Human Motion Modelling and Recognition: a Computational Approach

Barbara Bruno, Fulvio Mastrogiovanni, Antonio Sgorbissa, Tullio Vernazza and Renato Zaccaria

Abstract— **The design of computational methods to recognize human motions is among the most promising research activities in Ambient Intelligence. Accepted solutions use acceleration data provided by wearable sensors. To design general procedures for motion modeling and recognition, this article adopts Gaussian Mixture Modeling and Regression to build computational models of human motion learned from human examples that allow for an easy run-time classification. The main contributions are: (i) an optimized selection of the proper number of Gaussians for building motion models, which is usually assumed to be *a priori* known; (ii) a comparison between models built by keeping the acceleration axes independent (i.e., $6{\times}2D$ *approach*) and models taking axes correlation into account (i.e., referred to as $2{\times}4D$ approach).**

I. Introduction

Among the major research trends in Ambient Intelligence [1], motion modeling is fundamental for human behavior monitoring [2], [3], [4], [5], [6], [7], [8], [9], both for smart environments [10] and surveillance applications [11].

The degree of personal autonomy can be *measured* by checking the ability to perform the so-called "activities of daily living" [12]. These include personal hygiene (e.g., teeth brushing, bathing), feeding (e.g., preparing the meal, eating, drinking) or everyday tasks (e.g., walking, getting up). Smart homes are fundamental to detect the occurrence of (and the *lack of*) activities of daily living (ADL) while the assisted person is performing them [13]. To this aim, a number of sensors distributed in the environment can adequately recognize human behavior, possibly by composing simpler activities into complex action patterns [14], [15]. A number of smart homes have been presented [13], which verify the possibility of detecting the occurrence of ADL. The problem of modeling and detecting human motions (e.g., drinking, teeth brushing) with sensors distributed through the environment is still an open issue: each person performs the modeled action in a distinctive way.

To overcome these limitations, wearable sensors are an interesting approach. Computational models of human motion require to encode general information about the limb trajectories in space. Wearable accelerometers are a valuable source of such data [5], [16]. Computational models encoding motions and allowing for an easy automated motion recognition are of the utmost importance. This requires to: (i) build models adaptable to individual body structures; (ii) represent different models in an efficient and compact way; (iii) classify run-time acceleration data using available motion models. To this aim, we propose the use of Gaussian Mixture Modeling and Regression (GMM and GMR) to encode human motion models learned from examples, which can be used for run-time classification. It is worth noting that the proposed approach is not focused on bio-mechanical or kinematic modeling [17]. The contribution of this article is: (i) an optimized selection of the number of Gaussians for building models, which is usually assumed to be *a priori* known; (ii) a comparison between models built by keeping the acceleration axes independent (i.e., the $6{\times}2D$ *approach*) and models taking axes correlations into account (i.e., the $2{\times}4D$ *approach*): the second approach offers increased accuracy and shorter modeling times. The article is organized as follows. Section II introduces the main requirements. Section III describes the computational models. Experiments are discussed in Section IV. Conclusion follows.

TABLE I
ADLs AND RELATED MOTION PRIMITIVES

***ADL* activity**	**Motion primitive(s)**
Personal hygiene	Brush teeth Comb hair
Mobility	Climb stairs Descend stairs Walk
Feeding	Drink from a glass Pour water into a glass Eat with knife and fork Eat with spoon
Communication	Use the telephone
Functional transfers	Get up from the bed Lie down on the bed Stand up from a chair Sit down on a chair

II. System Requirements

Human Motion Primitives. We focus on movements related to the "Activities of Daily Living" [12], which are used to measure a person autonomy. However, ADL are too generic to be automatically modeled and recognized. We break relevant ADL down into low-level activities, which we call "motion primitives", i.e., motions that uniquely identify an activity and are associated with stereotyped movements (see Table I).

Instrumentation. Tri-axial accelerometers have been used to acquire data in human motion recognition applications [2], [18], [6], [8], [9]. The location of the accelerometer on the human body is related to the activity of interest: a waist placement is adopted for detecting posture [18], [8], but it is not suited to detect hand gestures, for which a wrist placement is preferable [3]. The nature of the motions we

B. Bruno, F. Mastrogiovanni, A. Sgorbissa, T. Vernazza and R. Zaccaria are with the University of Genova, Via Opera Pia 13, 16145 Genova, Italy. Corresponding author's email: barbara.bruno@unige.it.

consider lead us to use a right wrist location.

