A motion recognition algorithm using polytopic modeling

Pierre Moreau[1], David Durand[1], Jérôme Bosche[1] and Michel Lefranc[2]

***Abstract*— People's movements say a lot about their activities. Whether it concerns sports, music (playing an instrument), at work, in re-education, each domain has its own specific moves. However, some of it, such as sports competition, need high-precision movements. Tools are available permitting to measure movements in all sectors. First, sensors are placed on different strategic points on the person's body that allow us to retrieve temporal datas from the body of the user. In this work, no camera is used for motion recognition in order to let the user free to go in different spaces. Lots of algorithms help for movement recognition such as deep learning, convolutional neural networks or dynamic modelling, but in most cases, cameras are used. So, our approach consist of two phases. First, model the users thanks to whole-body sensors and save characteristic movements. Second, we still use sensors, to model the test person to find characteristic movements depending on the activity.**

I. INTRODUCTION

In the last decade, the recognition of movements has come a long way, mainly thanks to the evolution of deep learning. Most research uses cameras or depth cameras to analyze movements, and use image recognition algorithms. In [1], two algorithms are used: a neuronal network to identify hand movements and a convolutional neural network to analyse body movements. A related project aims to operate hand gestures and uses it on identification print service for phones. They use algorithms to find differences in movement between each user. Another work [3] uses a siamese neural network for gesture recognition. First, the data is retrieved by sensors such as accelerometers and gyrometers in smartphones, then, gestures and actions are recognized with siamese neural network based functions which reject uncertain and parasitic decisions. In [4], depth-cameras are used to capture the joints of the person in front of the camera. The depth added to the classic RGB camera allows to differentiate the person from the background and to distinguish if an object is handled by the person. So, skeleton and depth map approach are used for, respectively, body movement recognition (action) and object interactivity recognition (activities). Each joint is delivered by its 3D coordinates (x, y, z) and transformed in shape. Finally, k-nearest neighbors are employed to classify all shapes. At the end, a movement is a position sequence and each person's position may be viewed like an image. Google created GoogLeNet, a neural network architecture codenamed Inception [7] to compete in the ImageNet Large-Scale Visual Recognition Challenge and its goal is to do image recognition as quickly as possible. This algorithm is based on convolutional neural network and is an estate of function such as "max pooling" which reduces the image by keeping most important data, convolution matrix and filter. This neural network contains a total of twenty-two layers. This network is built with an optimal local construction and repeated several times. At the end, each local function analyses the correlation statistics of the last layer and clusters them, then this is repeated with the next layer. The main problem of action recognition is to know when the movement starts and when its stops. In [8], authors propose to use convolutional neural networks for challenging this issue by using a sliding window approach. This model is composed of two parts : one to detect if there is a movement in the sliding window and one to detect which category the movement belongs to. The detector uses weighted-cross entropy loss to avoid false positives and the classifier is based on a stochastic gradient descent. Finally, they add a Single-time Activation function in order to have smaller reaction time and have one time recognition.

Initially, the idea of this project comes from the will to help neurosurgeons to diagnose Parkinson's disease. Indeed, [2] analyses the effect of levodopa on patients suffering from Parkinson's disease. This medicine is effective, but gives rise to motor complications (dyskinesia) and the treatment must be regularly modified. For the purpose of recovering data, movement trajectories were extracted from a video and poses are estimated with a deep learning algorithm. Finally, a random forest analyses the severity of levodopa-induced dyskinesia. Other work [5] retrieves data from a triaxial accelerometer arranged at six different locations on the body. This data is labelled by physicians off-line so that it can be used to train a neural network.

The e-moove's project aims at assisting people in many different sectors. That's why we present this paper as follows. Section II presents the context of the e-mOove project. In Section III, we present main results and also algorithms. Finally, section IV illustrates the results and the database of Florence University that we use to test our algorithms.

II. CONTEXT

A. The e-moove project

Various techniques exist to capture these movements such as photography, sensors or cameras. In the e-mOove project, only wearable sensors (gyrometers, accelerometers and magnetometers) are used. Placed in a suit, this allows to model users in order to identify characteristic movements.

*This work was supported by the company Elivie.

[1]Pierre Moreau, David Durand and Jérôme Bosche are with the Modeling, Information and Systems (MIS) Laboratory, University of Picardie Jules Verne, Amiens, France pierre.moreau, david.durand, jerome.bosche}@u-picardie.fr

[2]Michel Lefranc is with the neurosurgery department of University Hospital Center, Amiens-Picardie, France Lefranc.Michel@chu-amiens.fr

The main advantage is to use this unobtrusive suit in real conditions. Moreover, users can wear it for some days before retrieved data from sensors. Recovered data are treated off-line with algorithms, such as deep learning, convolutional neural network or dynamic modelling.

