Encoding Human Actions with a Frequency Domain Approach

Dharmil Shah, Pietro Falco, Matteo Saveriano, and Dongheui Lee

***Abstract*— In this work, we propose a Frequency-based Action Descriptor (FADE) to represent human actions. In robotics, with the development of Programming by Demonstration (PbD) methods, representing and recognizing large sets of actions has become crucial to build autonomous systems that learn from humans. The FADE descriptor leverages Fast Fourier Transform (FFT) for action representation and is combined with the Manhattan distance for measuring similarities between actions. It is characterized by a low time and space complexity and is particularly suitable for classification of human actions. For clustering problems, we propose a modified version of FADE, called Uncompressed-FADE (U-FADE), which performs well in combination with Spectral Clustering algorithms at the price of a reduced compression. We compare FADE with action descriptors based on Singular Value Decomposition (SVD) and Hidden Markov Models (HMM) on the entire HDM05 motion capture database. Despite the high dimensionality of the problem, we obtained on the entire database a promising recognition rate of 78% combining FADE with a simple 1-NN classification algorithm. Furthermore, we achieved a rate of 98% on a small action set and 88% on a medium action set.**

I. INTRODUCTION

With the development of Programming by Demonstration (PbD) paradigms, observation, automatic segmentation and recognition of human actions has become a problem of key importance in the field of robotics and autonomous systems. In order to achieve a widespread usage of robotic systems in service and domestic environments, a key objective is to develop methods that allow nonexpert users to program robots in an intuitive fashion. One of the most promising solution is to provide robots with the ability to observe, interpret, and imitate human actions [1]. By observing human motion and being able to detect new actions, robots can incrementally learn by imitation during their entire life cycle. Robotic systems working in domestic and industrial environments require the capability to learn a number of motion primitives that is theoretically unlimited. In practice, robots with lifelong-learning capabilities will be provided with databases of segmented actions organized in a large numbers of classes and actions. In order to maintain such databases, actions must be represented by suitable descriptors in a compact form, which is also computationally efficient. When moving from laboratory environments to real applications, the scalability of action representation and classification becomes a key issue. As a consequence, particular attention needs to be paid to computational complexity and to the dimensionality of the action descriptors. Representation and recognition of human whole body action is a complex problem for diverse reasons:

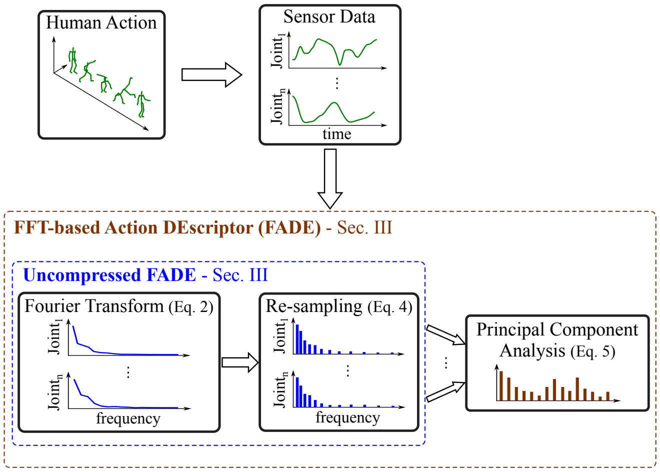
Human Action
Sensor Data
Joint$_1$
Joint$_n$
time
FFT-based Action DEscriptor (FADE) - Sec. III
Uncompressed FADE - Sec. III
Fourier Transform (Eq. 2)
Joint$_1$
Joint$_n$
frequency
Re-sampling (Eq. 4)
Joint$_1$
Joint$_n$
frequency
Principal Component Analysis (Eq. 5)

Fig. 1. Overview of FADE approach

- The dimensionality of the problem is high, due to the high number of human body DoFs
- The time length of actions is highly variable, even with very similar actions
- Representations that are not highly-compressed induce a heavy curse of dimensionality during the classification step

In robotics, moreover, there are additional issues. The action databases are likely characterized by examples split in a high number of classes. In this scenario, novel descriptors are needed that allow a fast and compact representation and, at the same time, show the capability to distinguish among different actions with a high number of different classes. Dealing with large datasets, the time and space complexity of the encoding process becomes a key issue.

In order to deal with this challenge, we propose FADE (FFT-based Action DEscriptor). FADE is a simple descriptor based on frequency analysis of the observed human motion signals.

As shown in Fig. 1, the signals are converted into the frequency domain through the Fast Fourier Transform (FFT) [2] and are resampled in a convenient way to exploit the properties of human motion in the frequency domain. We leverage, in particular, the property that human motion has frequency components only in a small range, typically $0 - 10\,\text{Hz}$ [3]. To reduce the dimensionality further, Principal Component Analysis (PCA) is applied in the frequency domain. In order to investigate accuracy and scalability properties, we tested FADE, combined with different supervised and unsupervised classification algorithms, on small, medium, and large-size datasets of the HDM05 database [4]. The small dataset

Dharmil Shah and Pietro Falco contributed equally to this work.

The authors are with the Chair of Automatic Control Engineering, Technical University of Munich, Germany. E-mail: {dharmil.shah, pietro.falco, matteo.saveriano, dhlee}@tum.de

includes 10 action classes and 100 actions, the medium-size dataset includes 78 action classes and 497 actions, the large dataset includes the entire HDM05 database, which is composed by 2337 actions and 80 classes. In the medium-size dataset, all the actions are performed by the same actor, while in the entire HDM05 dataset there are 5 actors. The medium-size dataset is also called Le Naour (LN) dataset, since it has been used for the first time by Le Naour, Courty, and Gibet in [5]. The classification of the entire HDM05 dataset is then much more challenging because of the increased number of actions and the increased variability due to different performers.

