h}(t) \circ h(t)^{-1}) \circ y(s))|_{s=t}.$$

Remark: Given any piecewise smooth development or rolling curve, the above definition ensures the existence and uniqueness of the corresponding rolling map [12, 25].

4. Rolling Special Orthogonal Group

In this work, we are interested in rolling SO_3 over the tangent plane $T_{R_0} SO_3$ at a point $R_0 \in SO_3$. Note that both SO_3 and $T_{R_0} SO_3$ are 3-dimensional manifolds embedded in the 9-dimensional Euclidean space $\mathcal{R}^{3 \times 3}$. Hence, we can describe the rolling of SO_3 using a curve $h(t) \in SE_9$. However, in [12], it has been shown that for rolling SO_3 over a tangent plane, the rotational and translational components of the original special Euclidean group SE_9 turn out to be SO_3^2 and $\mathcal{R}^{3 \times 3}$ respectively. Therefore, the rolling map can be represented using a curve $c(t) \in SO_3^2 R^9$.

Theorem 1 - Rolling maps for $\mathbf{SO_3}$:
Let $\{\Omega(t) \in \mathfrak{so}_3 \mid t \in [0, T]\}$ be any continuous curve. Let $c(t) = (U(t), V(t), X(t)) \in SO_3^2 \mathcal{R}^9$ be the solution of

$$\dot{X}(t) = \Omega(t) R_0, \quad \dot{U}(t) = -\frac{1}{2} \Omega(t) U(t),$$
$$\dot{V}(t) = \frac{1}{2} R_0^\top \Omega(t) R_0 V(t), \quad (16)$$

satisfying $c(0) = (I_3, I_3, 0)$. Then, the action of $c(t)$ on $SO_3 \subset \mathcal{R}^{3 \times 3}$ results in rolling of SO_3 over the tangent plane $T_{R_0} SO_3$ with the rolling and development curves given by

$$\alpha(t) = U(t)^\top R_0 V(t) \in SO_3,$$
$$\bar{\alpha}(t) = c(t) \circ \alpha(t) = R_0 + X(t) \in T_{R_0} SO_3. \quad (17)$$

Proof: Please refer to [12] for the proof.

The above theorem says that every continuous curve $\Omega(t)$ in the Lie algebra of SO_3 defines a rolling map $c(t)$ through the set of differential equations (16).

Rolling along a geodesic: If $\Omega(t) = \Omega = \mathbf{log}(R_1 R_0^\top)$, then the solution to (16) is given by

$$U(t) = \mathbf{e}^{-\frac{1}{2} t \Omega}, \ V(t) = R_0^\top \mathbf{e}^{\frac{1}{2} t \Omega} R_0, \ X(t) = t \Omega R_0. \quad (18)$$
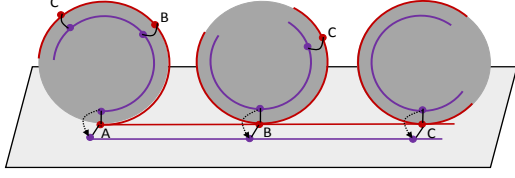
Figure 3: Unwrapping the blue curve onto a tangent space while rolling along the red curve.

In this case, the rolling curve

$$\alpha(t) = U(t)^\top R_0 V(t) = \mathbf{e}^{t\Omega} R_0 = \mathbf{e}^{t\,\log(R_1 R_0^\top)} R_0 \quad (19)$$

is the geodesic from R_0 to R_1, and the development curve is given by $\bar{\alpha}(t) = R_0 + t\Omega R_0$.

4.1. Rolling along a non-geodesic curve

Note that Theorem 1 starts with a curve $\Omega(t) \in \mathfrak{so}_3$ and explains how to obtain the corresponding rolling map $c(t)$ and rolling curve $\alpha(t)$. It doesn't say anything about how to compute the rolling map $c(t)$ starting from a rolling curve $\alpha(t)$. But, in this work, we are interested in rolling SO_3 along specific $\alpha(t)$, which are the nominal action curves obtained using DTW. If the rolling curve $\alpha(t)$ is a geodesic, then the corresponding rolling map $c(t)$ can be computed using (18). But, the nominal action curves along which we want to roll are usually non-geodesic.