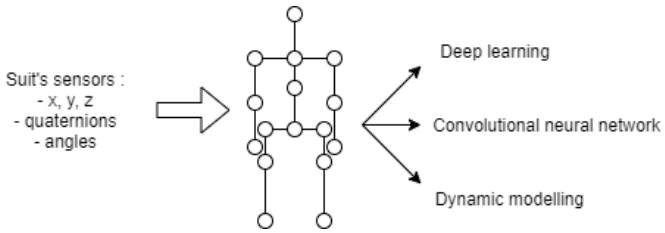

Fig. 1. Presentation of our approach

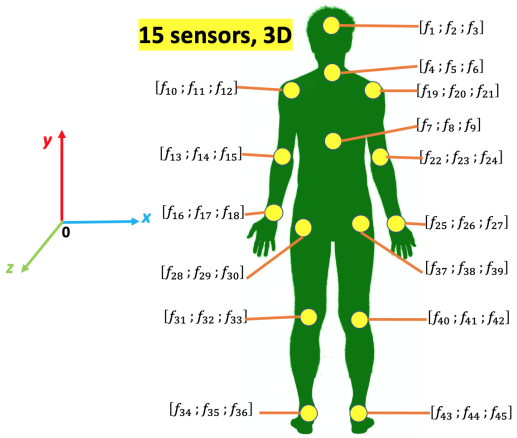

Fig. 2. 15 sensors connected jumpsuit

Sensors are used to get data from the user's body and these data are treated with algorithms in order to extract characteristic movements of an activity.
Whatever the algorithms, data must be labeled to identify the movement by comparing them. One of the main applications is the diagnosis of Parkinson's disease. This neurodegenerative disease implies problems of posture disorders such as tremors, trampling, falling or almost falling ... In this way, automatic detection of these characteristic movements, based on algorithms, would be must useful in helping the hospital practitioner to establish an accurate diagnosis. These algorithms must be trained on data from the same activity or disease, to find characteristic movements according associated with a type of Parkinson's. They can also be used in the field of sport, education, orthopedics, music...
Currently, data from the combination has not been retrieved yet. Many patients must wear the sensor's suit to create data and it takes time. Fortunately, the University of Florence [9] creates a database of different movements. That allow us to perform our algorithm as a first step even if it's not our prediction domain. Then, we will be able to train algorithms on our database from patients having Parkinson's disease.

B. The experimental context

As mentioned above, one of the originality - and constraint - of our approach consists in motion capture, only from sensors such as gyrometers or accelerometers. No image data from any camera is considered. In this work, it is assumed that the person is dressed with a special suit, equipped with several sensors, located on different parts of the human body. For example, a fifteen sensors combination is illustrated in Figure 2.

Each sensor allows the translation movement of the corresponding joint to be measured in 3D space: longitudinal movement (z), lateral movement (x) and vertical movement (y). According to this figure, the left wrist sensor will deliver 3 signals, denoted f_{25}, f_{26} and f_{27} corresponding to the wrist movement, respectively along the x, y and z axis. For the specific "wave gesture" corresponding to the left arm lifting, f_{25}, f_{26} and f_{27} signals are shown in Figure 3.

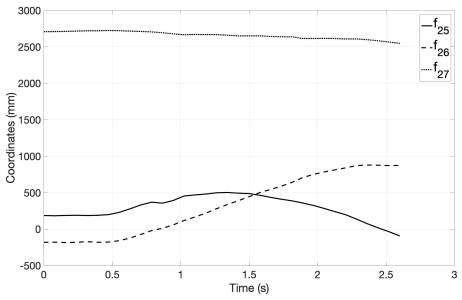

Fig. 3. World coordinates of the left wrist: *wave gesture*

From a database containing the measurements delivered by a number s of sensors for a set of m movements, the objective of this work is to propose an algorithm detecting a movement among m.
In general, the more data the device generates, the more precise the detection. On the other hand, the number of considered signals for the detection step has a direct impact on the computation time of the algorithm. This can be problematic in the case of real-time detection. Also, depending on the nature of the problem, a pre-analysis could consist in identifying and quantifying the relevant sensors for detection. For experimentation purpose, we use off-the-shelf dataset before grabbing data from the equipped suit.

III. Motion detection by polytopic modeling

In the literature, most motion recognition techniques use vision systems, such as cameras. Movements from video signals can then be represented in the form of curves and surfaces in a non-Euclidean multi-dimensional space. In this way, Riemannian manifolds have aroused great interest in recent years [10], [11]. In the context of the e-moove project, no vision system is used, only gyroscope and accelerometer type sensors. Also polytopic modeling is preferred to manifold representations.

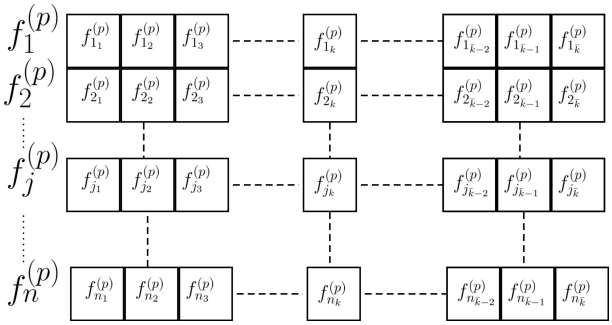

Fig. 4. Vector representation : experience p with n signals and $\bar{k}$ samples