In this work, the actions are observed as a set of joint angle trajectories. In computed vision community, however, the recent trend is to leverage the accurate knowledge of the human skeletal models [6], [7]. These approaches perform very well in terms of accuracy, however they use high-quality information that is not easily available in service and domestic robotic applications, i.e. accurate measurements of marker positions, and the accurate skeletal model of the performer. In laboratories equipped with expensive motion capture systems, it is feasible to provide measurements of the performer body and to derive quite an accurate skeletal model. However, in unstructured environments, visual estimation and tracking of the entire human kinematic chain are not guaranteed, especially when the human is interacting with objects, other humans, or with the robot itself. In order to alleviate this problem, research on low-cost wearable devices to measure human joint angles, such as Inertial Measurement Units (IMU) and Accelerometers, is gaining great interest [8], [9], [10]. Such devices allow measuring joint angle positions and do not require any skeletal model to observe human movement. Recent papers proposed descriptors based on joint angle signals [11], [12]. Even though it is well-known that recognizing actions with only joint angle signals is more challenging [11], research on the topic is still very active. The main reason is that joint angle signals are natively invariant to body roto-translations and are measurable with several types of sensing systems. Although FADE can be potentially applied to skeleton-based features, in this work we propose and leverage FADE to encode actions described by joint angles.

II. RELATED WORK

Kulic et al. proposed an unsupervised method based on Hidden Markov Models [13] to represent and cluster actions [14], [15]. This approach introduces the possibility to learn a number of actions in an unsupervised incremental fashion. Recently Takano and Nakamura [16] combined a HMM-based clustering method with a Real-time Unsupervised Segmentation method (RUS). The approach presented in [17] leverages Singular Value Decomposition (SVD) to represent human manipulation actions, Euclidean distance to measure the similarity among actions, k-means and k-NN for off-line and online clustering respectively. A more recent approach to segmentation and classification uses a neighborhood graph to segment the motion and to define motion primitives. Primitives are seen as repetition in the trajectories observed from humans. Zhou et al. proposed the Hierarchical Aligned Cluster Analysis (HACA) [18], which combines kernel k-means with generalized dynamic time alignment kernel to detect repetitions in a data streaming. A template-based approach to recognize actions [19] uses a small set of a-priori known actions called templates. To align observed actions with the example actions, the Dynamic Time Warping (DTW) is adopted [20]. Although this method is computationally expensive, it achieves a recognition rate of 98% for 9 classes in the HDM05 database. In [21], a local skeleton descriptor is proposed that encodes the relative position of joint quadruples. The input data are joint Cartesian coordinates. The authors achieve 94% accuracy on a subset of HDM05 constituted by 11 classes. An interesting paper, which considers frequency domain [22], exploits ensembles models learnt to represent each action and to capture the intra-class variance. The method shows promising results in dealing with data from depth cameras. The approach is supervised and it uses a Support Vector Machine (SVM) training method [23]. Compared with our approach, the method proposed in [22] adopts a different descriptor, which is based on pair-wise joint relative positions and it uses input data based on joint Cartesian position. It clearly shows how information in the frequency domain can be valuable in human action recognition.

One of the main limitations of the the state-of-the-art approaches is the scalability to a large number of actions and classes [6]. The scalability is difficult to achieve because of diverse problems: potential complexity in the representation of actions, potential complexity in computing distances between actions, difficulty to differentiate actions in presence of a high numbers of classes, and heavy curse of dimensionality in the classification process. Chen [6] proposed a method to alleviate this problem. The method leverages a skeleton-based action descriptor and extreme Learning Machines (ELM) for classification. The descriptor is defined as skeleton-based (or model-based) because it requires the knowledge of the skeletal model of the performer to obtain a user-independent normalized representation. It achieves up to 96% on 40 classes consisting mainly in stationary actions of the HDM05 database. Using the same skeleton-based features, in [7] a deep learning Neural Network is proposed to classify motion capture sequences. It achieves an accuracy higher than 90% on 2337 actions of HDM05 grouped in 65 classes. Despite its good performance on action recognition, deep learning requires a massive amount of training data and long training time. To compute the action representation, an accurate estimation of the skeleton model is required. Since these requirements are undesirable in many robotic applications, we aim at developing a motion descriptor which is suitable for online incremental learning and which does not require an accurate skeletal model of the actor.

Ofli et al. suggested the SMIJ (Sequence of the Most Informative Joint) for action recognition [24], which is based on ranking the informative joints involved in an action [12]. In particular, the set of joints that present the maximum

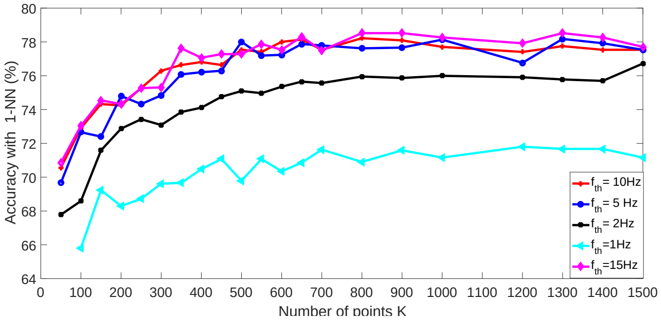

Fig. 2. Accuracy with 1-NN as a function of number of points K for different values of the frequency threshold f_{th}

variance during the motion are considered most informative. The approach was tested on 16 actions in the HDM05 database and on 11 actions in the Berkeley MHAD database [24]. On 11 actions of HDM05, the authors reach an accuracy of 84% with a supervised learning approach.