Let $\{R_0, R_1, \ldots, R_T\}$ be the discrete representation of the curve along which we want to roll SO_3. In Theorem 2, we show how to obtain a piecewise smooth rolling map $c(t)$ such that the corresponding rolling curve $\alpha(t)$ passes through R_t at time instance t for $t = 0, 1, \ldots, T$.

Theorem 2: Let $\{R_0, R_1, \ldots, R_T\}$ be the given (discrete) rolling curve. Let $\Omega_1, \Omega_2, \ldots, \Omega_T$ be T skew-symmetric matrices defined recursively using

$$\Omega_n = \mathbf{log}\left(\mathbf{e}^{-\frac{\Omega_{n-1}}{2}} \ldots \mathbf{e}^{-\frac{\Omega_1}{2}} R_n R_o^\top \mathbf{e}^{-\frac{\Omega_1}{2}} \mathbf{e}^{-\frac{\Omega_{n-1}}{2}}\right). \quad (20)$$

Let $c(t) = (U(t), V(t), X(t))$ be a curve defined as

$$
\begin{aligned}
U(t) &= \mathbf{e}^{-\frac{(t-n+1)\Omega_n}{2}} \mathbf{e}^{-\frac{\Omega_{n-1}}{2}} \ldots \mathbf{e}^{-\frac{\Omega_1}{2}}, \\
V(t) &= R_0^\top \mathbf{e}^{\frac{(t-n+1)\Omega_n}{2}} \mathbf{e}^{\frac{\Omega_{n-1}}{2}} \ldots \mathbf{e}^{\frac{\Omega_1}{2}} R_0, \\
X(t) &= \sum_{i=1}^{n-1} \Omega_i R_0 + (t-n+1)\Omega_n R_0, \\
t &\in [n-1, n], \ \ n = 1, 2, \ldots, T.
\end{aligned}
\quad (21)
$$

Then, the action of $c(t) \in SO_3^2 \mathcal{R}^9$ on SO_3 results in rolling of SO_3 over the tangent plane $T_{R_0}SO_3$ with a rolling curve $\alpha(t)$ that satisfies

$$\alpha(n) = R_n, \ \text{ for } \ n = 1, 2, \ldots, T. \quad (22)$$

Proof: Let $\{\Omega(t) \in \mathfrak{so}_3 \mid t \in [0, T]\}$ be a curve defined as

$$
\begin{aligned}
\Omega(t) &= 6\Omega_n\left((t-n+1)-(t-n+1)^2\right), \\
t &\in [n-1, n], \ \ n = 1, 2, \ldots, T.
\end{aligned}
\quad (23)
$$

For this $\Omega(t)$, the solution for differential equations (16) is given by (21). Hence by Theorem 1, the action of $c(t)$ on SO_3 results in rolling of SO_3 over the tangent space $T_{R_0}SO_3$ with the rolling curve given by

$$
\begin{aligned}
\alpha(t) &= U(t)^\top R_0 V(t) \\
&= \mathbf{e}^{\frac{\Omega_1}{2}} \ldots \mathbf{e}^{\frac{\Omega_{n-1}}{2}} \mathbf{e}^{(t-n+1)\Omega_n} \mathbf{e}^{\frac{\Omega_{n-1}}{2}} \ldots \mathbf{e}^{\frac{\Omega_1}{2}} R_0, \\
t &\in [n-1, n], \ \ n = 1, 2, \ldots, T.
\end{aligned}
\quad (24)
$$

which satisfies

$$
\alpha(n) = \mathbf{e}^{\frac{\Omega_1}{2}} \ldots \mathbf{e}^{\frac{\Omega_{n-1}}{2}} \mathbf{e}^{\Omega_n} \mathbf{e}^{\frac{\Omega_{n-1}}{2}} \ldots \mathbf{e}^{\frac{\Omega_1}{2}} R_0 = R_n,
\quad (25)
$$
for $n = 0, 1, \ldots, T$. $\blacksquare$

4.2. Unwrapping while rolling

Rolling maps can be used to flatten SO_3 by unwrapping the action curves (while rolling) onto the tangent space at a point using the logarithm map. Figure 3 illustrates this pictorially. In this figure, the blue curve is unwrapped onto a tangent space while rolling along the red curve.

