Fast Adaptive Reparametrization (FAR) With Application to Human Action Recognition

Enjie Ghorbel, Girum Demisse, Djamila Aouada, and Björn Ottersten

Abstract—In this letter, a fast approach for curve reparametrization, called Fast Adaptive Reparametrization (FAR), is introduced. Instead of computing an optimal matching between two curves such as Dynamic Time Warping (DTW) and elastic distance-based approaches, our method is applied to each curve independently, leading to linear computational complexity. It is based on a simple replacement of the curve parameter by a variable invariant under specific variations of reparametrization. The choice of this variable is heuristically made according to the application of interest. In addition to being fast, the proposed reparametrization can be applied not only to curves observed in Euclidean spaces but also to feature curves living in Riemannian spaces. To validate our approach, we apply it to the scenario of human action recognition using curves living in the Riemannian product Special Euclidean space SE(3)^n. The obtained results on three benchmarks for human action recognition (MSRAction3D, Florence3D, and UTKinect) show that our approach competes with state-of-the-art methods in terms of accuracy and computational cost.

Index Terms—Reparametrization, action recognition, Riemannian manifolds.

I. INTRODUCTION

CURVES have been shown to be a very powerful representation in the fields of computer vision and pattern recognition. In fact, due to their simplicity, they are used for the description of various observations such as objects or shapes, through their contours in 2D [1]–[3], or level sets in 3D [4]–[6]. Curves are also used to represent temporal information such as speech [7], [8], and motion field [9]–[15].

A curve may be defined as a mapping \( \alpha : I \rightarrow \mathcal{M} \) where \( I \subseteq \mathbb{R} \) is an interval of reals regulating the order of the data to be mapped to \( \mathcal{M} \), the manifold of observations. The interval \( I \) governs what is called the parametrization of the curve. However, curve parametrization is not unique and the same curve in \( \mathcal{M} \) can be expressed in an infinite number of different parameter spaces. Hence, this variability of parametrization can lead to the mismatch of similar curves. Thus, to compare and manipulate curves, a reparametrization step is usually undertaken where the domain \( I \) of a curve is to be redefined [16], [17], [3]. First, curve matching techniques have been based on the detection of a set of finite points called landmarks [18], [19] which are used to statistically describe curve shapes. In later works, more sophisticated reparametrization techniques using the global information of curves [20], [17] have been introduced. These curves are usually considered as varying in a Euclidean space. Nevertheless, some discriminative curves lie in nonlinear spaces. More particularly, curves living in Riemannian manifolds have shown great potential such as the space of Symmetric Positive Definite matrices [21] and the Special Euclidean group [22], [3]. Thus, the non-linearity of these informative spaces makes classical reparametrization methods not directly applicable.

To overcome this issue, several attempts have been made towards the generalization of reparametrization to Riemannian manifolds. For instance, Dynamic Time Warping (DTW) [23], which has been initially designed for curves in \( \mathbb{R}^n \), has been adapted to different Riemannian manifolds [24] by replacing the Euclidean distance used to compute the similarity measure by a more appropriate entity (geodesic distance, distance between curves projected to the Lie algebra). On the other hand, instead of using the similarity measure resulting from DTW, other approaches have proposed to define elastic distances [9], [2], [16]. This rate-invariant distance is usually computed by defining an optimization problem that finds the optimal parametrization. In [16], the Transported Square-Root Vector Field (TSRVF) which represents the extension of the Square-Root Vector Field (SRVF) method [3] to Riemannian manifolds has been introduced. Anirudh et al. [9] applied the TSRVF framework on two different Riemannian manifolds (\( \text{SE}(3) \)) and the Special Orthogonal group \( \text{SO}(3) \)). Also, Demisse et al. [2] described curves as sequences of transformation matrices of \( \text{SE}(3) \) and formulated the problem of curve matching as an optimization problem using the geodesic distance of \( \text{SE}(3) \).

However, in spite of their effectiveness, these generalized methods designed for curves in Riemannian spaces present some drawbacks: (1) they are highly dependent on the quality of data since curves are compared in pairs; (2) they rely on optimization leading to an important computational complexity, and (3) they are performed using discrete data that are matched without necessarily corresponding to each other.