In [5] the authors proposed a representation that exploits the pair-wise joint-to-joint distances in the skeletal model. Afterwards, the dimensionality is reduced by Principal Component Analysis (PCA). The descriptor is associated to a 2-NN method to classify the actions. The approach proposed in [5] is tested on a set of classes that is larger compared to most state-of-the-art works. The dataset is constituted by a large part of the HDM05 database, which consists of 497 actions and 78 classes. Another interesting feature is that such a dataset presents a high number of classes but a relatively small number of repetitions for each class. This situation can be realistic in robotics applications, where the user is supposed to provide the robot with a limited number of repetitions in the training process.

With respect to the papers in the literature, we focus on two particular key issues. The first is how to represent and classify the observed trajectories with a skeleton-free, computationally efficient, and low-dimensionality strategy. The second is the scalability of action representation and classification in terms of accuracy and in computational time. To take a step towards the solution of the first issue, we propose the FADE descriptor, which is fast to compute, presents a very low dimensionality, and exploits the effectiveness of frequency analysis, already proven in the signal processing domain. To show the scalability of FADE, we test the descriptor on three different action sets of HDM05: the small dataset (10 classes, 100 actions), the Le Naour's dataset (LN) [5] (78 classes, 497 actions), and the complete HDM05 database (80 classes, 2337 actions).

III. PROPOSED APPROACH

The proposed strategy for human action classification can be divided in three tightly connected problems: action representation, similarity metrics between actions, and class selection.

A. Action Representation

We define the FADE action representation as the function $f : \mathcal{A} \rightarrow \mathbb{R}^m$, where $\mathcal{A}$ is the set of all the human actions. In this work each action $A \in \mathcal{A}$ is associated to a matrix $\boldsymbol{A}_t$ of the form

$$\boldsymbol{A}_t = \begin{pmatrix} q_1(t_1) & q_2(t_1) & \dots & q_J(t_1) \\ q_1(t_2) & q_2(t_2) & \dots & q_J(t_2) \\ \dots & \dots & \dots & \dots \\ q_1(t_N) & q_2(t_N) & \dots & q_J(t_N) \end{pmatrix} \qquad (1)$$

The vector $\boldsymbol{q}_k = [q_1(t_k), \dots, q_J(t_k)]$ contains the joint angular values at the discrete time frame k. In our work, J is the number of joint angle signals. In HDM05 database we have $J = 56$. In this work the sampling frequency f_s is set to 60 Hz. In Eq. (1) we use the symbol $\boldsymbol{A}_t$ to signify that the matrix contains values in the time domain. The first step of FADE is to compute the Discrete Fourier Transform (DFT) of the time-domain signals. In order to compute the DFT, we leverage the Fast Fourier Transform algorithm. For more details about the FFT, refer to [2]. Applying the FFT algorithm we have:

$$\boldsymbol{A}_\omega = \mathrm{FFT}(\boldsymbol{A}_\mathrm{t}) \qquad (2)$$

The columns of the matrix $\boldsymbol{A}_\omega$ contains the FFT of the columns of the matrix $\boldsymbol{A}_t$

$$\boldsymbol{A}_\omega = \begin{pmatrix} q_1(\omega_1) & q_2(\omega_1) & \dots & q_J(\omega_1) \\ q_1(\omega_2) & q_2(\omega_2) & \dots & q_J(\omega_2) \\ \dots & \dots & \dots & \dots \\ q_1(\omega_N) & q_2(\omega_N) & \dots & q_J(\omega_N) \end{pmatrix} \qquad (3)$$

Since the human motion does not contain significant frequency components beyond 10 Hz [3], we can remove the values above a given threshold f_{th}, with $f_{th} \leq 10$ Hz. As shown in Fig. 5, we found empirically that $f_{th} = 5$ Hz guarantees the same performance as $f_{th} = 10$ Hz. Since the pulse ω is related to the frequency f by the relation $\omega = 2\pi f$, the pulse threshold will be $\omega_{th} = f_{th}/2\pi$. In order to have a consistent representation for all the actions of the set $\mathcal{A}$, we remove the frequencies bigger than ω_{th} and resample the data with the sampling rate in the frequency domain $\Delta\omega$. In the standard FADE representation, we use a constant value for $\Delta\omega$. In this work, we used $\Delta f = 0.010$ Hz. As a consequence, we have $\Delta\omega = 0.020/2\pi$ rad/s. The set of all the selected sampling frequencies is denoted as Ω_K, where K is the number of points in the frequency domain sampled by the DFT of each joint signal. In this work, we have chosen $K = 500$. This value offers a good trade-off between accuracy and computational time.

After computing the sampling point in the frequency domain we obtain the matrix

$$\boldsymbol{A}_{\tilde{\omega}} = \begin{pmatrix} q_1(\tilde{\omega}_1) & q_2(\tilde{\omega}_1) & \dots & q_J(\tilde{\omega}_1) \\ q_1(\tilde{\omega}_2) & q_2(\tilde{\omega}_2) & \dots & q_J(\tilde{\omega}_2) \\ \dots & \dots & \dots & \dots \\ q_1(\tilde{\omega}_{th}) & q_2(\tilde{\omega}_{th}) & \dots & q_J(\tilde{\omega}_{th}) \end{pmatrix} \qquad (4)$$

Given the matrix $\boldsymbol{A}_{\tilde{\omega}}$, we compute the Principal Component Analysis (PCA) on the matrix $\boldsymbol{A}_{\tilde{\omega}}$, in order to maximize the compression of our descriptor:

$$\boldsymbol{V}_{\tilde{\omega}} = \mathrm{PCA}(\boldsymbol{A}_{\tilde{\omega}}) \qquad (5)$$