Let $c(t) = (U(t), V(t), X(t)) \in SO_3^2 \mathcal{R}^9$ be the rolling map corresponding to the rolling curve $\alpha(t) \in SO_3$. Let $\bar{\alpha}(t) \in T_{\alpha(0)}SO_3$ be the development curve of $\alpha(t)$. Then, unwrapping (using the logarithm map) of a curve $\beta(t) \in SO_3$ while rolling along $\alpha(t)$ gives the following curve $\bar{\beta}(t) \in T_{\alpha(0)}SO_3$ [27]:

$$
\begin{aligned}
\bar{\beta}(t) &= \log_{SO_3}\left(\alpha(0), \ c(t)o\beta(t) - \bar{\alpha}(t) + \alpha(0)\right) + \bar{\alpha}(t) \\
&= \log_{SO_3}\left(\alpha(0), \ U(t)\beta(t)V(t)^\top\right) + \alpha(0) + X(t).
\end{aligned}
\quad (26)
$$

4.3. Advantage of unwrapping while rolling

The main motivation for using rolling maps in this work is that flattening the SO_3 by unwrapping (via the logarithm map) the action curves while rolling is better than flattening it by using the logarithm map at a single point.

Theorem 3: Let $\{\alpha(t), \ \beta(t) \in SO_3 : t \in [0, T]\}$ be two curves. Let $\bar{\alpha}(t), \bar{\beta}(t) \in T_{\alpha(0)}SO_3$ respectively be the curves obtained by unwrapping (via the logarithm map) $\alpha(t)$ and $\beta(t)$ while rolling the SO_3 over the tangent space at $\alpha(0)$ along the curve $\alpha(t)$. Then, we have

$$d_{T_{\alpha(0)}SO_3}\left(\bar{\beta}(t), \bar{\alpha}(t)\right) = d_{SO_3}(\beta(t), \alpha(t)) \ \forall t, \quad (27)$$

where d_{SO_3} represents the geodesic distance on SO_3 and $d_{T_{\alpha(0)}SO_3}$ represents the standard Euclidean distance in the tangent space $T_{\alpha(0)}SO_3$.

Proof: Let $c(t) = (U(t), V(t), X(t)) \in SO_3^2 \mathcal{R}^9$ be the rolling map corresponding to the rolling curve $\alpha(t)$. Then, by (26) we have

$$\bar{\beta}(t) = \log_{SO_3}\big(\alpha(0),\ U(t)\beta(t)V(t)^\top\big) + \alpha(0) + X(t). \tag{28}$$

Since $\alpha(t)$ is the rolling curve, $\bar{\alpha}(t) = \alpha(0) + X(t)$ from Theorem 1. Hence, we have

$$
\begin{aligned}
& d_{T_{\alpha(0)}SO_3}\big(\bar{\beta}(t), \bar{\alpha}(t)\big) \\
& = \|\bar{\beta}(t) - \bar{\alpha}(t)\|_{Fr} \\
& = \|\log_{SO_3}\big(\alpha(0),\ U(t)\beta(t)V(t)^\top\big)\|_{Fr} \\
& = d_{SO_3}\big(\alpha(0),\ U(t)\beta(t)V(t)^\top\big) \\
& = d_{SO_3}\big(U(t)^\top\alpha(0)V(t),\ \beta(t)\big) \\
& = d_{SO_3}\big(\alpha(t),\ \beta(t)\big).
\end{aligned}
\tag{29}
$$

Here, the second last equality follows from the fact that d_{SO_3} is bi-invariant [14]. ∎

As mentioned earlier, in this work, we first compute a nominal curve for each action category, and warp all the action curves to these nominal curves. Then, we roll the Lie group along the nominal curves and unwrap all the action curves onto the Lie algebra while rolling. As stated in Theorem 3, the main advantage of flattening the action curves by unwrapping while rolling is that the distances between the action curves and the nominal curves are preserved. This is not the case with flattening using the logarithm map at a single point.