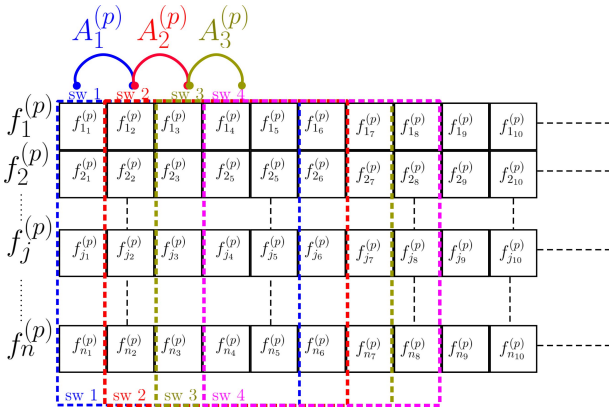

Fig. 5. The Sliding Window Modeling process

In this work, it is assumed that a number s of sensors allow the measurement of translation movements in 3D space. This means that $n = 3s$ time signals are delivered by the combination and must be used to characterize a particular movement. So, over a given acquisition period $T_a = \bar{k}T$ (T is the sampling period), the n functions f_i can be assimilated by a "standard" data vector representation, as shown Figure 4.

A. A discret state-space representation for learning

Let us consider the vector $X_k \in \mathbb{R}^{n\times 1}$ concanetating the n signals $f_j^{(p)}$ of the k^{th} period. The discret state-space model of the movement system associated with the experience p is defined by (1)

$$X_{k+1}^{(p)} = \begin{bmatrix} f_{1_{k+1}}^{(p)} \\ \vdots \\ f_{j_{k+1}}^{(p)} \\ \vdots \\ f_{n_{k+1}}^{(p)} \end{bmatrix} = A^{(p)} X_k^{(p)} = A^{(p)} \begin{bmatrix} f_{1_k}^{(p)} \\ \vdots \\ f_{j_k}^{(p)} \\ \vdots \\ f_{n_k}^{(p)} \end{bmatrix}. \quad (1)$$

where $A \in \mathbb{R}^{n\times n}$ is the state-space matrix of the discrete autonomous system $S^{(p)}$. Of course, this system is highly nonlinear and it is impossible to model its dynamics over all of the $\bar{k}$ periods by a single state matrix A. So, one solution consists in proceeding by a "packet modeling". The idea is to develop a static model for a reduced number $\gamma < \bar{k}$ of periods belonging to a time window. This time window is called a sliding window (sw) insofar as it is offset by a period T in order to generate the static state matrix A_k, modeling the dynamics of the window sw_{k+1} as a function of the window sw_k. This modeling process is illustrated in the figure 5.

Figure 5 considers a number $\gamma = 5$ of periods for the sliding window. γ results from a compromise in the precision of the model obtained from the n signals and the computation time necessary to generate it. Thus, $x_1^{(p)}$ corresponds to the concanenation of the measurements of the signals $f_j^{(p)} \forall\, j \in \{1,..,n\}$ during the period sw_1, $x_2^{(p)}$ during the period sw_2, $x_k^{(p)}$ during the period sw_k,... Finally, it comes:

$$x_{k+1}^{(p)} = A_k^{(p)} x_k^{(p)} \quad \forall k \in \{1,..,\bar{k}-\gamma+1\} \quad (2)$$

For each period k, a linear system of the form (3) must be solved.

$$M_k\, y_k = x_{k+1}^{(p)} \quad (3)$$

with

$y_k \in \mathbb{R}^{n^1\times 1}$ is the unknown vector containing all the elements of $A_k^{(p)}$ whereas M_k is expressed as (4)

$$M_k^T = \mathbb{I}_n \otimes x_k^{(p)} \quad (4)$$

where $\mathbb{I}_n$ is the identity matrix of order n and $\otimes$ the product of Kronecker. It means that M_k is a hollow matrix and the solution vector y_k is computed from the expression (5)

$$y_k = y_0 + M_k^{\perp} \times \omega \quad (5)$$

where y_0 is a particular solution of system (3), $M_k^{\perp}$ is the kernel of M_k and ω is a vector of appropriate dimension whose values are generated randomly.

Under these conditions, each matrix $A_k^{(p)}$ can be considered as a time signature of the corresponding movement. In conclusion, the learning phase consists in generating $\bar{k}-\gamma+1$ matrices $A_k^{(p)}$, and this, for each experiment p associated with a movement, $\forall p \in \{1,..,N\}$, .

B. A polytopic representation for detection

Polytopic modeling is widely used to model complex systems [6]. It is obtained in a fairly conventional way by linearizing the nonlinear system around a number of more or less important equilibrium points so as to provide a satisfactory approximation of the nonlinear behavior of the system. In the case of motion capture systems considering time signals (as is the case in this work), this type of modeling trends to to be very interesting.