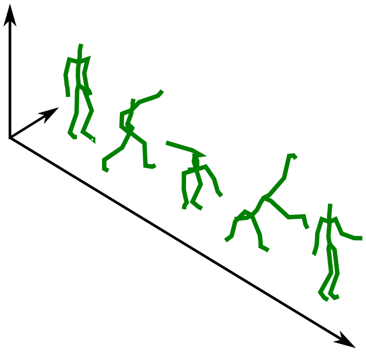

Fig. 3. Visualization of a selected action: cartwheel

The matrix $\boldsymbol{V}_{\tilde{\omega}}$ contains the PCA coefficients and has dimension $J \times J$, where J is the number of joints. To derive the FADE representation, we choose the first column of the matrix $\boldsymbol{V}_{\tilde{\omega}}$ and denote it as $\boldsymbol{v}$. The vector $\boldsymbol{v}$ is then the FADE representation of the action $\boldsymbol{A}_t$ and we can use the notation: $\boldsymbol{v} = \text{FADE}(\boldsymbol{A}_{\text{t}})$. The dimension of our descriptor is then $J \times 1$. A sequential description of the representation procedure is described in Algorithm 1. We have chosen PCA for compressing our descriptor because (i) it is a very mature and well-known technique, (ii) it is very easy to find optimized software implementations in most programming languages, (iii) it can be applied for both supervised and unsupervised learning approaches, since PCA does not require data labels, and (iv) it does not require a training phase like, for example, neural autoencoders. As shown in Sec. IV, for some unsupervised algorithms like Spectral Clustering (SC), it is convenient to use an uncompressed version of FADE that we called U-FADE. The main reason is that Spectral Clustering computes internally a Singular Value Decomposition, exploiting information on eigenvectors to cluster the actions. To derive U-FADE, we simply reshape the matrix $\boldsymbol{A}_{\tilde{\omega}}$ into a $J \times K$ column vector, where K is the number of points in the frequency domain. Algorithm 2 reports the procedure to derive U-FADE. In Fig 3 an example of the HDM05 action "cartwheel" is shown. Fig. 4 show the joint angular signals of the same action in the time domain. In Fig. 5 the FFT of the matrix $\boldsymbol{A}_t$ relative to the action cartwheel is computed. Each signal depicted in Figures 4 and 5 is associated to a color and represents one of the 56 joint angles as a function of the time and of the frequency respectively. From Fig. 5 it is evident how the significant information about a complex action like cartwheel is all contained in the range $0-5$ Hz. For any action or automatically segmented primitive of any time duration, the significant information is contained always in a very limited interval of frequencies. Figure 6 shows the values of FADE descriptors for the actions cartwheel, walk, and punch. It is evident how walk and cartwheel are encoded with more FADE components, while punch presents high values on a limited number of components. From the figure it can be seen that FADE is able to encode different actions into clearly distinct patterns.

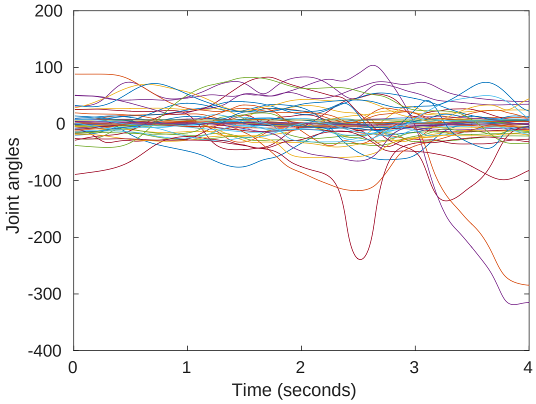

Fig. 4. Joint angular values in the time domain of the action cartwheel

The asymptotic complexity of FADE and U-FADE as a function of the time-frame number is $O(n \log n)$. To derive such complexity we consider the steps of Algorithm 1. In line 2 we compute the FFT, whose $O(.)$ complexity is $O(n \log n)$. In line 3 we resample the signal in the frequency domain stopping at f_{th}. Using linear interpolation for resampling the complexity is $O(n)$. Line 4 computes the PCA of the matrix $\boldsymbol{A}_{\tilde{\omega}}$. Since the dimension of $\boldsymbol{A}_{\tilde{\omega}}$ is fixed and does not depend on n, we have a $O(1)$ computational cost. The total cost is then $O(n \log n)+O(n)+O(1)$, that is asymptotically equal to $O(n \log n)$. The dimensionality is only J for FADE and $K \times J$ for U-FADE. The number of frequency domain points K does not depend on the number of frames in the time domain. As a consequence, after fixing the number of joints, both FADE and U-FADE have $O(1)$ for memory requirements. As shown in Sec. IV, with such a high compression rate, FADE shows good accuracy and low computational cost.

In order to give an example on how K and f_{hz} affect the performance of the recognition pipeline, we plot the performance of FADE, combined with 1-NN and Manhattan distance, evaluated on the whole HDM05 dataset. It is possible to notice that $f_{th} = 5\,\text{Hz}$, $f_{th} = 10\,\text{Hz}$, and $f = 15\,\text{Hz}$ show similar performance. As expected,

Algorithm 1 FADE Action Representation

```
1: $\boldsymbol{v} = \text{FADE}(\text{ActionMatrix}\,\boldsymbol{A}_{\text{t}})$
2: $\boldsymbol{A}_{\omega} = \text{FFT}(\boldsymbol{A}_{\text{t}})$
3: $\boldsymbol{A}_{\tilde{\omega}} = \text{resample}(\boldsymbol{A}_{\omega}, \Omega_{\text{K}})$
4: $\boldsymbol{V}_{\tilde{\omega}} = \text{PCA}(\boldsymbol{A}_{\tilde{\omega}})$
5: $\boldsymbol{v} = \boldsymbol{V}_{\tilde{\omega}}(:,1)$ //select the first column
6: return $\boldsymbol{v}$
```