Alternative interpretation: The idea of *unwrapping while rolling along the nominal curve* can also be interpreted as the extension of the idea of *tangent plane mapping at Karcher mean* from points to curves. When dealing with points, Karcher mean is commonly used as the anchor point for tangent plane projection. Since we are dealing with curves rather than points in this work, the Karcher mean is replaced by the mean curve. In the case of points, the logarithm map at Karcher mean is used to map the points to a common tangent space. Since we are dealing with curves (a curve can go through various points that are quite far apart), using the logarithm map at a single point to flatten entire curves is not a good idea because, as we move away from the anchor point (which will happen in the case of curves), the distortion due to the logarithm map increases. Instead, it is better to use logarithm maps at multiple points spread over the nominal curve. This is exactly what we are doing while rolling and unwrapping.

5. Proposed Action Recognition Approach

Our 3D skeleton-based human action recognition system consists of the following steps: (1) Skeletal representation, (2) Nominal curve computation using DTW, (3) Rolling and unwrapping, (4) Linear SVM classification.

Table 1: Algorithm for computing a nominal curve

Input: Curves $\gamma_1(t), \ldots, \gamma_N(t)$ at $t = 0, 1, \ldots, T$. Maximum number of iterations max and threshold δ.
Output: Nominal curve $\gamma(t)$ at $t = 0, 1, \ldots, T$.

Initialization: $\gamma(t) = \gamma_1(t)$, iter = 0.
 while iter $< max$
    Warp each curve $\gamma_i(t)$ to the nominal curve $\gamma(t)$ using DTW to get a warped curve $\gamma_i^w(t)$.
    Compute a new nominal $\gamma_{new}(t)$ using
    $\gamma_{new}(t) =$ Karcher mean $\big(\{\gamma_i^w(t)\}_{i=1}^N\big)$.
    if $\sum_{t=0}^T dist\,(\gamma_{new}(t), \gamma(t)) \leq \delta$
       break
    end
    $\gamma(t) = \gamma_{new}(t)$; iter = iter + 1;
 end

Skeletal representation: In this work, we represent a 3D human skeleton using the relative 3D rotations between all pairs of body parts. Since 3D rotations are members of the Lie group SO_3, our skeletal representation becomes a point in the Lie group $SO_3 \times \ldots \times SO_3$. As mentioned earlier, using only the relative 3D rotations makes the skeletal representation scale-invariant and reduces the feature dimensionality by half compared to [33].

Nominal curves: Using the above skeletal representation, we represent human actions as curves in the Lie group $SO_3 \times \ldots \times SO_3$. During training, for each action category, we compute a nominal curve using the algorithm summarized in Table 1, and warp all the curves to this nominal using DTW. This step helps in handling rate variations. For DTW computations, we use the squared Euclidean distance in the Lie algebra. We also performed DTW using the geodesic distance in SO_3, but did not get any improvement in the final classification results. Hence, for faster computations, we use the Lie algebra distance in this paper. Note that in order to compute nominal curves, all the action curves must have same number of samples. For this, we use the interpolation algorithm presented in section 2 and re-sample the curves in $SO_3 \times \ldots \times SO_3$. Interpolation on $SO_3 \times \ldots \times SO_3$ is performed by simultaneously interpolating on individual SO_3.