Feature Modeling. It is necessary to define the set of features to model. On the one hand, the work reported in [18], [8] describes modeling algorithms based on gravity and body acceleration along the x, y and z axes, as well as the spectrum of each axis. On the other hand, the technique described in [4] relies on time-domain features only for the on-line motion recognition. We decided to consider only time-domain features, respectively gravity $g_x(t)$, $g_y(t)$, $g_z(t)$ and body accelerations $b_x(t)$, $b_y(t)$, $b_z(t)$.

Requirements on the Modeling Method. Different methods have been proposed to model human motions. However, it is extremely complex to compare them according to the quality of the resulting models. In order to evaluate benefits and drawbacks, we adopted as criteria the size of the final models and the overall time and resources required by the recognition procedure. Various methods have been presented to obtain motion models of relatively small size (e.g., MLP neural networks in [9], Gaussian Mixture Models in [18], Hidden Markov Models in [3]). However, few solutions optimize other parameters (e.g., the approach in [7] compares run-time data with the initial dataset, thereby completely disregarding the modeling phase). No method seems to combine the benefits of the two kind of approaches. Inspired by the work described in [19], we decided to adopt GMM and GMR to build "expected curves" from human examples of motions to store the information in a compact way and, by creating models in the same space of the raw data, to make the comparison simpler and fast.

III. Human Motion Primitives Modeling

A. Overview of GMM and GMR

Once S representative trials for each motion primitive are collected, the distinctive traits of each feature $f \in \{g_x(t), g_y(t), g_z(t), b_x(t), b_y(t), b_z(t)\}$ are encoded therein. Each trial is made of N data points and is defined as $\Xi^{fs} = \{\xi_1, \ldots, \xi_N\}$. As shown in Figure 1, each data point i is a couple (ξ_i^t, ξ_i^a), respectively time and the corresponding measured acceleration.

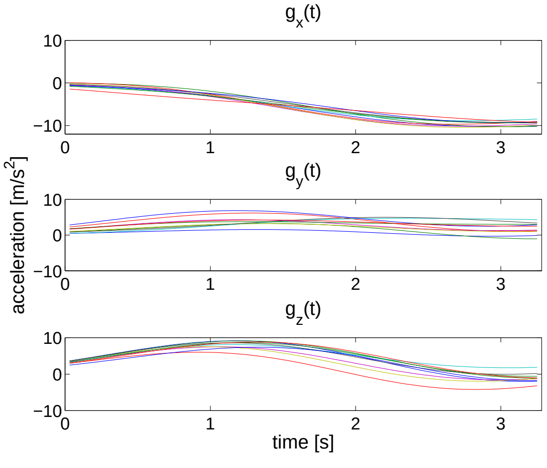

Fig. 1. Gravity features for 10 trials used to model "stand up from a chair".

Given the sets Ξ^{fs}, GMM and GMR encapsulate the information in a unique set $\hat{\Xi}^f = \{\hat{\xi}_i, \hat{\Sigma}_i^{aa}\}$ with $i = 1, \ldots, N$ where $\hat{\xi}_i = \{\xi_i^t, \hat{\xi}_i^a\}$ represents the expected curve, $\hat{\xi}_i^a$ is the expected acceleration retrieved for the time instant ξ_i^t and $\hat{\Sigma}_i^{aa}$ is the associated covariance matrix computed over the S trials. For each feature f of each activity, GMM and GMR group the data points belonging to different trials by clustering them using K Gaussian distributions and maximizing, via the EM algorithm, the probability for the data points to be generated by the Gaussian mixture. Specifically, (i) a unique set Ξ^f is built as the union of the S sets Ξ^{fs} and K-means is used to cluster data points in Ξ^f; (ii) data points in Ξ^f are clustered with GMM and the modeling parameters are trained with EM; (iii) GMR is applied to retrieve $\hat{\Xi}^f$.

B. Selection of the Number of Gaussians

The GMM principle is to cluster the data so that, given the number K of clusters, the resulting groups enhance the distinctive traits of considered motions. The K parameter can vary with the motion and the feature and its precise tuning is a crucial step. Approaches in the Literature neglect this operation [18]. A good value for K is a compromise between two opposite goals: on the one hand, it must represent all the representative traits of the dataset with high accuracy, which suggests high values for K; on the other hand, it must generalize the dataset, which bounds the maximum suitable value for K.

Given a feature dataset $\Xi^f = \{\xi_1, \ldots, \xi_{NS}\}$, estimating a good value for K is a process ranging from a simple "rule-of-thumb" that defines it as a function of the size NS of the dataset, up to the "Information Criterion" approach [20], which associates a likelihood function to the clustering model and then solves the optimization problem given by its constrained maximization.