Algorithm 2 U-FADE Action Representation

```
$\boldsymbol{v}_U = \text{UFADE}(\text{ActionMatrix}\,\boldsymbol{A}_{\text{t}}\ )$
$\boldsymbol{A}_{\omega} = \text{FFT}(\boldsymbol{A}_{\text{t}})$
$\boldsymbol{A}_{\tilde{\omega}} = \text{resample}(\boldsymbol{A}_{\omega}, \Omega_{\text{K}})$
$\boldsymbol{v}_U = \text{reshape}(\boldsymbol{A}_{\tilde{\omega}}, \text{K} \cdot \text{J}, 1)$
return $\boldsymbol{v}$
```

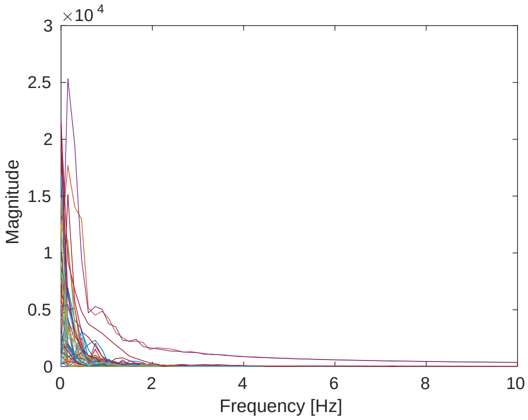

Fig. 5. FFT of the joint angular values of the action cartwheel

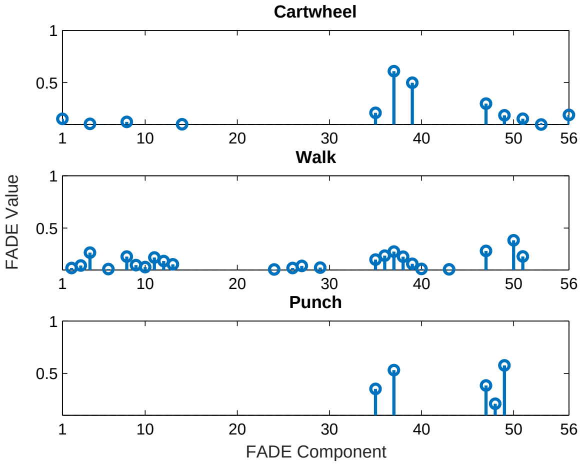

Fig. 6. FADE descriptors of three actions: cartwheel, walk, and punch.

frequencies beyond 10 Hz do not add valuable information. Reducing the frequency threshold to $2\,Hz$ and to 1 Hz, the accuracy decreases for every value of K. In terms of number of points K, the accuracy increases significantly before a threshold. Increasing the number of points K after a certain threshold, the improvement of the accuracy becomes minor. The threshold is around $K = 500$. The number of points affects the computational cost. When limited computational power is available, it can be important to choose a trade-off between accuracy and computational cost.

We found that FADE is particularly convenient to represent and classify segmented actions due to the following properties:

Human motion is bounded at a low frequency: Signals captured from human actions have a narrow frequency spectrum: low-pass filter with a bandwidth bound width around 10 Hz.

Representation independent from the time duration of an action: We can represent all the actions with the same number of points in the frequency domain, independently from the duration of the motion.

Efficient algorithms to compute Fourier Transform: There are efficient algorithms to compute the Fourier Transform such as Fast Fourier Transform (FFT). FFT is well-known for general signal processing and it is easy to implement in CPU for wearable devices. Its computational complexity is $\mathrm{O}(n \log n)$.

Suitable both for single and cyclic motion: We can recognize both one repetition of an action and cyclical repetitions with the same representation.

Robustness to noise: FADE takes into account only a well-known frequency interval where information of human motion is contained [3]. As a consequence, it is more robust to measurement noise and has ”native” filtering properties.

Good performance with low dimensionality: The dimensionality of the compressed FADE is only $1 \times J$. Despite such a compression rate, we achieve a scalable classification accuracy with most popular classification methods.

B. Similarity Between Actions

As described in Sec. III, FADE allows representing a matrix time sequence of dimension (N, J) into a vector of dimension $J \times 1$. To measure the similarity among actions, we evaluated different distance measures: Euclidean, Cosine, Manhattan, and Mahalanobis. We observed that the Manhattan distance performed best to measure similarity among FADE representation of human actions. Our empirical observation is confirmed by [25], which shows how Manhattan distance performs better with high dimensional data. The Manhattan distance between two vectors $\boldsymbol{x} = (x_1, x_2, ..., x_n)$ and $\boldsymbol{y} = (y_1, y_2, ..., y_n)$ is computed by the following equation:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{n} |x_i - y_i| \right). \tag{6}$$

C. Action Classification and Clustering

In order to test FADE in both classification and clustering problems, we combine FADE and U-FADE with popular supervised and unsupervised approaches. Concerning supervised methods, we considered k-Nearest-Neighbors (k-NN) and Support Vector Machines (SVM). Since we are particularly interested in simplicity and low computational cost, we combined FADE with 1-NN and 2-NN. SVMs, on the other hand, are well-known for their effectiveness in supervised classification, even if the training process is more computationally expensive. We compare k-NN and SVM to evaluate how much we gain in accuracy by accepting a longer training time.