To address these limitations, a novel and fast curve reparametrization framework applicable to Euclidean and Riemannian spaces called Fast Adaptive Reparametrization (FAR) is introduced. Instead of comparing curves in pairs, this method...
One of the strengths of FAR is denoted as is and be two curves with α between two curves is some-
I which include a.
E̸ α F α α is computed based I ◦ as follows:
definition 1): curves are considered equivalent if they ROPOSED using a score function s α and →M d R:
F, s = \[α = [0\ f s a R s α : I → I and I f (3) = [E → (2) α ∈ a a a by heuristically defining s →M α α SE , I I α → in link with the application. The function s α(t) is therefore computed and applied to α(t) as described below.
\[α(s_α(t)) = α ∘ s_α(t). \] (2)
This heuristic approach is inspired by the TVR algorithm pro-
posed in [10]. However, while TVR is restricted to the specific case of temporal normalization for human action recognition using curves lying in Euclidean spaces, our approach presents a generalized formulation applicable to various applications for curves lying not only in Euclidean spaces but also in Riemannian manifolds. Therefore, the main idea of our approach is to define s α using a score function f α : \([a_α, b_α] → \mathbb{R}\) which include a heuristic invariance under reparametrization of relevant curve properties. Instead of relying on the pure mathematical definition of curve equivalence, a physical component is included to orient the choice of the function f α. In other words, our main goal is not to make the curves themselves invariant to reparametrization, but to restrict this invariance to the discriminative curve features in link with the application. The function f α is computed based on the curve α itself. Therefore, we note that when designing f α, it is important to take into account the topological structure of the space M. Indeed, if curves are living in a non-linear space M, classical measures such as Euclidean distance, derivative, etc. are not suitable.
Since open curves are numerically assimilated to ordered data, s α(t) is constrained to be strictly increasing with respect to the variable t. To that aim, an increasing energy function E α is designed using f α as follows:
\[E_α(t) = \int_a^t \|f_α(u)\|^2 du. \] (3)
Furthermore, in order to ensure the variation in a single interval \(I = [0, 1]\), the parametrization function s α(t) is computed as follows:
\[s_α(t) = \frac{E_α(t)}{E_α(b)}. \] (4)
Note that arclength reparametrization can be considered as a specific case of this reparametrization, for which \(f_α(t) = \sqrt{α(t)}\).
Numerical consideration: One of the strengths of FAR is that it takes into consideration the discrete nature of the data. Instead of trying to match non-corresponding curve points, we propose to follow a different strategy: we realize an interpolation of curves followed by uniform sampling. Figure 2 illustrates the different components of the proposed framework.

However, in pattern recognition, the definition of curve equivalence is slightly different from the original mathematical defi-
nition (definition 1): curves are considered equivalent if they describe a similar pattern. Thus, the notion of equivalence can be confused with the notion of class. In fact, the goal of reparametrizing curves is to ensure that curves describing the same class look more similar. Therefore, it can be stated that the meaning of equivalence widely depends on the final application and on the properties that we aim to extract from curves. Thus, we propose to make use of our prior knowledge to define s(t) in or

dd to overcome execution rate variability, the curves are reparametrized using the FAR approach by respectively computing s α 1 and s α 2.
reparametrizes any curve α : I → M by heuristically defining an adequate homeomorphism s α 1 : I α → I such that α : I α → M. To avoid relying on discrete data, the reparametrized nu-
merical curves are interpolated and resampled uniformly. To evaluate our approach, FAR is adapted to the context of human action recognition in \(\mathbb{SE(3)}^n\), as illustrated in Figure 1, and tested on three different benchmarks.

II. PROPOSED FAST ADAPTIVE REPARAMETRIZATION

The main goal of the proposed approach is to extract the properties of curves that are invariant under reparametrization in order to recognize equivalent curves. The question to be raised is, therefore, the following: what is the meaning of equivalent curves?
Let α 1 : I 1 → M and α 2 : I 2 → M be two curves with M as the observation manifold and I 1 = [a 1, b 1], I 2 = [a 2, b 2] ∈ \(\mathbb{R}\) as their respective parameter domains. Follows the classical mathematical definition of equivalent curves.
definition 1]: The two curves α 1 and α 2 are considered to be equivalent (α 1 ∼ α 2) if and only if there exists a one-to-one function s(t) : I 2 → I 1 such that \(∀t, s(t) ≠ 0\) and \(α 2(s(t)) = α 1(t)\).
The primary challenge is therefore to find the appropriate parametrization function s(t).
Usually, reparametrization techniques as in [9], [2], [3] define s(t) by minimizing a well-chosen similarity measure \(d\) between a pair of curves α 1 and α 2 such that
\[s^* = \min_s d(α 1 ∘ s, α 2), \] (1)
where \(s^*\) denotes the optimal parametrization of α 2 with respect to α 1. Unfortunately, this leads to polynomial complexity. Note that the similarity measure \(d\) between two curves is sometimes implicitly defined using an intermediate mapping such as TSRVF [16], SRVF [3], etc.
III. APPLICATION TO HUMAN ACTION RECOGNITION IN $\mathbb{SE}(3)$