Due to the peculiar nature of Ξ^f (where all the points ξ_i belong to the same cluster) we developed a simple algorithm for the selection of the optimal K^* based on "silhouettes" [21]. The silhouette of a data point is a measure of how similar that point is to the other points in the same cluster, with respect to the similarity to points in other neighbor clusters. Silhouette ranges from -1 (i.e., wrong clustering) to $+1$ (i.e., optimal clustering). The choice of K^* starts with an arbitrary small number of clusters and increases it until clusters are appropriate to represent the data.

At a given time instant, K clusters Ξ_k have been produced (i.e., $\Xi^f = \bigcup_k \Xi_k$). The following definitions are in order.

- $dist(\xi_i, \Xi)$ returns the set of the Euclidean distances between ξ_i and each point in Ξ. When $\xi_i \in \Xi_k$, $dist(\xi_i, \Xi^f \setminus \Xi_k)$ is the set of distances between ξ_i and all the points in Ξ^f not belonging to Ξ_k, and $dist(\xi_i, \Xi_k)$ is the set of distances between ξ_i and the other points in Ξ_k.
- $avg(X)$, $min(X)$ and $max(X)$ return the average, minimum and maximum value in X.

Then, the silhouette of a point $\xi_i \in \Xi_k$ is defined as:

$$s_i = \frac{min(dist(\xi_i, \Xi^f \setminus \Xi_k)) - avg(dist(\xi_i, \Xi_k))}{max(avg(dist(\xi_i, \Xi_k)), min(dist(\xi_i, \Xi^f \setminus \Xi_k)))}, \tag{1}$$

The “fitness” F_k of a cluster is defined as:

$$F_k = \frac{1}{N_k} \sum_{i\ s.t.\ \xi_i \in \Xi_k} s_i, \tag{2}$$

where N_k is the number of points in Ξ_k. The overall fitness F is defined as:

$$F = \frac{1}{K} \sum_k F_k. \tag{3}$$

The overall fitness is a non-increasing function, which can be used as a penalty to threshold the accuracy of the corresponding model.

```
Algorithm 1 Compute $K^*$
Require: The set $\Xi^f = \{\xi_1, \ldots, \xi_{NS}\}$ of data points
Ensure: A suitable number of clusters $K^*$
 1: $loop \leftarrow true$
 2: $K \leftarrow 2$
 3: while $loop = true$ do
 4:     $\{\Xi_1, \ldots, \Xi_K\} \leftarrow$ Kmeans($K$, $\Xi^f$)
 5:     for all $\xi_i \in \Xi^f$ do
 6:         $s_i \leftarrow$ CompSil($\xi_i, \{\Xi_1, \ldots, \Xi_k, \ldots, \Xi_K\}$)
 7:     end for
 8:     for all $k = 1, \ldots, K$ do
 9:         $F_k \leftarrow$ CompFit($\{s_1, \ldots, s_{NS}\}$)
10:     end for
11:     $F \leftarrow$ CompOvFit($\{F_1, \ldots, F_K\}$)
12:     if $F > 0.69$ then
13:         $K \leftarrow K + 1$
14:     else
15:         $K^* \leftarrow K$
16:         $loop \leftarrow false$
17:     end if
18: end while
19: $return(K^*)$
```

The procedure is outlined in Algorithm 1. The Algorithm loops until the first suitable value K^* is found. Functions `CompSil`, `CompFit` and `CompOvFit` implement the equations (1), (2) and (3). The fitness test value of 0.69 at line 12 has been experimentally determined as a good trade-off between precise and general models. Experiments show that K^* is usually between 5 and 20 (for the considered activities) with smaller values for the gravity components and higher values for the body accelerations.

C. Gaussian Mixture Modeling and Regression

The modeling procedure is made up of three steps, namely (i) model initialization with K-means clustering; (ii) EM based GMM training; (iii) GMR based retrieval of expected curves. Figure 2 shows the result of K-means clustering on the set of body accelerations measured along the z axis and recorded in 10 trials of “drink from a glass”.

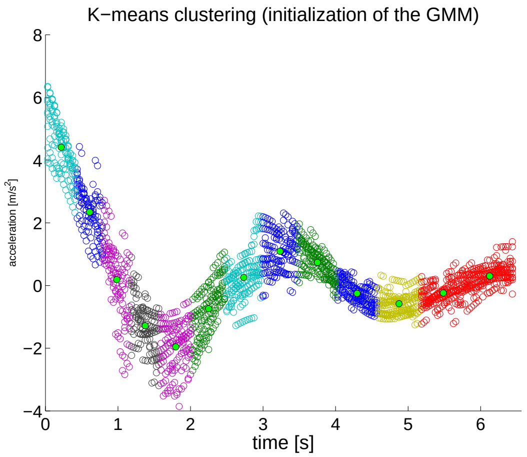

Fig. 2. Body accelerations measured along the z axis while drinking from a glass - 10 trials.