We note that the recently proposed transported square-root vector field [30] representation of curves, which is an extension of the earlier square-root velocity representation [29] to Riemannian manifolds, provides a distance metric that is invariant to temporal warping (i,.e., the distance between two curve does not change if both curves undergo the same temporal warping). Using this distance metric for DTW and nominal curve computations could further improve our performance.
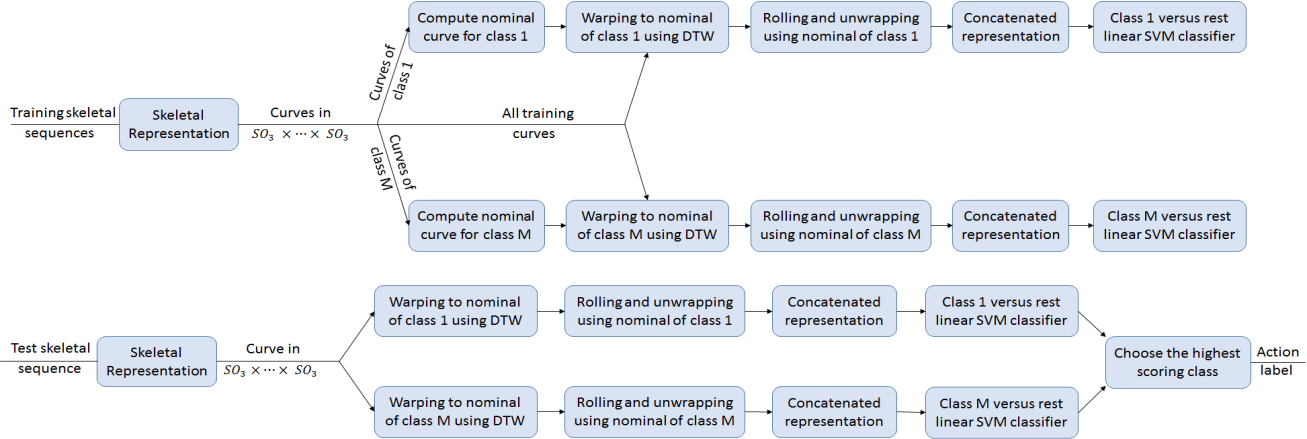
Figure 4: Proposed approach: The top row corresponds to the training phase and the bottom row corresponds to the test phase.

Rolling and unwrapping: In this step, we roll the Lie group $SO_3 \times \ldots \times SO_3$ over its lie algebra $\mathfrak{so}_3 \times \ldots \times \mathfrak{so}_3$ (by rolling each SO_3 individually over its Lie algebra) along each nominal action curve, and unwrap all the action curves onto the Lie algebra. The rolling map for a given (discrete) rolling curve can be obtained using Theorem 2 and the unwrapping operation can be performed using (26). Since a nominal action curve may not start from the identity element (remember that Lie algebra is the tangent space at the identity element), we first roll the Lie group from the identity element to the starting point of the nominal curve and then roll along the nominal curve.

SVM classification: In this step, we first convert each unwrapped action curve into a feature vector by concatenating all the temporal samples, and then classify these feature vectors using a one-vs-all linear SVM classifier.

Figure 4 gives an overview of the proposed approach. The top row shows all the steps involved in training and the bottom row shows all the steps involved in testing.

6. Experiments

We evaluate the proposed human action recognition approach using three action datasets captured with Kinect sensor: Florence3D-Action [23], MSRAction Pairs [20] and G3D-Gaming [2]. The code used for our experiments can be downloaded from http://ravitejav.weebly.com/rolling.html

Florence3D-Action [23] dataset consists of nine different daily actions like *drink water*, *answer phone*, *read watch*, *tight lace*, etc. performed by 10 different subjects. Each subject performed every action two or three times resulting in a total of 215 action sequences. The 3D locations of 15 joints are provided with the dataset.

MSRAction Pairs [20] dataset consists of six action pairs (12 actions in total) like *pick up a box/put down a box*, *wear*

a hat/take off a hat, etc. performed by 10 different subjects. Each subject performed every action two or three times resulting in a total of 353 action sequences. This dataset was collected to analyze how the temporal order affects action recognition. The 3D locations of 20 joints are provided with the dataset.