Now, we wish to detect, in real time if possible even if we plan to do post-treatment, the movement carried out by an individual λ equipped with the same combination mentioned in section *II-B*. Consider that the learning phase presented in the previous section consist of N experiences. The goal is to express, for each acquisition period k, the dynamics

of the n signals $f_j^{(\lambda)}$ as a function of the models associated with state-matrices $A_k^{(p)}$, generated during the learning phase. More precisely, it is question to generate matrices $\mathbb{A}_k^{(\lambda)}$ such as:

$$x_{k+1}^{(\lambda)} = \mathbb{A}_k^{(\lambda)} x_k^{(\lambda)} \quad \forall k \in \{1,..,\bar{k}-\gamma+1\} \tag{6}$$

where $\mathbb{A}_k^{(\lambda)} = A_k^{(\lambda)}(\theta^{(\lambda)})$ is a non linear matrix that represents a LPV model considering the polytope $\mathbf{A}_k^{(\lambda)}$ defined by

$$\mathbf{A}_k^{(\lambda)} = \left\{ A_k^{(\lambda)}(\theta^{(\lambda)}) = \sum_{p=1}^{N} \theta_p^{(\lambda)} A_k^{(p)} ; \theta^{(\lambda)} \in \Theta^{(\lambda)} \right\} \tag{7}$$

and where $\Theta^{(\lambda)}$ is the set of the barycentric coordinates:

$$\Theta^{(\lambda)} = \left\{ \theta^{(\lambda)} = \begin{bmatrix} \theta_1^{(\lambda)} \\ . \\ . \\ \theta_{N^{(\lambda)}} \end{bmatrix} \in \{\mathbb{R}^+\}^N \mid \sum_{p=1}^{N} \theta_p^{(\lambda)} = 1 \right\} \tag{8}$$

It is important to note that the state representation allows to express the dynamics of each signal $f_j^{(\lambda)}$ as a function of all signals. In other words, signals are not modeled separately and the model of the global system results in a single matrix, denoted by $\mathbb{A}_k^{(\lambda)}$.

According to the complexity of the system (number N of experiments, number γ of periods for the sliding window, ...), solving the system of equations (6) with (7) and (8) is not so trivial. This is actually an optimization problem defined by:

$$inf \ \{ f(\theta_p^{(\lambda)}) \mid \sum_{p=1}^{N} \theta_p^{(\lambda)} = 1 \} \tag{9}$$

with

$$f(\theta_p^{(\lambda)}) = \left(x_{k+1}^{(\lambda)} - \mathbb{A}_k^{(\lambda)} x_k^{(\lambda)} \right)^T \left(x_{k+1}^{(\lambda)} - \mathbb{A}_k^{(\lambda)} x_k^{(\lambda)} \right) \tag{10}$$

In this work, 'interior-point' algorithm will be used to solve the problem (9).

Finally, in the hypothesis of a satisfactory solution $\theta^{(\lambda)^\star}$ for the optimization problem (9), the values of $\theta_p^{(\lambda)^\star}$ can be considered as resemblance coefficients of the corresponding experiments of the learning phase. Each experience being associated with one of the movements m to be detected, it is then easy to compute the coefficient of resemblance linking the current experiment with the knowledge models.

Considering $\theta^{(\lambda_1)^\star}$ the N_1 coefficients of the vector $\theta^{(\lambda)^\star}$ associated with movement 1, $\theta^{(\lambda_2)^\star}$ those associated with movement 2, $\theta^{(\lambda_m)^\star}$ those associated with movement m... In this way, it is possible to compute $\phi_{mk}^{(\lambda)}$ the coefficient of resemblance of the current experiment with the movement m during the period k such as:

$$\phi_{mk}^{(\lambda)} = \sum_{i=1}^{N_m} \theta_i^{(\lambda_1)^\star} \tag{11}$$

IV. Experimental results

A. Florence 3D Action dataset

The diagnostic method presented in this paper was evaluated on the dataset collected at the University of Florence during 2012 and captured using a Kinect camera [9]. The recognition system relies on a skeletal based representation of the human body which can be compared to the connected jumpsuit considered in the *e-moove* project, illustrated on figure 2. The Kinect device's RGB-D data stream provides a wired skeleton at a speed of 30 frames per second, generating a skeleton joint acquisition frequency of 30 Hz. To reduce the effect of noise that may affect the coordinates of skeleton joints, a smoothing filter is applied to each sequence. The skeleton joints considered by the authors of [9] are the same 15 as those considered in section *II-B*. In this work, we just used the Cartesian coordinates of the 15 joints in Cartesian coordinates and, consequently, 45 signals f_j (3 signals per joint). In this work, the authors' contribution does not come at all from the hardware part used for data collection. This hardware part generates recurring problems such as measurement noise, sensor sensitivity (...) which are solely the work [9] and not at all presented in this article. The Florence 3D Action Dataset includes 9 activities: 1-wave, 2-drink from a bottle, 3-answer phone, 4-clap, 5-tight lace, 6-sit down, 7-stand up, 8-read watch, 9-bow. During acquisition, 10 subjects were asked to perform these actions. This resulted in a total of 215 activity samples. The objective is of course to propose an algorithm for the recognition of actions included in the dataset. In [9], a Naive-Bayes Nearest-Neighbor (NBNN) classifier is applied for action recognition. More precisely, 4 variants of this algorithm are presented (NBNN, NBNN+parts, NBNN+time and NBNN+parts+time) and then compared to other approaches in the literature.