In GMM, the data set $\Xi^f = \{\xi_1, \ldots, \xi_{NS}\}$ of NS data points of size D is modeled by a mixture of K^* components and defined by the probability density function:

$$p(\xi_i) = \sum_{k=1}^{K^*} p(k) p(\xi_i | k), \tag{4}$$

where $p(k)$ is the prior and $p(\xi_i|k)$ is the conditional probability density function. In the case of a mixture of K^* Gaussian distributions of size D, they are:

$$\begin{aligned} p(k) &= \pi_k, \\ p(\xi_i|k) &= \mathcal{N}(\xi_i; \mu_k, \Sigma_k) \\ &= \frac{1}{\sqrt{(2\pi)^D |\Sigma_k|}} e^{-\frac{1}{2}((\xi_i - \mu_k)^T {\Sigma_k}^{-1} (\xi_i - \mu_k))}. \end{aligned} \tag{5}$$

Each GM model Θ_f (modeling feature f for a given activity) is associated with the K^* components, each one defined by the parameters $\{\pi_k, \mu_k, \Sigma_k, E_k\}$, which stand for the prior probability, the mean vector, the covariance matrix and the cumulated posterior probability $E_k = \sum_{i=1}^{NS} p(k|\xi_i)$, computed using the Bayes theorem.

The *average log-likelihood* of a model Θ_f to the data points $\{\xi_1, \ldots, \xi_{NS}\}$ of the modeling set is computed as:

$$\mathcal{L}_{\Theta_f} = \frac{1}{NS} \sum_{i=1}^{NS} \log(p(\xi_i)), \tag{6}$$

where $p(\xi_i)$ is the probability that ξ_i belongs to the model, computed using (4).

The GMM parameters are initialized by K-means (prior probabilities correspond to the assignments, means to centroids, covariances to standard deviations) and trained in *batch mode* by the EM algorithm. The algorithm updates the estimated values of the modeling parameters by iteratively executing two steps, the Expectation and the Maximization step respectively, until convergence is reached[1], i.e., when $\frac{\mathcal{L}^{t+1}}{\mathcal{L}^t} < C$, where $C = 10^{-10}$.

[1] The use of EM is out of scope in this article. Please refer to [22].

GMR is a method to retrieve an expected curve from the initial data, once modeled with GMM. This allows for a simple comparison with the incoming data at run-time [22]. A regression problem is defined by a set $X \in \mathbb{R}^p$ of predictor variables and a set $Y \in \mathbb{R}^q$ of response variables. The aim of the regression is to estimate the conditional expectation of Y given X on the basis of a set of observations in the form (X, Y). We use regression to retrieve the *expected curves* of the features representing each motion primitive, which are computed over a set of observed curves. Focusing on one specific feature (e.g., body acceleration along the z axis, under the $6{\times}2D$ approach), and dropping the f index for the sake of clarity, we assume it to be defined by the Ξ set of N data-points (ξ_i^t, ξ_i^a), where Ξ^t corresponds to the set of predictor variables and Ξ^a corresponds to the set of response variables. The features extracted from the trials of the modeling dataset provide the observations (Ξ^t, Ξ^a) and regression allows to estimate the conditional expectation of Ξ^a given Ξ^t. The joint probability distribution $p(\Xi^t, \Xi^a)$ of a set of data points in the form (ξ^t, ξ^a) is computed with GMM. Then, $E[p(\Xi^a|\Xi^t)]$ and $E[cov(p(\Xi^a|\Xi^t))]$ allow to estimate the curve and its constraints.