Regarding clustering approaches, we investigate Spectral Clustering (SC), K-Means (KM), and Agglomerative Clustering (AC) [23]. SC and KM are mature state-of-the-art approaches for data mining applications. The main limitation of such approaches is that they require as an input the maximum number of clusters. In order to alleviate this issue, we tested AC [23], which does not require such a parameter. Unsupervised classification of segmented human actions is a challenging research problem, especially when scaling from a dozen actions and few classes to hundred classes and thousand actions. The return in developing new strategies for unsupervised clustering of big action datasets will have a high impact in robotics. Clustering approaches can provide autonomous systems with the capability to interpret and learn diverse classes of actions without the direct supervision of humans. In particular, a robotic system can autonomously observe humans, encode the segmented actions with FADE, and automatically cluster the observed actions without any human intervention.

IV. EXPERIMENTAL RESULTS

A. Comparisons

Among different methods for skeleton-free action representation, we decided to compare FADE with representations based on SVD and HMM [23]. Both are mature approaches that can be adopted for action representation with different segmentation and recognition strategies.

1) SVD-based descriptors: In order to derive a descriptor based on SVD [17], let us compute the Singular Value Decomposition of the matrix $\boldsymbol{A}_t$, where $\boldsymbol{A}_t$ is defined in Eq. (1). It is:

$$(\boldsymbol{U}_A, \boldsymbol{\Sigma}_{\mathbf{A}}, \boldsymbol{V}_{\mathbf{A}}) = \mathrm{svd}(\boldsymbol{A}_{\mathrm{t}}) \tag{7}$$

According to the properties of SVD, matrices $\boldsymbol{U}_A, \boldsymbol{V}_A$ belong to $\mathbb{R}^{n\times m}$, where m is the number of DOFs and n is the number of time frames. The dimension of the matrix $\boldsymbol{V}_A$ is $m \times m$. As a consequence, it does not depend on the number of time frames. A SVD descriptor, as proposed in our previous work [17], can be obtained by extracting the first column of the matrix $\boldsymbol{V}_A$.

2) HMM-based descriptors: An HMM represents spatiotemporal variations of motion trajectories as a set of S hidden states. It is described by the set of parameters $\boldsymbol{\lambda}$, learned from motion trajectories using the Baum-Welch algorithm [13]. In case of continuous time input, the output probability is represented as a mixture of M Gaussians. We tried the mixture of 1, 2, 5, and 8 Gaussians. In our analysis, the accuracy decreases when the number of Gaussians increases. This is due to the relatively limited number of points per state (20 per state on average). As a consequence, in this work we use one Gaussian for each state, i.e. $M = 1$. A left-to-right HMM, where state transitions occur only between two consecutive states [13], is the most suitable to represent actions. Human actions, in fact, have a starting and an ending point. The procedure to classify incoming actions requires two steps. First, an HMM for the new action is learned. Second, the action is assigned to the closest class using the HMM distance. The distance between two HMMs is computed using the Kullback-Leibler distance:

$$d_{12} = \frac{1}{T}\left[logP(O_2|\boldsymbol{\lambda}_1)) - logP(O_2|\boldsymbol{\lambda}_2))\right] \tag{8}$$

where $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$ are two models, O_2 is an observation sequence generated by $\boldsymbol{\lambda}_2$ [13] and T is the length of O_2. Since d_{12} in (8) is not symmetric, the distance between two HMMs is computed as $d_{1,2}^{hmm} = d_{2,1}^{hmm} = (d_{12} + d_{21})/2$. In order to ensure that the representation is implemented in an optimal way, we selected the optimal number of states by exploiting the Akaike Information Criterion (AIC) [26]. In order to reduce the search space of AIC, we limit the maximum number of states to 15.

Representation	Time Complexity	Space Complexity
FADE	$O(n \log n)$	$O(1)$
SVD	$O(n^2)$	$O(1)$
HMM	$O(n^2)$	$O(s)$

TABLE I
TIME AND SPACE COMPLEXITY OF THE ENCODING PROCEDURE AS A FUNCTION OF NUMBERS OF FRAMES IN THE TIME DOMAIN

B. Analysis of Space and Time Complexity

We report in Table I the time (computational) and space complexity of the action encoding procedure as a function of the number of frames n. We consider the number of joints to be fixed to J. As described in Section III, the computational complexity of FADE is $O(n \log n)$, while the dimension is J and it does not depend on the the number of frames. The spatial complexity is then $O(1)$. The time complexity of U-FADE is $O(n \log n)$, while the spatial complexity is $J \times K$. Since both K and J remain constant when n increases, U-FADE has a spatial complexity $O(1)$. Concerning SVD, we have a complexity of $O(n^2)$, since we have to compute the SVD on the entire time-domain sequence [23]. The spatial complexity for SVD is constant and it is equal to J. Therefore the space complexity does not depends on n. According to Sec. IV-A.2 and to [13], the time complexity of HMM is in our case $O(n^2)$ [13], while the space complexity is $O(S)$, where S in the number of states selected to encode the actions. If we choose a number of states that depends on the length of the action, we have a space complexity of $O(n)$.

C. Comparative Results

In order to show the performance of our approach on datasets of different sizes, we have chosen three sets of data:

- *Small action subset* (Table II). We picked up 10 classes and 100 actions of HDM05
- *Le Naour's subset* (Table III) We considered the dataset adopted in [5], which contains 78 classes and 497 actions of the HDM05 database. These actions were performed by a single actor.
- *Entire HDM05 dataset* (Table IV) We used all the actions and the classes in the HDM05 dataset without removing lower quality actions. The actions were performed by five different actors.

For each data set we evaluated how FADE performs with supervised and unsupervised classification methods. In particular, we compared FADE, U-FADE and SVD combined with the classification approaches described in Section IV-A.