In this section, we present the application of FAR in the context of action recognition using features expressed in $\mathbb{SE}(3)$. First, a brief mathematical background that overviews the properties of the Riemannian manifold $\mathbb{SE}(3)$ is recalled. Then, the used human motion representation, as well as the adaptation of FAR in this particular case, are described.

A. Special Euclidean Group $\mathbb{SE}(3)$

The Special Euclidean group $\mathbb{SE}(3)$ is defined as the set containing the $4 \times 4$ square matrices with the following form:

$$T(R, \vec{t}) = \begin{pmatrix} R & \vec{t} \\ 0 & 1 \end{pmatrix}, \quad (5)$$

where $R$ is a three-dimensional rotation matrix and $\vec{t}$ is a three-dimensional translation vector. As $\mathbb{SE}(3)$ is a Lie group and consequently is a non-linear structure, classical operations are not trivial. The more common practice is to map elements of $\mathbb{SE}(3)$ to $\mathfrak{se}(3)$, and vice-versa, the exponential and the logarithm maps respectively denoted by $\exp$ and $\log$ are used. For more details about differential geometry and the Lie group $\mathbb{SE}(3)$, we respectively refer the reader to [25] and [26].

Distance in $\mathbb{SE}(3)$: The geodesic distance $d_G$ between two matrices $T_A(R_A, \vec{t}_A)$ and $T_B(R_B, \vec{t}_B) \in \mathbb{SE}(3)$ can be computed as follows [2]:

$$d_G(T_A, T_B) = \left\| \log(T_A^T R_B) \right\|_F^2 + \left\| t_A - t_B \right\|_2^2 \right)^{1/2}. \quad (6)$$

Distance in $\mathbb{SE}(3)^n$: The cross product space $\mathbb{SE}(3)^n$ inherits the Lie group structure of $\mathbb{SE}(3)$. Therefore, the geodesic distance $D$ between two elements $\alpha_1 = (T_1, T_2, \ldots, T_n)$ and $\alpha_2 = (Q_1, Q_2, \ldots, Q_n) \in \mathbb{SE}(3)^n$, with $T_i$ and $Q_i \in \mathbb{SE}(3)$ for all $i$, is computed as follows:

$$D(\alpha_1, \alpha_2) = \left( \sum_{i=1}^{n} d_G(T_i, Q_i)^2 \right)^{1/2}. \quad (7)$$

B. FAR Reparametrization in $\mathbb{SE}(3)^n$

To model human motion using skeletal data, we propose to use the representation introduced in [22] called Lie Algebra Relative Pairs (LARP) representing trajectories lying in $\mathbb{SE}(3)^n$.

In [22], authors suggest that a human skeleton $S$, composed of $m$ edges, at an instant $t$ can be represented by an ordered set of transformation matrices $\alpha(t) = \{T_{1,2}(t), T_{2,1}(t), \ldots, T_{m,m-1}(t), T_{m-1,m}(t)\} \in \mathbb{SE}(3)^n$. These transformation matrices are estimated between using the skeleton edges. Thus, the full action can be seen as a trajectory $(\alpha(t))_{t \in [t_0, t_0+L]}$ varying in the product Special Euclidean Group $\mathbb{SE}(3)^n$ with $L$ the duration of the action, $t_0$ the starting time and $n = 2C_m^2$, where $C_m^2$ is the number of combinations. More details about this representation can be found in the original letter [22].

To overcome execution rate variability, we propose to apply to the LARP representation, the FAR approach on the obtained curve $\alpha$, as described in (4). To compute the parametrization function $s_{\alpha}(t)$, two different score functions $f_{\alpha}(t)$ are proposed, namely, arclength and motion quantity.