The temporal and acceleration values of the k component of a Θ_f GM model are separated as:

$$\mu_k = \{\mu_k^t, \mu_k^a\}, \qquad \Sigma_k = \begin{pmatrix} \Sigma_k^{tt} & \Sigma_k^{ta} \\ \Sigma_k^{at} & \Sigma_k^{aa} \end{pmatrix}. \tag{7}$$

The expected distribution of Ξ_k^a given Ξ^t is defined as:

$$\begin{cases} p(\Xi_k^a|\Xi^t, k) = \mathcal{N}(\Xi_k^a; \hat{\Xi}_k^a, \hat{\Sigma}_k^{aa}), \\ \hat{\Xi}_k^a = \mu_k^a + \Sigma_k^{at}(\Sigma_k^{tt})^{-1}(\Xi^t - \mu_k^t), \\ \hat{\Sigma}_k^{aa} = \Sigma_k^{aa} - \Sigma_k^{at}(\Sigma_k^{tt})^{-1}\Sigma_k^{ta}. \end{cases} \tag{8}$$

The parameters $\hat{\Xi}_k^a$ and $\hat{\Sigma}_k^{aa}$ are mixed according to the probability of the k component to be responsible for Ξ^t, which is defined as:

$$\begin{cases} p(\Xi^a|\Xi^t) = \sum_{k=1}^{K} \beta_k \mathcal{N}(\Xi^a; \hat{\Xi}_k^a, \hat{\Sigma}_k^{aa}), \\ \beta_k = \dfrac{p(k)p(\Xi^t|k)}{\sum_{j=1}^{K} p(j)p(\Xi^t|j)} = \dfrac{\pi_k \mathcal{N}(\Xi^t; \mu_k^t, \Sigma_k^{tt})}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\Xi^t; \mu_j^t, \Sigma_k^{tt})}. \end{cases} \tag{9}$$

Finally, the estimate of the conditional expectation of Ξ^a given Ξ^t is computed using:

$$\hat{\Xi}^a = \sum_{k=1}^{K} \beta_k \hat{\Xi}_k^a, \qquad \hat{\Sigma}^{aa} = \sum_{k=1}^{K} \beta_k^2 \hat{\Sigma}_k^{aa}. \tag{10}$$

By evaluating $\{\hat{\xi}_i^a, \hat{\Sigma}_i^{aa}\}$ at different time steps ξ_i^t, it is possible to represent the features with the expected form of the curves $\hat{\xi} = \{\xi^t, \hat{\xi}^a\}$ and associated covariance matrices. The two quantities $\{\hat{\xi}_i^a, \hat{\Sigma}_i^{aa}\}$ can be projected back in the original data space by using the linear transformation property of a Gaussian distribution:

$$\begin{aligned} \xi_i &\sim \mathcal{N}(\mu_k, \Sigma_k), \\ \to A\xi_i &\sim \mathcal{N}(A\mu_k, A\Sigma_k A^T). \end{aligned} \tag{11}$$

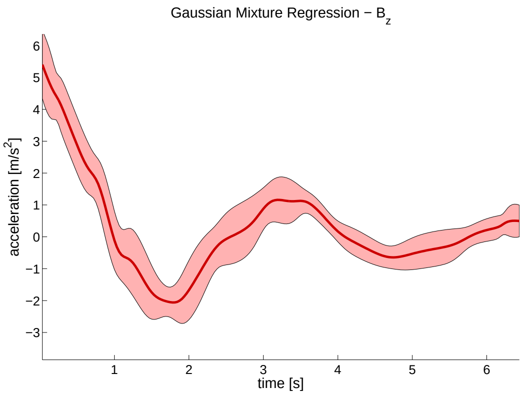

Fig. 3. Expected curve of the feature b_z of the motion “drink from a glass” retrieved via GMR.

Figure 3 shows the expected curve representing the body acceleration along the z axis while drinking from a glass. It is worth noting that by explicitly expressing the time index of the considered point we can reduce the number of points used to represent the feature (e.g., we could discard the points with a large associated standard deviation, since they are only very loosely representative of the motion) and thus obtain lighter models for each activity, with no loss in the classification accuracy.

D. Dimensionality of Data Points: 2D and 4D Approaches

The peculiar nature of acceleration values generated by human motions and measured by a tri-axial accelerometer allows us to consider two representation possibilities, namely to assume the acceleration axes to be independent from each other (i.e., $6{\times}2D$ *approach*) or not (i.e., $2{\times}4D$ *approach*). This leads to two possible problem characterizations: in the first case we have 6 different $2D$ features (i.e., $(t, g_x), (t, g_y), (t, g_z), (t, b_x), (t, b_y), (t, b_z)$), while in the latter case we have only 2 different $4D$ features (i.e., $(t, g), (t, b)$). The differences between the two approaches have never been thoroughly analyzed in the Literature [18]. Furthermore, the correlations among the three axes of the accelerometer have been considered only in [8], where the Pearson correlation coefficient is used.