B. Data pre-processing phase

The acquisition period of the activities carried out as part of this experience is not identical. Consequently, the number of measurement points obtained during this acquisition is also different from one activity to another. However, the learning phase of the proposed approach consists of a discret-time state-space modeling phase, obtained from data vectors of the same dimension. Consequently, a linear interpolation is carried out for each signal $f_j^{(p)}$, $\forall \{n,p\} \in \{(1,..,27) \times (1,..,N)\}$. This interpolation does not change the dynamics of the signal since it is performed as a function of the corresponding acquisition time vector. It finally follows that the dimension of each vector is such that $f_j^{(p)} \in \mathbb{R}^{1\times 34}$, which corresponds to the largest size of the function $f_j^{(p)}$ contained

in the database.
Then, it is important to indicate that some of the actions considered for this test can either be performed by left or right limbs (arms or legs). For example, the "wave action" can either be associated with a raising of the left arm or with a raising of the right arm.

Algorithm 1 Data pre-processing algorithm

Input: Original signals $g^{(p)} = \left[g_k^{(p)}\right]$ for $k \in \{1,..,45\}$

Output: Processed signals $f^{(p)} = \left[f_j^{(p)}\right]$ for $j \in \{1,..,27\}$

Let $j = 1$.
while $j \leq 27$ **do**
  if $i \leq 9$ **then**
    $f_j^{(p)} = g_j^{(p)}$.
  else {**if** $j \leq 18$}
    $f_j^{(p)} = sup(g_j^{(p)} + g_{j+9}^{(p)})$
  else
    $f_j^{(p)} = sup(g_{j+9}^{(p)} + g_{j+18}^{(p)})$
  end if
  $j = j+1$.
end while
return $f^{(p)}$

This amounts to considering the 9 sensors connected jumpsuit illustrated in the figure.

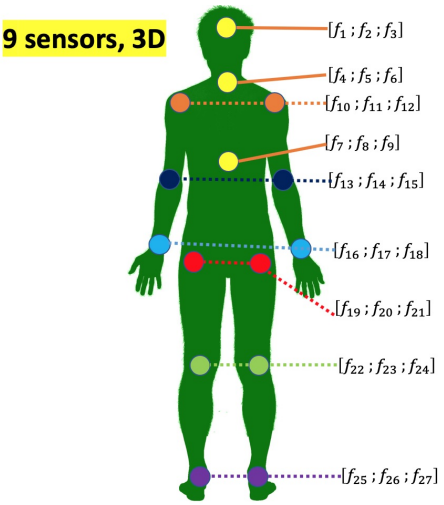

Fig. 6. 9 sensors connected jumpsuit

C. Learning stage

The discret-time state-space modeling method presented in section III-A is now applied for the learning phase. The size of the considered sliding window is $\lambda = 5$ periods. This learning stage consist in generating 29 state-matrices $A_k^{(p)} \in \mathbb{R}^{27\times 27}$ per activity p, $p \in \{1,..,215\}$.

Note that the resolution of the system (3) does not cause any difficulty since it is under-determined. The computations was performed with MATLAB R2017a on a Mac OS X system, processor 2.8 Ghz Intel Core i5. The computation time required to generate the 29×215 matrices is $17836\,s$, or almost 5 hours.
In the following, some of the 215 activities are considered. For these 12 activities, a signal f_j (solid line) as well as the response of the corresponding obtained model (tilled point) are plotted.

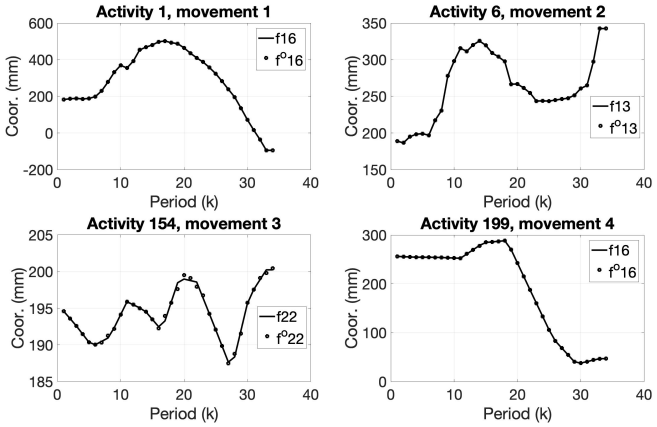

Fig. 7. State-space model validation - I

We can see from Figures 7, 8 and 9 that all signals are near estimated values. These simulation results show the effectiveness of the control law presented in this paper.