In Table II, the results for the small action set are shown. In this dataset, FADE outperforms both SVD and HMM

Representation	Classification	Type	Accuracy(%)
FADE	1-NN	Supervised	89.0
FADE	2-NN	Supervised	87.0
FADE	SVM	Supervised	89.0
U-FADE	1-NN	Supervised	96.0
U-FADE	2-NN	Supervised	95.0
U-FADE	**SVM**	Supervised	**96.0**
SVD	1-NN	Supervised	84.0
SVD	2-NN	Supervised	81.0
SVD	SVM	Supervised	84.0
HMM	HMM	Supervised	90.0
FADE	SC	Unsupervised	80.0
FADE	K-Means	Unsupervised	76.0
FADE	AC	Unsupervised	79.0
U-FADE	**SC**	Unsupervised	**98.0**
U-FADE	K-Means	Unsupervised	82.0
U-FADE	AC	Unsupervised	88.0
SVD	SC	Unsupervised	60.0
SVD	K-Means	Unsupervised	61.0
SVD	AC	Unsupervised	62.0

TABLE II

RESULTS FOR THE SMALL ACTION SET FOR SUPERVISED APPROACHES (TOP) AND UNSUPERVISED APPROACHES (BOTTOM)

Representation	Classification	Type	Accuracy(%)
FADE	1-NN	Supervised	86.5
FADE	2-NN	Supervised	84.9
FADE	**SVM**	Supervised	**88.0**
U-FADE	1-NN	Supervised	81.5
U-FADE	2-NN	Supervised	85.9
U-FADE	SVM	Supervised	85.9
SVD	1-NN	Supervised	84.7
SVD	2-NN	Supervised	81.4
SVD	SVM	Supervised	85.0
HMM	HMM	Supervised	82.5
FADE	SC	Unsupervised	59.8
FADE	K-Means	Unsupervised	59.2
FADE	AC	Unsupervised	63.2
U-FADE	**SC**	Unsupervised	**67.2**
U-FADE	K-Means	Unsupervised	59.1
U-FADE	AC	Unsupervised	58.8
SVD	SC	Unsupervised	54.7
SVD	K-Means	Unsupervised	53.5
SVD	AC	Unsupervised	57.1

TABLE III

RESULTS FOR THE LN ACTION SET PROPOSED IN [5] FOR SUPERVISED APPROACHES (TOP) AND UNSUPERVISED APPROACHES (BOTTOM)

based descriptors in terms of accuracy and in terms of computational complexity. The time complexity of FADE-based encoding is $O(n \log n)$. For the SVD approach it is $O(n^2)$, and for HMM it is $O(n^2)$, where n is the number of the frames in the time domain. It is interesting to note how with a relatively small number of actions the unsupervised methods achieve good performance, especially the Spectral Clustering algorithm. When dealing with a relatively small number of actions, the benefits of compression are not directly visible. U-FADE reaches 98% with spectral clustering and 88% with agglomerative clustering. Using SVM, U-FADE achieves 96% accuracy. For supervised methods, we adopted a 10-fold cross-validation approach.

Representation	Classification	Type	Accuracy(%)
FADE	1-NN	Supervised	77.4
FADE	2-NN	Supervised	68.2
FADE	**SVM**	Supervised	**79.0**
U-FADE	1-NN	Supervised	74.5
U-FADE	2-NN	Supervised	70.9
U-FADE	2-SVM	Supervised	74.1
SVD	1-NN	Supervised	75.1
SVD	2-NN	Supervised	65.9
SVD	SVM	Supervised	73.9
HMM	HMM	Supervised	63.5
FADE	SC	Unsupervised	37.5
FADE	K-Means	Unsupervised	37.0
FADE	AC	Unsupervised	36.2
U-FADE	**SC**	Unsupervised	**46.6**
U-FADE	K-Means	Unsupervised	40.4
U-FADE	AC	Unsupervised	39.2
SVD	SC	Unsupervised	34.0
SVD	K-Means	Unsupervised	33.3
SVD	AC	Unsupervised	34.4

TABLE IV

RESULTS ON THE ENTIRE HDM05 DATABASE, FOR SUPERVISED APPROACHES (TOP) AND UNSUPERVISED APPROACHES (BOTTOM)

Table III summarizes the results obtained with the action set adopted in [5], which is denoted as LN dataset. It is composed by 78 classes and 497 actions. FADE reaches a recognition rate of 88% with SVM and 87% with 1-NN, cross-validated with 10-fold. Despite the lowest computational complexity, it outperforms SVD in all the approaches and unsupervised approaches we considered. This larger dataset shows how the scalability of unsupervised methods is limited. With FADE we reach 63% with Agglomerative Clustering (AC) and 60% with K-means. Considering that AC is unsupervised and it does not even require the number of clusters as an input parameter, the result can be considered promising. Concerning Spectral Clustering (SC), we can observe that the extreme compression of FADE causes a small decrease in the performance with respect to U-FADE.

The results on the entire HDM05 are shown in Table IV. The entire dataset contains 80 classes and 2337 actions. With respect to the LN dataset, the complete HDM05 presents only 2 classes more but around 5 times the number of actions as it considers the actions performed by all the actors. This represents an interesting challenge on the scalability of action representation and classification methods. As shown in Table IV, FADE with 1-NN obtains a recognition rate of 77% and with SVM it achieves an accuracy of almost 80% compared to 88% on the LN dataset. Considering its simplicity and its compression rate, FADE shows good properties of scalability with supervised learning approaches. However, unsupervised approaches are less scalable and suffer more with high compression rate. U-FADE combined with SC performs almost 10% better than FADE with SC.

V. CONCLUSION AND FUTURE WORK

In this work we proposed FADE, a compact and computationally efficient frequency-based action descriptor. The computation of the descriptor has a complexity of $O(n \log n)$

and the dimensionality is only J, where n is the number of time-frames and J is the number of signals used to represent human motion.