In principle, the correlation is expected to provide additional knowledge to the model, thereby resulting in an increased classification accuracy: (i) Is it true that the $2{\times}4D$ approach leads to better performance? How big is the gain with respect to the $6{\times}2D$ approach? (ii) What are the drawbacks of the $2{\times}4D$ approach? How and how much do they affect the overall system? The experimental tests we have set up aim to evaluate the accuracy of the models built under the two different approaches as well as to estimate the memory space required to store the models and the time required to retrieve them.

IV. Experimental Results

Adopted instrumentation. All the data have been obtained using a *sensing bracelet* mounted on the user right wrist. The sensor is a tri-axial accelerometer able to sense up to

TABLE II

OVERVIEW OF THE PILOT MODELS

Model	# modelling trials	# volunteers	*N*
Drink_glass	16	5	206
Climb_stairs	7	1	206
Sitdown_chair	13	5	104
Standup_chair	16	4	91

$3G$ and to code the information using 6 bit per axis, with a sampling frequency $F_s = 32Hz$.

Feature extraction. We filtered the raw data with a median filter of size 3 to reduce the high frequency noise, then with a low-pass filter to discriminate between gravity and body acceleration. The low-pass filter is a Chebyshev I 5° order filter with $F_{cut} = 0.25Hz$, $A_{pass} = 0.001dB$, $A_{stop} = -100dB$, $F_{stop} = 2Hz$.

Models. We recorded 152 trials of 4 pilot activities ("drink from a glass", "stand up from a chair", "sit down on a chair" and "climb stairs") from 9 volunteers (5 men and 4 women with age ranging from 19 to 81) to be used both for the creation and the validation of the models (see Table II). *Drinking* and *climbing the stairs* are complex motions, whereas *sitting down* and *standing up* are postural transitions. Climbing the stairs is a reiterated activity, which we represent as climbing three steps. The accelerations associated with sitting down and standing up are very common patterns that can be found in the initial and final stages of a wide variety of activities, including drinking and climbing the stairs.

Space complexity. GMR defines the models $\hat{\Xi} = \{\hat{\Xi}_f\}$ as sets of expected curves (one for each feature), defined as a collection of expected points:

$$\hat{\Xi} = \bigcup_f \left(\bigcup_i \{\xi^t_{i,f}, \hat{\xi}^a_{i,f}, \hat{\Sigma}^{aa}_{i,f}\} \right), \tag{12}$$

where f cycles over the independent features and i cycles over the number N of points defining each expected curve, which can be virtually set to any value $N^* \in [1, \ldots, N]$ according to the required accuracy/size ratio. Recalling that, for a point i of feature f, $\xi^t_{i,f}$ is the time value, $\hat{\xi}^a_{i,f}$ is the expected acceleration value and $\hat{\Sigma}^{aa}_{i,f}$ is the estimated standard deviation matrix, then the $6\times 2D$ approach produces models of size $6\times(2\times N + N) = 18\times N$, since both each $\xi^a_{i,f}$ and each $\hat{\Sigma}^{aa}_{i,f}$ are scalar values, whereas the $2\times 4D$ approach produces models of size $2\times(4\times N + N\times 3\times 3) = 26\times N$, since each $\xi^a_{i,f}$ is composed of three values and each $\hat{\Sigma}^{aa}_{i,f}$ is a 3×3 matrix. As a consequence, models retrieved by applying GMM and GMR over $2D$ data result to be 30.77% smaller than the ones generated using the $2\times 4D$ approach.

Modeling time. Table III reports the average and maximum modeling times to create three different models using the two modeling approaches. The resulting data are computed using 10 trial executions[2].

TABLE III

AVERAGE AND MAXIMUM MODELING TIMES FOR THE $6\times 2D$ AND $2\times 4D$ APPROACHES

Model		T_{2D}	T_{4D}
climb_stairs	avg	142.477s	21.714s
	max	169.078s	28.422s
drink_glass	avg	360.217s	97.047s
	max	492.328s	148.953s
standup_chair	avg	131.595s	19.153s
	max	223.250s	24.016s

Expecting the modeling time to be dependent on the two parameters N and S, the models are chosen to be indicators of this dependency (see Table III): "climb_stairs" and "drink_glass" have the same number $N = 206$ of data points, but a different number of trials to create the model from ($S = 7$ for "climb_stairs", $S = 16$ for "drink_glass"), thereby allowing us to estimate the dependency of the modeling time with respect to S; "standup_chair" is generated starting from 16 modeling trials, but it is defined by a smaller number of data points ($N = 104$) with respect to "drink_glass", so the comparison between "standup_chair" and "drink_glass" provides a measure of the dependency of the modeling time with respect to N.