G3D-Gaming [2] dataset consists of 20 different gaming actions like *golf swing*, *tennis serve*, *bowling*, *aim and fire gun*, etc. performed by 10 different subjects. Each subject performed every action three or more times resulting in a total of 663 action sequences. The 3D locations of 20 joints are provided with the dataset.

Evaluation setting: We followed cross-subject test setting, in which half of the subjects were used for training and the other half were used for testing. All the results reported in this paper were averaged over ten different random combinations of training and test subjects.

Parameters: As explained in section 5, for each dataset, all the action curves were re-sampled to have same length. The reference length was chosen to be the maximum number of samples in any curve in the dataset before re-sampling. The value of SVM parameter C was chosen based on cross-validation.

6.1. Unwrapping while rolling Vs logarithm map

In this work, we are using rolling and unwrapping for flattening the Lie group $SO_3 \times \ldots \times SO_3$. An alternative way to flatten this Lie group is to map the action curves to its Lie algebra using the logarithm map. Table 2 compares the action recognition performance of both these approaches when a linear SVM classifier is used with the concatenated feature representation.

Note that the concatenated representation is nothing but the vectorized version of unwrapped curves (without any additional processing steps). Hence, the results obtained us-

Table 2: Comparison (in terms of classification accuracy) between using the logarithm map at a point and unwrapping while rolling.

Approach	Florence3D [23]	MSRPairs [20]	G3D [2]
Logarithm map at a point	86.83	92.96	87.82
Unwrapping while rolling	**89.82**	**94.09**	**87.95**

ing this representation directly compare the effects of using the logarithm map at a point and unwrapping while rolling. As we can see from Table 2, unwrapping while rolling outperforms the logarithm map by 3% on Florence3D dataset and by 1.1% on MSRPairs dataset. On G3D dataset, both rolling and logarithm map perform equally well. These results suggest that it is better to flatten SO_3 by unwrapping while rolling instead of using the logarithm map at a point.

6.2. Comparison with state-of-the-art

Note that while we use a simple classification scheme in which the Lie algebra curves are first vectorized by concatenating the temporal samples and then classified using a linear SVM classifier, existing state-of-the-art approaches like [1, 33] use additional processing steps like Fourier temporal pyramid representation (FTP) [33] (originally proposed by [34]), principal geodesic analysis [1], etc. While our simple approach does produce impressive results, it may not be sufficient to achieve state-of-the-art results. Hence, to compare with the state-of-the-art approaches, we incorporate the FTP representation proposed by [34] into our classification scheme. Instead of using the simple concatenated representation, we represent each unwrapped Lie algebra curve using the FTP representation, and then classify them using a linear SVM classifier.

Table 3 compares the results of the proposed approach with state-of-the-art (skeleton-based) results reported on Florence3D, MSRPairs and G3D datasets. As we can see, the proposed approach performs better or equally well when compared to the recent state-of-the-art skeleton-based approaches. Note that since the focus of this work is on skeleton-based action recognition, we use only skeleton-based approaches for comparison. Though combining skeletal features with depth-based features may improve the accuracy, feature fusion is beyond the scope of this work.

7. Conclusion and Future Work

In this work, we used rolling maps for flattening SO_3 to perform human action recognition from 3D skeletal data. We represented each human skeleton as a point in the Lie group $SO_3 \times \ldots \times SO_3$ using the relative 3D rotations between all pairs of body parts. Using this skeletal representation, we represented human actions as curves in $SO_3 \times \ldots \times SO_3$. For each action category, we computed a

Table 3: Comparison with state-of-the-art.