Arclength: To compute the arclength of the curve $\alpha(t)$ at an instant $t$ on the manifold $\mathbb{SE}(3)^n$, we propose to compute the geodesic distance $D$ between $\alpha(t)$ and $\alpha(t-1)$,

$$f_{\alpha}(t) = D(\alpha(t), \alpha(t-1)). \quad (8)$$

Therefore, the faster the motion is, the higher $f(t)$ is. Integrating this function in the parametrization function $s(t)$ makes $\alpha$ invariant to velocity variation. This can be seen as the extension of arclength reparametrization for the space $\mathbb{SE}(3)^n$.

Pose motion: We define pose motion as the distance between the current pose $\alpha(t)$ and the rest pose denoted as $\alpha_{ref}$. We propose to compute it using the geodesic distance $D$ as follows:

$$f_{\alpha}(t) = D(\alpha(t), \alpha_{ref}). \quad (9)$$

The closer to the rest pose the current pose is, the less important it is considered. In the experiments, we assume that the rest pose is encoded in the first frame of the video since the used datasets are composed of segmented actions. Nevertheless, this approach can be further extended by learning a generic rest pose model.

Interpolation and uniform sampling: We propose to use the interpolation scheme for $\mathbb{SE}(3)$ as introduced in [27] and used in [22]. Each reparametrized temporal sequence $\mathbb{SE}(3)$ is interpolated individually. For each sequence $\alpha_j$ defined by $T_j(t_0), T_j(t_1), \ldots, T_j(t_i), \ldots, T_j(t_0 + L) \in \mathbb{SE}(3)$, the following interpolation formula is used. Considering $s(t) \in [s(t_i), s(t_{i+1})]$, we have:

$$\hat{\alpha}_j(s(t_i)) = T_j(s(t_i)) \exp \left( \frac{s(t) - s(t_i)}{s(t_{i+1}) - s(t_i)} B_i \right), \quad (10)$$

with $B_i = \log(T_j(s(t_i)))^{-1} T_j(s(t_{i+1})))$.

Finally, the interpolated curve $\hat{\alpha} = (\hat{\alpha}_j)_{j \in [1, n]}$ is uniformly sampled to recover the final descriptor.

IV. EXPERIMENTS

Our reparametrization approach is applied to LARP, as presented in Section III, and tested on three benchmarks collected for action recognition: Florence3D dataset [28], UTKinect dataset [29] and MSRAAction3D dataset [30].
Florence3D dataset [28] is composed of 9 different actions. Each action is performed by 10 different subjects from two to three times. This dataset acquired using Kinect provides RGB images, depth maps as well as skeleton sequences. Each skeleton is constituted of 15 joints. UTKinect dataset proposed in [29] contains 10 different classes of action. Each action class is repeated by 10 different subjects twice. Captured using Kinect, the three RGB-D modalities are also provided for this benchmark. However, in contrast to Florence3D, skeletons are formed by 20 joints. MSRAction3D introduced in [30] is probably one of the most well-known datasets. This dataset is composed of 20 actions performed by 10 different subjects from two to three times. It is also captured using Kinect. However, only skeletons and depth maps are provided. As in the UTKinect dataset, each skeleton is composed of 20 joints.

Our reparametrization applied to LARP is denoted as LARP+FA-R-A (when using the arclength as a score function \( f_\alpha \)) and LARP+FA-R-MQ (when using the motion quantity as \( f_\mu \)). As discussed in [9], the Fourier Temporal Pyramid (FTP), which is a tool used to remove noise from skeletons, is not applied to non-Euclidean features. Indeed, analyzing and exploring the warping is not a trivial task when using FTP. In the experiments, we follow the same protocol proposed in [9], where five train-test protocols are realized and then averaged. For each protocol, half of the subjects are used for the training, while the rest is used for testing. For the classification, we also use the same one-vs-all SVM classifier proposed in [22].

To evaluate our approach, we combine the LARP representation with different reparametrization techniques. We compare our approach to LARP without carrying out any warping, using the modified DTW introduced in [22], the TSRVF method proposed in [9], the TSRVF with manifold functional variant of Principal Component Analysis (mfPCA) [9] and Principal Geodesic Analysis introduced in [31]. These approaches are respectively denoted as LARP, LARP+DTW, LARP+TSRVF, LARP+mfPCA, and LARP+PGA.

The results obtained on the Florence3D, UTKinect and MSRAction3D datasets, are reported in Table I.