Parameters N and S have similar weights on modeling time, regardless of the approach: "climb_stairs" and "standup_chair", having $S_{climb} \approx \frac{1}{2} S_{drink}$ and $N_{standup} \approx \frac{1}{2} N_{drink}$ respectively, always have comparable modeling times. Both N and S affect modeling time: for the $6\times 2D$ approach, by halving either N or S we have a reduction of more than 60% (modeling time of "climb_stairs" is on average 39.55% of that of "drink_glass" and the one of "standup_chair" is on average 36.53%); for the $2\times 4D$ approach the reduction is around 80% (modeling time of "climb_stairs" is on average 22.37% of that of "drink_glass" and the one of "standup_chair" is on average 19.74%). Applying GMM and GMR with the $2\times 4D$ approach is sensibly faster than with the $6\times 2D$ approach, being the former modeling times constantly less than 30% of the equivalent latter modeling times (4D modeling times for "climb_stairs", "drink_glass" and "standup_chair" are, respectively, 15.24%, 26.94% and 14.55% of their 2D equivalent, on average).

Model validation. We adopt the Mahalanobis distance and a decision tree to validate the classification. The Mahalanobis distance $D_M(x)$ is used to determine the similarity of an actual acceleration trajectory to the models provided as expected curves and computed by GMR. The Mahalanobis distance of a multivariate vector $x = (x_1, \ldots, x_N)^T$ with mean $\mu = (\mu_1, \ldots, \mu_N)^T$ and covariance matrix S is defined as:

$$D_M(x) = \sqrt{(x-\mu)^T S^{-1} (x-\mu)}. \tag{13}$$

The comparison provides one Mahalanobis distance for each feature (6 and 2 distances for the $6\times 2D$ and $2\times 4D$ approaches): we define the overall distance as the mean of the feature distances. Decision trees are simple classification

[2] We used a standard HP® Compaq® NX6110 notebook, equipped with an Intel® Pentium® M Processor 740 ($2M$ cache, $1.73GHz$, $533MHz$ FSB) and $512MB$ of RAM.

TABLE IV

CLASSIFICATION ACCURACY FOR THE $6 \times 2D$ AND $2 \times 4D$ APPROACHES

Model	2D		4D	
	TP	TN	TP	TN
Climb_stairs	36.67%	94.29%	83.34%	87.14%
Drink_glass	96.67%	100%	93.34%	94.29%
Sitdown_chair	60%	97.50%	60%	98.75%
Standup_chair	70%	87.50%	80%	90%

algorithms usually adopted in real-time applications for the limited time and resources they require [4].

The developed classification algorithm first computes the overall distance of the actual data w.r.t. each model; then checks which models are closer than a given threshold distance; finally, it labels the actual data either as *unknown* if no model is sufficiently *close* or as an instance of the model with minimum distance otherwise. Thresholds vary with the models and are computed as the Mahalanobis distance of the *farthest curve* belonging to the model (i.e., the expected curve plus the standard deviation), multiplied by a scaling factor. The adopted values for the scaling factors have been experimentally set.

Table IV summarizes the results: as we expected, the motion primitives "climb stairs", "sit down on a chair" and "stand up from a chair" were harder to recognize due to their similarities, whereas the $2 \times 4D$ approach proved to be more effective by increasing both the true positive (TP) and the true negative (TN) rates. Moreover, the validation dataset was composed of trials of volunteers both known and unknown to the models, in order to further test the quality of the models. As the results of the "climb stairs" model prove, the $2 \times 4D$ approach allows to build larger models with fewer trials, thus increasing the probability of obtain general models. The motion primitive "drink_glass", modeled with trials of 5 volunteers and presenting acceleration patterns clearly different from those of the other known activities, gives a measure of the possibilities of the proposed modeling procedure when set up with clearly distinguishable activities.

V. CONCLUSIONS

We propose a method for modeling human motion primitives based on Gaussian Mixture Modeling and Gaussian Mixture Regression. We developed a simple algorithm, based on silhouettes, which allows us to identify suitable values for the number of Gaussians K. Finally, we found worth exploring the $2 \times 4D$ approach to modeling accelerometer data, which takes the correlation among the three axes of the accelerometer into account. Experiments prove that it is more accurate than the commonly adopted $6 \times 2D$ approach.

REFERENCES

[1] F. Mastrogiovanni and N. Chong, *Handbook of Research on Ambient Intelligence and Smart Environments*. Hershey, Pennsylvania, USA: IGI Global, May 2011.

[2] N. Ravi, N. Dandekar, P. Mysore, and M. Littman, "Activity recognition from accelerometer data," in *Proceedings of the the 2005 Conference on Innovative Applications of Artificial Intelligence (IAAI'05)*, Pittsburgh, USA, July 2005.