Florence3D dataset	
Multi-Part Bag-of-Poses [23]	82.00
Motion trajectories [7]	87.04
Elastic Functional Coding [1]	89.67
Relative 3D geometry [33]	90.71
Proposed (concatenated representation)	89.82
Proposed (FTP representation)	**91.40**
MSRPairs dataset	
Relative 3D geometry [33]	93.65
Proposed (concatenated representation)	94.09
Proposed (FTP representation)	**94.67**
G3D dataset	
RBM + HMM [17]	86.40
Relative 3D geometry [33]	**91.09**
Proposed (concatenated representation)	87.95
Proposed (FTP representation)	90.94

nominal curve and warped all the action curves to this nominal using DTW. Then, we rolled $SO_3 \times \ldots \times SO_3$ over its Lie algebra along the nominal curves and unwrapped all the action curves onto the Lie algebra. Finally, we represented the unwrapped curves using either the concatenated representation or the FTP representation and classified them using a one-vs-all linear SVM classifier. By evaluating on three action datasets, we showed that flattening SO_3 by unwrapping while rolling performs better than flattening SO_3 by using logarithm map a single point. The proposed approach also outperforms various state-of-the-art skeleton-based action recognition approaches.

Note that in order to roll along the nominal curves, we should be able to compute the rolling map corresponding to a given non-geodesic rolling curve. In this work, we showed how to compute a piecewise smooth rolling map such that the rolling curve passes through a given set of points in SO_3 at given instances of time.

The rolling map is a general concept that can be used with any Riemannian manifold. Hence, as part of future work, we plan to use rolling maps for classification of time series data on other manifolds like Grassmann manifold and the manifold of symmetric positive definite matrices.

References

[1] R. Anirudh, P. Turaga, J. Su, and A. Srivastava. Elastic Functional Coding of Human Actions: From Vector-Fields to Latent Variables. In *CVPR*, 2015. 1, 8

[2] V. Bloom, D. Makris, and V. Argyriou. G3D: A Gaming Action Dataset and Real Time Action Recognition Evaluation Framework. In *CVPR Workshops*, 2012. 7, 8

[3] R. Caseiro, J. F. Henriques, P. Martins, and J. Batista. Beyond the Shortest Path : Unsupervised Domain Adaptation by Sampling Subspaces Along the Spline Flow. In *CVPR*, 2015. 2

[4] R. Caseiro, P. Martins, J. F. Henriques, F. S. Leite, and J. Batista. Rolling Riemannian Manifolds to Solve the Multi-class Classification Problem. In *CVPR*, 2013. 2

[5] A. A. Chaaraoui, J. R. Padilla-López, and F. Flórez-Revuelta. Fusion of Skeletal and Silhouette-based Features for Human Action Recognition with RGB-D Devices. In *ICCV Workshops*, 2013. 1

[6] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal. Bio-inspired Dynamic 3D Discriminative Skeletal Features for Human Action Recognition. In *CVPR Workshops*, 2013. 1

[7] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. D. Bimbo. 3D Human Action Recognition by Shape Analysis of Motion Trajectories on Riemannian Manifold. *IEEE Transactions on Cybernetics*, PP(99), 2014. 8

[8] M. A. Gowayyed, M. Torki, M. E. Hussein, and M. El-Saban. Histogram of Oriented Displacements (HOD): Describing Trajectories of Human Joints for Action Recognition. In *IJCAI*, 2013. 1

[9] J. Gu, X. Ding, S. Wang, and Y. Wu. Action and Gait Recognition From Recovered 3-D Human Joints. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 40(4):1021–1033, 2010. 1

[10] K. Hüper, M. Kleinsteuber, and F. S. Leite. Rolling Stiefel Manifolds. *International Journal of Systems Science*, 39(9):881–887, 2008. 2

[11] K. Hüper and F. S. Leite. Smoothing Interpolation Curves on Manifolds with Applications to Path Planning. In *Mediterranean Conference on Control and Automation*, 2002. 2

[12] K. Hüpper and F. S. Leite. On the Geometry of Rolling and Interpolation Curves on Sn, SOn, Grassmann Manifolds. *Journal of Dynamical and Control Systems*, 13(4):467–502, 2007. 2, 4