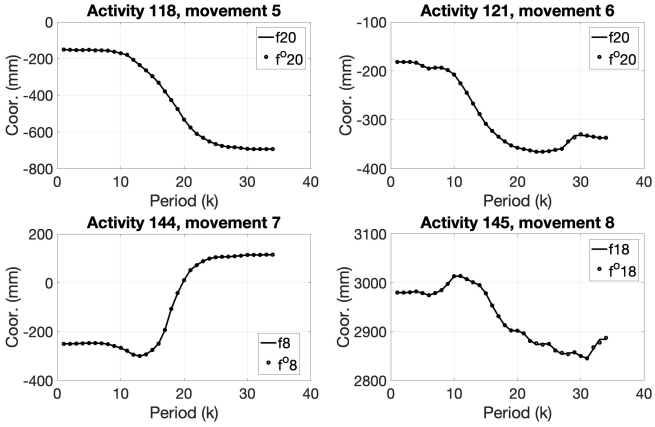

Fig. 8. State-space model validation - II

This comparison shows the efficiency of the modeling method and validates all the state-space models generated during the learning phase.

D. Detection stage

The authors of [9] suggest a leave-one-actor-out protocol: train your classifier using all the sequences from 9 out of 10 actors and test on the remaining 1. This validation case is denoted **out-1** in the sense that actor 1 is out of the protocol for the classifier. Repeat this procedure for all actors (**out-2** to **out-10**) and average the 10 classification accuracy values.

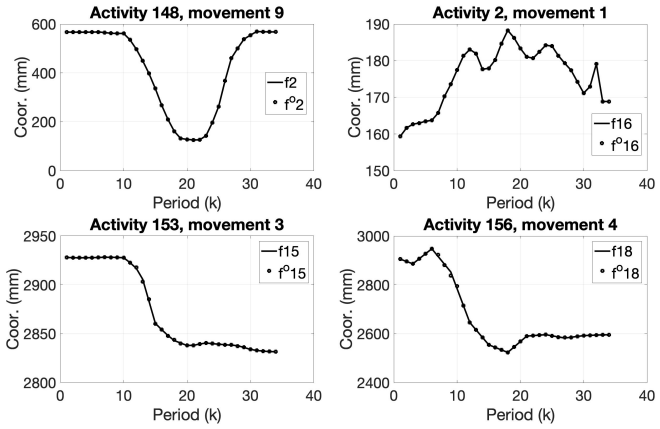

Fig. 9. State-space model validation - III

In this work, the same protocol is considered. In a general way, the validation test considering activities carried out by actor α and the other activities for the learning part, is denoted T_α. According to this, the following procedure is considered for this test step and repeated for all T_α, $\alpha \in \{1,..,10\}$.

Step 1 : Remove from the knowledge models bank resulting from the learning phase, the state-space matrices modeling the dynamics of the $\bar{N} = 215 - N$ activities of the actor λ. Then, for each activity λ, $\forall \lambda \in \{1,..,\bar{N}\}$, specific to the actor λ, initialize the variable τ to zero and repeat **Step 2** to **Step 5**.

Step 2 : For each period k, $\forall k \in \{1,..,34\}$, repeat **Step 2-a** and **Step 2-b** :

Step 2-a : Solve the optimization problem defined by (9) and (10) and generate the corresponding 34 vectors $\theta^{(\lambda)}$ defined by (8).

Step 2-b : Compute $\phi_{mk}^{(\lambda)}$ defined in (11).

Step 3 : Generate matrix $\phi^{(\lambda)} \in \mathbb{R}^{9\times 34}$ resulting from the concatenation of the coefficients $\phi_{mk}^{(\lambda)}$ and compute $\Upsilon^{(\lambda)} \in \mathbb{R}^{9\times 1}$, the column vector corresponding to the mean values of the 9 rows of the $\phi^{(\lambda)}$.

Step 4 : Extract the largest value $\Upsilon_{m^\star}^{(\lambda)}$ from $\Upsilon^{(\lambda)}$ as well as the number $m^\star$ of the associated line. $m^\star$ corresponds to the movement detected by the algorithm whereas $\Upsilon_{m^\star}^{(\lambda)}$ is the global pronosis.

Step 5 : Consider m the movement actually performed by the actor, if $m^\star = m$, then $\tau = \tau + 1$.

Step 6 : For each T_α, the score associated with the effectiveness of our approach is computed such as:

$$\rho_\alpha = \frac{\tau}{\bar{N}} \tag{12}$$

and finally, the final score is defined as in (13)

$$\rho = \frac{1}{10}\sum_{\alpha=1}^{10} \rho_\alpha \tag{13}$$

In order to illustrate this process, the case **out-3** is now considered. The activities associated with this actor are those numbered from 48 to 68. This means that the learning phase is carried out from activities 1 to 47 and 69 to 215, and the test is carried out on the $\bar{N} = 21$ activities 48 to 68.
First, consider the specific case of activity 58 during which actor 3 performed movement 5. The lines 4, 5 and 6 of $\phi_{mk}^{(58)}$ corresponding to the 3 highest similarity coefficients for this test (for clarity, the 6 others have been removed from the graph) are plotted in Fig. 10 for the 34 periods. In this case, our algorithm is rather very efficient since it detects, for most periods, the right movement with a high similarity coefficient. In the end, only the average values of the 9 lines of $\phi_{mk}^{(58)}$ (one per movement) are kept and then compared. The different values are presented in Tab. I and we can note that the average value selected by the algorithm (the largest) is largely associated with the movement performed.