### TABLE I

| Method            | Florence3D | UTKinect | MSRAction3D |
|-------------------|------------|----------|-------------|
| LARP [22]         | 86.27%     | 93.57%   | 75.57%      |
| LARP+DTW [22]     | 86.74%     | 92.17%   | 78.75%      |
| LARP+PGA [31]     | 79.01      | 91.26    | 72.06       |
| LARP+TSRVF [9]    | 89.50      | 94.47    | 84.62       |
| LARP+mfPCA [9]    | 89.67      | 94.87    | 85.16       |
| LARP+FA-R-A (ours)| 90.88      | 95.35    | 83.17       |
| LARP+FA-R-MQ (ours)| 92.96     | 96.16    | 83.03       |

### Arclength (A) vs. Motion Quantity (MQ)

The results show that LARP+FA-R present better results on Florence3D and UTKinect datasets when associated to the MQ motion signal (with respectively 92.96% and 86.16% using LARP+FA-R-MQ against 90.88% and 95.35% using LARP+FA-R-A). This could be explained by the fact that MQ function highlights the frames containing key poses and is less sensitive to noise resulting from undesired motion. However, on MSRAction dataset, the obtained result using A as a score function is slightly higher than the one registered for LARP+FA-R-MQ (only a difference of 0.15%).

### Comparison with state-of-the-art

Compared to LARP+DTW and LARP+PGA, our method improves the performance on the three datasets by around 4% and 13%, respectively, on Florence3D, and 4% and 5% on UTKinect, and 4% and 9% on MSRAction3D. Also, compared to LARP+TSRVF and LARP+mfPCA, our approach enhances the results by respectively 3% and 3% on Florence3D and by 2% for both on UTKinect. Nevertheless, LARP+TSRVF and LARP+mfPCA present better performance than LARP+FA-R on MSRAction3D by slightly exceeding it by respectively 1% and 2%. Despite that, it can be noted that globally the FAR approach achieves competitive results for action recognition.

### Computational cost

In comparison to state-of-the-art approaches, our method has the advantage to present very low complexity. While LARP+TSRVF and LARP+mfPCA have a polynomial complexity, LARP+FA-R requires only a complexity of \( O(N) \), with \( N \) being the number of points forming the curve. This is explained by the fact that FAR is applied independently to each trajectory without the use of any reference, in contrast to other approaches which rely mainly on optimization between pairs of curves. As an illustration, DTW (having a complexity of \( O(N^2) \)) and FAR are compared in terms of execution time on the UTKinect dataset. Although DTW is implemented in C, FAR remains almost two times faster (LARP + DTW is executed in 272.10 s, while LARP is executed in 164.72 s).

### Limitations and future works

It is important to note that the proposed approach in its current shape does not completely answer the equivalence formulation. For example, eliminating the variability by redefining the parameter space makes the calculation of a mean trajectory not straightforward in comparison to elastic approaches. Furthermore, the proposed functions might be affected by noisy data. Finally, an extension to closed curves could be important for some applications. All these points remain interesting to investigate.

### V. Conclusion

In this letter, a fast open curve reparametrization technique in Riemannian spaces, called Fast Adaptive Reparameterization (FAR), is introduced. Instead of reparametrizing curves with respect to a reference, this approach reparametrizes each curve independently, ensuring a complexity of \( O(N) \). This is done through the heuristic definition of a parametrization function depending on the application of interest. The FAR approach was applied to the LARP descriptor [22] in the domain of skeleton-based action recognition. This allowed evaluating our approach on curves in the Riemannian space \( SE(3)^m \). The obtained results on three well-known datasets show the effectiveness of our approach as compared to other state-of-the-art techniques for action alignment in terms of accuracy and computational complexity.

### ACKNOWLEDGMENT

The authors would like to thank R. Baptista and K. Papadopoulos for their help in enhancing this manuscript.
REFERENCES

[1] G. G. Demisse, D. Aouada, and B. Ottersten, “Similarity metric for curved
shapes in Euclidean space,” in Proc. IEEE Conf. Comput. Vision Pattern Recognit.,
2016, pp. 5042–5050.

[2] G. G. Demisse, D. Aouada, and B. Ottersten, “Deformation based curved
shape representation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40,
no. 6, pp. 1338–1351, Jun. 2018.

[3] A. Srivastava, E. Klassen, S. H. Joshi, and I. H. Jermyn, “Shape analysis
of elastic curves in Euclidean spaces,” IEEE Trans. Pattern Anal. Mach. Intell.,
vol. 33, no. 7, pp. 1415–1428, Jul. 2011.

[4] D. Aouada and H. Krin, “Squigraphs for fine and compact modeling of
3-D shapes,” IEEE Trans. Image Process., vol. 19, no. 2, pp. 306–321,
Feb. 2010.