[13] M. Hussein, M. Torki, M. Gowayyed, and M. El-Saban. Human Action Recognition Using a Temporal Hierarchy of Covariance Descriptors on 3D Joint Locations. In *IJCAI*, 2013. 1

[14] D. Q. Huynh. Metrics for 3D Rotations: Comparison and Analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009. 6

[15] Y. M. Lui. Tangent Bundles on Special Manifolds for Action Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(6):930–942, 2012. 2

[16] F. Lv and R. Nevatia. Recognition and Segmentation of 3D Human Action Using HMM and Multi-class Adaboost. In *ECCV*, 2006. 1

[17] S. Nie and Q. Ji. Capturing Global and Local Dynamics for Human Action Recognition. In *ICPR*, 2014. 8

[18] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Sequence of the Most Informative Joints (SMIJ): A New Representation for Human Skeletal Action Recognition. In *CVPR Workshops*, 2012. 1

[19] E. Ohn-bar and M. M. Trivedi. Joint Angles Similarities and HOG^2 for Action Recognition. In *CVPR Workshops*, 2013. 1

[20] O. Oreifej and Z. Liu. HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences. In *CVPR*, 2013. 7, 8

[21] M. Reyes, G. Dominguez, and S. Escalera. Feature Weighting in Dynamic Time Warping for Gesture Recognition in Depth Data. In *ICCV Workshops*, 2011. 1

[22] S. Said, N. Courty, N. L. Bihan, and S. J. Sangwine. Exact Principal Geodesic Analysis for Data on SO(3). In *EU-SIPCO*, 2007. 1, 2

[23] 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 *CVPR Workshops*, 2013. 7, 8

[24] Z. Shao and Y. F. Li. A New Descriptor for Multiple 3D Motion Trajectories Recognition. In *ICRA*, 2013. 1

[25] R. W. Sharpe. *Differential Geometry*. Springer-Verlag, New York, 1996. 2, 4

[26] Y. Sheikh, M. Sheikh, and M. Shah. Exploring the Space of a Human Action. In *ICCV*, 2005. 1

[27] Y. Shen, K. Hüper, and F. S. Leite. Smooth Interpolation of Orientation by Rolling and Wrapping for Robot Motion Planning. In *ICRA*, 2006. 2, 5

[28] 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 a Single Depth Image. In *CVPR*, 2011. 1

[29] A. Srivastava, E. Klassen, S. H. Joshi, and I. H. Jermyn. Shape Analysis of Elastic Curves in Euclidean Spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1415–1428, 2011. 6

[30] 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*, 8(1), 2014. 1, 6

[31] J. Sung, C. Ponce, B. Selman, and A. Saxena. Unstructured Human Activity Detection from RGBD Images. In *ICRA*, 2012. 1

[32] M. Tournier, X. Wu, N. Courty, E. Arnaud, and L. Revéret. Motion Compression using Principal Geodesics Analysis. *Comput. Graph. Forum*, 28(2):355–364, 2009. 1, 2

[33] R. Vemulapalli, F. Arrate, and R. Chellappa. Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group. In *CVPR*, 2014. 1, 2, 3, 6, 8

[34] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining Actionlet Ensemble for Action Recognition with Depth Cameras. In *CVPR*, 2012. 1, 8

[35] J. Wang and Y. Wu. Learning Maximum Margin Temporal Warping for Action Recognition. In *ICCV*, 2013. 1

[36] P. Wei, N. Zheng, Y. Zhao, and S. C. Zhu. Concurrent Action Detection with Structural Prediction. In *ICCV*, 2013. 1

[37] L. Xia, C. C. Chen, and J. K. Aggarwal. View Invariant Human Action Recognition Using Histograms of 3D Joints. In *CVPR Workshops*, 2012. 1

[38] Y. Yacoob and M. J. Black. Parameterized Modeling and Recognition of Activities. In *ICCV*, 1998. 1

[39] X. Yang and Y. Tian. EigenJoints-based Action Recognition Using Naïve-Bayes-Nearest-Neighbor. In *CVPR Workshops*, 2012. 1