TABLE I
RECOGNITION ACCURACY CASE **OUT-3**, ACTIVITY 58

m (mvt)	1	2	3	4	**5**	6	7	8	9
ϕ_{mk} (%)	0.9	1.6	1.5	4.8	**68.4**	14.3	4.1	0.9	3.5

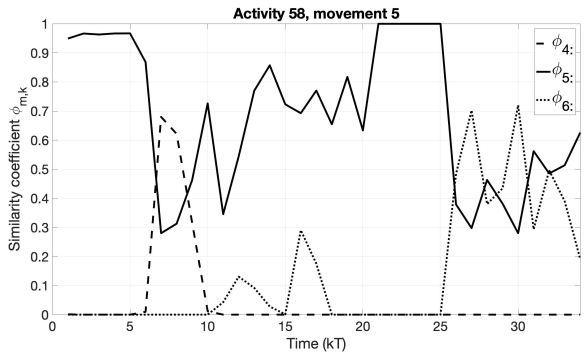

Fig. 10. Right dedection: case **out-3**, activity 58

Now, consider the specific case of activity 67 during which actor 3 performed movement 9. The same type of plot is proposed in Fig. 11 with the 2, 3 and 9 of $\phi_{mk}^{(58)}$, with the corresponding Tab. II. In this case, the coefficient remained by the algorithm is $\bar{\phi} = \phi_{3k}^{(58)} = 30.1$ which is associated with movement 3 whereas the movement carried out by the actor is 9. There is therefore a confusion of the algorithm between these two movements which are however very different .

Also, this "false detection" could be due to the resolution of the optimization problem.

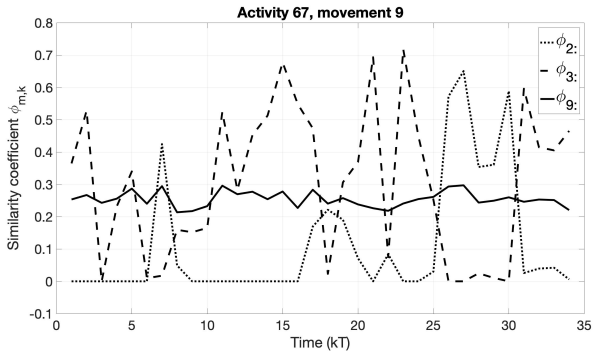

Fig. 11. False detection: case **out-3**, activity 67

TABLE II
RECOGNITION ACCURACY CASE **OUT-3**, ACTIVITY 67

m (mvt)	1	2	3	4	5	6	7	8	**9**
ϕ_{mk} (%)	7.1	11.4	**30.1**	12.8	10.3	0	0.2	0.3	27.8

Tab. III summarizes all the results obtained in case out-3 where 21 were considered. Only one "false detection" appears in this table (activity 67), which leads to an overall score of 20 "right detections" out of 21 possible, *i.e.* $\rho = 95.24\%$. Concerning the bad identification obtained for activity 67, there is not really precise reason which explains the confusion between movements 3 and 9 since they are very different. That said, the scores obtained for movement 9 (activities 66 to 68) are relatively low (less than 30%). For these activities, it means that several movements can obtain a similar score, the movement selected by the algorithm being the one corresponding to the highest score, even it is is barely higher than the second.

Table 4 shows the overall scores obtained in the 10 cases: **out**-1 to **out**-10. The global accurary involved by our approach is 97.25%.

Many techniques for action recognition have been proposed recently. They are effective and regularly compared to each other in the literature [12], [13]. These comparisons are possible thanks to several benchmark datasets, such as MSR-Action3D dataset [14], UTKinect dataset [13] or Florence dataset [9], for example. In this work, only the Florence dataset is used and the proposed approach is compared to four other approaches in the literature presented in [17]. This comparaison is reported in Table V.

E. Results analysis

The results presented in this section highlight the qualities of our algorithm for the detection of movements from the database of the University of Florence. However, these results should be put into perspective by listing certain aspects that could possibly put this algorithm at fault:

TABLE III
SUMMARY OF RESULTS - CASE **OUT-3**

Activity	actual/detected movements	$\bar{\phi}$
48	1 / 1	51.6661
49	1 / 1	45.8590
50	1 / 1	46.6734
51	2 / 2	28.9389
52	2 / 2	47.4374
53	3 / 3	46.0748
54	3 / 3	49.6033
55	4 / 4	30.2201
56	4 / 4	39.8524
57	5 / 5	31.0113
58	5 / 5	68.4080
59	6 / 6	31.8841
60	6 / 6	32.0132
61	7 / 7	28.1289
62	7 / 7	33.8440
63	8 / 8	29.0401
64	8 / 8	27.8872
65	8 / 8	29.1144
66	9 / 9	28.4524
67	**9 / 3**	**27.8292**
68	9 / 9	28.2153

TABLE IV
RECOGNITION ACCURACY COMPARISON

case **out-**	1	2	3	4	5	6	7	8	9	10
ρ	1	1	0.95	1	1	0.77	1	1	1	1

1) **sequence of movements**: The scenarios of this test are very specific, in the sense that only one movement per scenario is considered. In the case of scenarios considering several movements in the same activity, it would also be advisable to dissociate the different movements before applying the proposed algorithm.
2) **variety of population**: the movements of this study were carried out by actors of different sex but all relatively young. Also, our technique could show certain limitations in the case of movements made by very young subjects (children) or the elderly.
3) **measurement disturbances**: As mentioned above, several hardware points such as measurement noise, sensor sensitivity, camera accuracy (...) can alter the data. Developing robust state models could potentially solve this problem.