[5] R. Malladi, J. A. Sethian, and B. C. Vemuri, “Shape modeling with front
propagation: A level set approach,” IEEE Trans. Pattern Anal. Mach. Intell.,
vol. 17, no. 2, pp. 158–175, Feb. 1995.

[6] M. Rousson and N. Paragios, “Shape priors for level set representations,”
in Proc. Eur. Conf. Comput. Vision, 2002, pp. 78–92.

[7] C. C. Tappert, “Cursive script recognition by elastic matching,” IBM J. Res.
Dev., vol. 26, no. 6, pp. 765–771, 1982.

[8] Y. Tohkura, “A weighted cepstral distance measure for speech recogni-
tion,” IEEE Trans. Acoust., Speech, Signal Process., vol.-ASSP 35, no. 10,
pp. 1414–1422, Oct. 1987.

[9] R. Anirudh, P. Turaga, J. Su, and A. Srivastava, “Elastic functional coding
of Riemannian trajectories,” IEEE Trans. Pattern Anal. Mach. Intell.,
vol. 39, no. 5, pp. 922–936, May 2017.

[10] E. Ghorbel, R. Boutteau, J. Boonaert, X. Savatier, and S. Lecoeuche,
“Kinematic spline curves: A temporal invariant descriptor for fast action
recognition,” Image Vision Comput., vol. 77, pp. 60–71, 2018.

[11] E. Ghorbel, J. Boonaert, R. Boutteau, S. Lecoeuche, and X. Savatier,
“An extension of kernel learning methods using a modified log-Euclidean
distance for fast and accurate skeleton-based human action recognition,”
Comput. Vision Image Understanding, vol. 175, pp. 32–43, 2018.

[12] R. Vemulapalli, F. Arrate, and R. Chellappa, “Human action recognition
by representing 3D skeletons as points in a Lie group,” in Proc. IEEE Conf.
Comput. Vision Pattern Recognit., 2014, pp. 588–595.

[13] J. Su et al., “Statistical analysis of trajectories on Riemannian manifolds:
Bird migration, hurricane tracking and video surveillance,” Ann. Appl.
Statist., vol. 8, no. 1, pp. 530–552, 2014.

[14] L. Younes, “Computable elastic distances between shapes,” SIAM J. Appl.
Math., vol. 58, no. 2, pp. 565–586, 1998.

[15] R. F. Vemulapalli, F. Arrate, and R. Chellappa, “Silhouette-based gesture and action recognition via modeling trajectories
on Riemannian shape manifolds,” Comput. Vision Image Understanding,
vol. 115, no. 3, pp. 439–455, 2011.

[16] M. F. Abdelkader, W. Abd-Almageed, A. Srivastava, and R. Chellappa,
“Silhouette-based gesture and action recognition via modeling trajectories
on Riemannian shape manifolds,” Comput. Vision Image Understanding,
vol. 115, no. 3, pp. 439–455, 2011.

[17] R. Murtagh, A Mathematical Introduction to Robotic Manipulation.
Boca Raton, FL, USA: CRC Press, 2017.

[18] B. Hall, Lie Groups, Lie Algebras, and Representations: An Elementary
Introduction, Berlin, Germany: Springer, vol. 222, 2015.

[19] M. Zezulka and V. Kumar, “Two methods for interpolating rigid body
motions,” in Proc. IEEE Int. Conf. Robot. Autom., 1998, pp. 2922–2927.

[20] L. Seidenari, V. Varano, S. Berrettì, A. Bimbo, and P. Pala, “Recognizing
actions from depth cameras as weakly aligned multi-part bag-of-poses,”
in Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops, 2013,
pp. 479–485.

[21] L. Xia, C.-C. Chen, and J. Aggarwal, “View invariant human action
recognition using histograms of 3D joints,” in Proc. IEEE Conf. Comput.
Vision Pattern Recognit. Workshops, 2012, pp. 20–27.

[22] L. Li, Z. Zhang, and Z. Liu, “Action recognition based on a bag of 3D
points,” in Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops,
2010, pp. 9–14.

[23] P. T. Fletcher, C. Lu, S. M. Pizer, and S. Joshi, “Principal geodesic analysis
for the study of nonlinear statistics of shape,” IEEE Trans. Med. Imag.,
vol. 23, no. 8, pp. 995–1005, Aug. 2004.