[3] D. Minnen and T. Starner, "Recognizing and discovering human actions from on-body sensor data," in *Proceedings of the 2005 IEEE International Conference on Multimedia and Expo (ICME'05)*, Amsterdam, The Netherlands, July 2005.

[4] D. Karantonis, M. Narayanan, M. Mathie, N. Lovell, and B. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 156–167, 2006.

[5] T. Choudhury, S. Consolvo, B. Harrison, J. Hightower, A. LaMarca, L. LeGrand, A. Rahimi, A. Rea, G. Borriello, B. Hemingway, P. Klasnja, K. Koscher, J. Landay, J. Lester, D. Wyatt, and D. Haehnel, "The mobile sensing platform: an embedded activity recognition system," *IEEE Pervasive Computing Magazine*, vol. 7, no. 2, pp. 32–41, 2008.

[6] X. Long, B. Yin, and R. Aarts, "Single-accelerometer-based daily physical activity classification," in *Proceedings of the 2009 IEEE International Conference on Engineering in Medicine & Biology Society (EMBS'09)*, Minneapolis, USA, September 2009.

[7] T. Brezmes, J. Gorricho, and J. Cotrina, "Activity recognition from accelerometer data on a mobile phone," in *Proceedings of the 2009 International Work-Conference on Artificial Neural Networks (IWANN'09)*, Salamanca, Spain, June 2009.

[8] G. Krassnig, D. Tantinger, C. Hofmann, T. Wittenberg, and M. Struck, "User-friendly system for recognition of activities with an accelerometer," in *Proceedings of the 2010 International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth'10)*, Munich, Germany, May 2010.

[9] M. Lee, A. Khan, and T. Kim, "A single tri-axial accelerometer-based real-time personal life log system capable of human activity recognition and exercise information generation," *Personal and Ubiquitous Computing*, vol. 15, no. 8, pp. 887–898, 2011.

[10] F. Mastrogiovanni, A. Sgorbissa, and R. Zaccaria, "A distributed architecture for symbolic data fusion," in *Proceedings of the 2007 International Joint Conference on Artificial Intelligence (IJCAI'07)*, Hyderabad, India, January 2007.

[11] M. Castelnovi, P. Musso, A. Sgorbissa, and R. Zaccaria, "Surveillance robotics: Analyzing scenes by colors analysis and clustering," in *Proceedings of the 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA'03)*, July 2003.

[12] S. Katz, A. Chinn, and L. Cordrey, "Multidisciplinary studies of illness in aged persons: a new classification of functional status in activities of daily living," *Journal of Chronic Disease*, vol. 9, no. 1, pp. 55–62, 1959.

[13] T. Yamazaki, "Assistive technologies in smart homes," in *F. Mastrogiovanni and N.Y. Chong (Eds.) Handbook of Research on Ambient Intelligence and Smart Environments*, Hershey, Pennsylvania, USA, May 2011, pp. 165–181.

[14] F. Mastrogiovanni, A. Sgorbissa, and R. Zaccaria, "A cognitive model for recognizing human behaviours in smart homes," *Annales of Telecommunications*, vol. 65, no. 9, 2010.

[15] ——, "Activity recognition in smart homes: from specification to recognition," *International Journal of Fuzzy and Intelligent Systems*, vol. 21, no. 1-2, pp. 33–48, 2010.

[16] D. Olguin and A. Pentland, "Human activity recognition: accuracy across common locations for wearable sensors," in *Proceedings of the 2006 IEEE International Symposium on Wearable Computers (ISWC'06)*, Montreaux, Switzerland, October 2006.

[17] C. Dickerson, R. Hughes, and D. Chaffin, "Experimental evaluation of a computational shoulder musculoskeletal model," *Clinical Biomechanics*, vol. 23, no. 7, pp. 886–894, 2008.

[18] F. Allen, E. Ambikairajah, N. Lovell, and B. Celler, "Classification of a known sequence of motions and postures from accelerometry data using adapted gaussian mixture models," *Physiological Measurement*, vol. 27, no. 10, pp. 935–953, 2006.

[19] S. Calinon, F. D'Halluin, E. Sauser, D. Caldwell, and A. Billard, "Learning and reproduction of gestures by imitation," *IEEE Robotics & Automation Magazine*, vol. 6, pp. 44–54, 2010.

[20] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[21] P. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[22] H. Sung, *Gaussian Mixture Regression and Classification*. Houston, USA: Rice University Press, 2004.