V. CONCLUSIONS

This work is part of a research project whose objective is to develop motion detection algorithms. Even if the **e-moove** project is more particularly dedicated to pathologies related to Parkinson's disease, the algorithm proposed in this paper is general and can be applied to any type of movement: sport, medical education, orthopedics... The proposed approach seems to be original for this type of detection problem. It considers a discret-time state modeling valid over a short "time window", but sliding over the whole experiment period : sliding window principle. The movement to be detected

TABLE V

RECOGNITION ACCURACY COMPARISON

Method	Accuracy (%)
Seidenari et al.2013[9]	82.15
Vemulapalli et al.2014 [15]	90.88
Devanne et al.2015 [16]	87.04
Wang et al.2015 [17]	94.25
Our approach	**97.25**

is modeled by a convex formulation of the state models obtained from the dataset, leading to an similarity index of the actual movement with the learning models.

A numerical illustration, considering the Florence 3D dataset, offers a comparison of the present approach with other techniques from the literature. This comparaison shows the effectiveness of this approach, which improves the action recognition accuracy.

REFERENCES

[1] N. Neverova, Deep Learning for Human Motion Analysis, Artificial Intelligence [cs.AI], Université de Lyon, 2016.

[2] M. H. Li, T. A. Mestre and S. H. Fox and B. Taati, Vision-Based Assessment of Parkinsonism and Levodopa-Induced Dyskinesia with Deep Learning Pose Estimation, Journal of NeuroEngineering and Rehabilitation, 2018.

[3] S. Berlemont, G. Lefebvre, S. Duffner and C. Garcia, Siamese Neural Network based Similarity Metric for Inertial Gesture Classification and Rejection, International Conference on Automatic Face and Gesture Recognition, May 2015.

[4] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi and A. del Bimbo, Reconnaissance d'actions humaines 3D par l'analyse de forme des trajectoire de mouvement, Compression et Représentation des Signaux Audiovisuels (CORESA), Reims, France, Nov 2014.

[5] N. Keijsers, M. Horstink and S. Gielen, Automatic Assessment of Levodopa-Induced Dyskinesia in Daily Life by Neural Networks, Movement Disorder, pp. 70-80, 2003.

[6] J. Bernussou, J. C. Geromel, and P. L. D. Peres, A linear programming oriented procedure for quadratic stabilization of uncertain systems, Systems and Control Letters, vol(13), pp 65-72, 1989.

[7] C. Szegedy, W. Liu, Y. Ji , P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, Going Deeper with Convolution, 2014.

[8] O. Kopuklu, A. Gunduz, N. Kose and G. Rigoll, Real-time Hand Gesture Detection and Classification Using Convolutional Neural Networks, 2019.

[9] L. Seidenari, V. Varano, S. Berretti, A. Del Bimbo and P. Pala, Recognizing Actions from Depth Cameras as Weakly Aligned Multi-PartBag-of-Poses, IEEE Conference on Computer Vision and Pattern Recognition Workshops, Portland, Oregon, 2013.

[10] W. Bian, D. Tao, and Y. Rui, Cross-domain human action recognition, In CVPR, IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 42, no. 2, pp. 298âĂŞ307, Apr. 2012.

[11] L. Liu, L. Shao, X. Zhen, and X. Li, Learning discriminative key poses for action recognition, IEEE Trans. on Cybernetics, vol. 43, no. 6, pp. 1860âĂŞ1870, Dec 2013.

[12] Q. D. Tran, and N. Ly, Sparse spatio-temporal representation of joint shape-motion cues for human action recognition in depth sequences, In IEEE Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), pp. 253âĂŞ258, 2013.

[13] L. Xia, C.-C. Chen, and J. Aggarwal, View invariant human action recognition using histograms of 3d joints, in IEEE CVPR, pp. 20-27, 2012.

[14] W. Li, Z. Zhang, and Z. Liu, Action recognition based on a bag of 3d points, in IEEE Conference on CVPRW, pp. 9-14, 2010.
Li, W.; Zhang, Z.; and Liu, Z. 2010. Action recognition based on a bag of 3d points. In (CVPRW), 2010 IEEE Conference on, 9âĂŞ14. IEEE.

[15] R.Vemulapalli, F. Arrate and R. Chellappa, Human action recognition by representing 3d skeletons as points in a lie group, In CVPR, IEEE Conference on, 588-595. IEEE, 2014.

[16] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi and A. Del Bimbo, 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold, Cybernetics, IEEE Transactions on 45(7):1340-1352, 2015.

[17] C. Wang, J. Flynn, Y. Wang, and A. L. Yuille, Recognizing Actions in 3D Using Action-Snippets and Activated Simplices, Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona USA, 2016.