FADE has been tested on three datasets of the HDM05 motion capture database: small (10 classes, 100 actions), medium (78 classes, 497 action) and the entire dataset (80 classes, 2337 actions). The dimension of FADE does not depend on the time duration of the action and can be used to encode single or multiple repetitions, and cyclical actions by leveraging the properties of the Fourier Transform. Having a reduced dimensionality is crucial to minimize the execution time of any learning algorithm.

We applied FADE with both supervised and unsupervised learning for classification of human actions. For unsupervised approaches, we introduced also U-FADE, a branch of FADE that performs better in combination with Spectral Clustering methods, at the price of a reduced compression. FADE with supervised methods presents good accuracy and scalability properties, while unsupervised methods show relatively low recognition rate with the entire HDM05 dataset. However, with the small dataset, unsupervised approaches perform very well. Concerning future work, the first step will be to investigate the performance of FADE with other motion databases, such as the Berkeley MHAD database [24], CMU Graphics Lab Motion Capture Database [27], and KIT Whole-Body Human Motion Database [28]. In this work, FADE has been applied to joint angles. However, especially in the computer vision community, a significant number of approaches work with joint Cartesian positions. Since the proposed descriptor is general, we can encode with FADE (and U-FADE) human actions described with joint 3D positions. We will investigate the performace of FADE in this case compared with state-of-the-art approaches like [6]. Another research direction will consist in combining FADE with skeleton-free invariant representations [29] of marker and torso movements.

VI. ACKNOWLEDGMENTS

This work has been partially funded by the Marie Curie Action LEACON, EU project 659265.

REFERENCES

[1] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[2] J. S. Walker, *Fast fourier transforms*. CRC press, 1996, vol. 24.

[3] N. Forestier and V. Nougier, "The effects of muscular fatigue on the coordination of a multijoint movement in human," *Neuroscience letters*, vol. 252, no. 3, pp. 187–190, 1998.

[4] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, "Documentation mocap database hdm05," Universität Bonn, Tech. Rep. CG-2007-2, June 2007.

[5] T. Le Naour, N. Courty, and S. Gibet, "Fast motion retrieval with the distance input space," in *Motion in Games*. Springer, 2012, pp. 362–365.

[6] X. Chen and M. Koskela, "Skeleton-based action recognition with extreme learning machines," *Neurocomputing*, vol. 149, pp. 387–396, 2015.

[7] K. Cho and X. Chen, "Classifying and visualizing motion capture sequences using deep neural networks," in *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, vol. 2. IEEE, 2014, pp. 122–130.

[8] K. J. ODonovan, R. Kamnik, D. T. OKeeffe, and G. M. Lyons, "An inertial and magnetic sensor based technique for joint angle measurement," *Journal of biomechanics*, vol. 40, no. 12, pp. 2604–2611, 2007.

[9] W. Quan, H. Wang, and D. Liu, "A multifunctional joint angle sensor with measurement adaptability," *Sensors*, vol. 13, no. 11, pp. 15 274–15 289, 2013.

[10] T. Seel, J. Raisch, and T. Schauer, "Imu-based joint angle measurement for gait analysis," *Sensors*, vol. 14, no. 4, pp. 6891–6909, 2014.

[11] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal, "Bio-inspired dynamic 3d discriminative skeletal features for human action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 471–478.

[12] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (smij): A new representation for human skeletal action recognition," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 24–38, 2014.

[13] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[14] D. Kulić, W. Takano, and Y. Nakamura, "Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains," *The International Journal of Robotics Research*, vol. 27, no. 7, pp. 761–784, 2008.

[15] D. Kulić, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura, "Incremental learning of full body motion primitives and their sequencing through human motion observation," *The International Journal of Robotics Research*, p. 0278364911426178, 2011.

[16] W. Takano and Y. Nakamura, "Real-time unsupervised segmentation of human whole-body motion and its application to humanoid robot acquisition of motion symbols," *Robotics and Autonomous Systems*, vol. 75, pp. 260–272, 2016.

[17] A. Cavallo and P. Falco, "Online segmentation and classification of manipulation actions from the observation of kinetostatic data," *Human-Machine Systems, IEEE Transactions on*, vol. 44, no. 2, pp. 256–269, 2014.

[18] F. Zhou, F. De la Torre, and J. K. Hodgins, "Hierarchical aligned cluster analysis for temporal clustering of human motion," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 3, pp. 582–596, 2013.

[19] D. Leightley, B. Li, J. S. McPhee, M. H. Yap, and J. Darby, "Exemplar-based human action recognition with template matching from a stream of motion capture," in *Image Analysis and Recognition*. Springer, 2014, pp. 12–20.

[20] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *Transactions on Acoustics, Speech, and Signal Processing*, pp. 43–49, 1978.

[21] G. Evangelidis, G. Singh, and R. Horaud, "Skeletal quads: Human action recognition using joint quadruples," in *ICPR 2014-International Conference on Pattern Recognition*, 2014.

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

[23] C. M. Bishop, "Pattern recognition," *Machine Learning*, 2006.

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

[25] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, *On the surprising behavior of distance metrics in high dimensional space*. Springer, 2001.

[26] K. P. Burnham and D. R. Anderson, "Multimodel inference understanding aic and bic in model selection," *Sociological methods & research*, vol. 33, no. 2, pp. 261–304, 2004.

[27] R. Gross and J. Shi, "The cmu motion of body (mobo) database," Tech. Rep., 2001.

[28] C. Mandery, Ö. Terlemez, M. Do, N. Vahrenkamp, and T. Asfour, "The kit whole-body human motion database," in *Advanced Robotics (ICAR), 2015 International Conference on*. IEEE, 2015, pp. 329–336.

[29] R. Soloperto, M. Saveriano, and D. Lee, "A bidirectional invariant representation of motion for gesture recognition and reproduction," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 6146–6152.