Dimensionality Reduction for Whole-Body Human Motion Recognition

Christian Mandery, Matthias Plappert, Júlia Borràs, and Tamim Asfour
High Performance Humanoid Technologies Lab (H^{2}T)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
mandery@kit.edu, asfour@kit.edu

***Abstract*—We address the problem of feature space dimensionality reduction for the recognition of whole-body human action based on Hidden Markov Models. First, we describe how different features are derived from marker-based human motion capture and define a total number of 29 features with a total of 702 dimensions to describe human motion. We then propose a strategy for a systematic exploration of the space of possible subsets of these features and the identification of meaningful low-dimensional feature vectors for motion recognition. We evaluate our approach using a data set consisting of 353 motions grouped into 23 different types of whole-body actions. Our results show that a lower-dimensional feature space is sufficient to achieve a high motion recognition performance and that, using just four dimensions, we can achieve an accuracy of 94.76% on our data set, which is comparable to feature vectors that consider a much higher-dimensional feature like the joint angles.**

I. Introduction

Recognizing human actions is an important research field with major interest in areas which range from action understanding, imitation learning, natural human-robot interaction to human motion analysis and rehabilitation robotics. Today, state-of-the-art motion capture systems deliver an outstanding performance in recording human motion, allowing the collection of large-scale data sets of human whole-body motion, containing motion recordings for a variety of actions like the ones shown in Fig. 1. However, the growing number of data recorded also raises the question how this data can be classified and categorized in an automatic manner. Hence, much effort has already been put into developing motion recognition systems that can classify human motion into classes which correspond to different types of motions. The performance of these systems crucially depends on the representation of the motion data, i.e. which features are selected to constitute the feature vector.

In this work, we investigate whether there may be features useful for motion recognition that are not in common use today. For this purpose, we establish a list of 29 features categorized into six groups to describe whole-body human motion, which includes both very common features like joint angles, but also seldom considered features like the whole-body angular momentum. Subsets of these features are then used to constitute the feature vector in a motion recognition system based on Hidden Markov Models (HMMs). In general, it is preferable to reduce the dimensionality of this feature vector as much as possible, because such a dimensionality reduction allows for computational advantages, e.g. shorter training time, and reduces both the amount of data necessary to train the system and the risk of overfitting during training [1]. We therefore explore the space of possible subsets of features in a systematic way with focus on low-dimensional feature subsets. Our goal is to explore whether popular features like joint angles are truly the best choice, or whether other features or combinations thereof may allow for better motion recognition in terms of recognition performance or run-time.

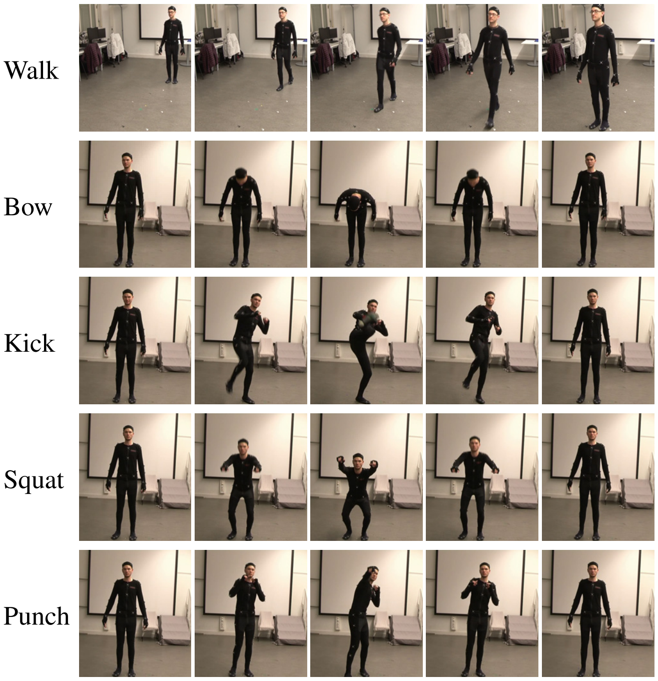

Fig. 1. Key frames from motions of the data set used for evaluation in this paper (see Section V-A).

To the best of the authors' knowledge, the problem of feature selection for whole-body human motion recognition has not been approached by a lot of authors. Freeman [2] provides a review of measures that can be used in filter methods for feature selection (see Section IV), and also includes two applications of the proposed methods to human whole-body motion using the data set presented in [3] based on joint angles and to hand gesture recognition from electromyography data. Ren et al. [4] present a silhouette-based vision interface resting upon an ensemble classifier using AdaBoost to select features. Although HMMs are very commonly used to represent human motion [5], [6], [7], [8], a large variety of other approaches exist, such as Factorial HMMs [9], [3], Parametric HMMs [10], [11], [12], neural networks [13], mixture models of Gaussian and Bernoulli distributions [14], tree structures [15], [16], Conditional Restricted Boltzmann Machines [17], [18], and more. Since the question of how human motion can be represented is a universal one and not limited to the learning of HMMs for motion recognition, the results of this work may be relevant for such other approaches as well.

The remainder of this paper is structured as follows. In Section II, we describe how motion data is acquired, define the features which we derive from captured motion data, and outline the processing steps required to obtain them. Then, in Section III, we discuss how HMMs can be used to recognize human motion. After that, we explain our strategy to find useful subsets of the computed features for motion recognition in Section IV. In Section V, we present the results of our evaluation and discuss our findings. Finally, we draw conclusions from the evaluation in Section VI.

II. Whole-Body Human Motion Features

A. Acquisition and Representation of Whole-Body Motion

In this work, we are considering motion data that has been acquired by marker-based motion capture. In our evaluation data set (see Section V-A), we are using motion recordings freely available from the KIT Whole-Body Human Motion Database [19]. These motion recordings are based on the KIT reference marker set, which consists of a total number of 56 markers, placed at characteristic anatomical landmarks of the human subject. The capture procedures and the marker set are described in [19] and online[1].

In addition to raw trajectories of the motion capture markers, we are also considering motion features based on the Master Motor Map (MMM) framework [20], [21]. The MMM provides an open-source framework for the representation and the analysis of human motion. Its fundamental goal is to provide a unifying representation for human motion, and to establish procedures for the reconstruction of motion in this representation from different input data, such as marker-based or marker-less motion capture systems. The procedures in the MMM framework to achieve this reconstruction from marker-based motion capture data are described in more detail in [21]. The result of the motion reconstruction are the joint angles and the 6D root pose (location and rotation) for every frame of the captured motion. This motion representation is based on the MMM reference model, which provides a model of the human body with kinematic and dynamic specifications derived from existing biomechanics literature by Winter [22] and de Leva [23]. The kinematic model offers 104 degrees of freedom (DoF): 6 for the root pose, 52 for the body torso, extremities, head, and eyes, and $2 \cdot 23$ for both hands. The dynamic model provides center of mass (CoM) and inertia tensor for every segment of the human body.

TABLE I
List of the features considered in this work, including their respective numbers of dimensions

Feature Group	Feature Name	Dimensions
Marker Features	marker_pos	$56 \cdot 3 = 168$
	marker_vel	$56 \cdot 3 = 168$
	marker_vel_norm	1
	marker_acc	$56 \cdot 3 = 168$
	marker_acc_norm	1
Joint Features	joint_pos	40
	joint_vel	40
	joint_vel_norm	1
	joint_acc	40
	joint_acc_norm	1
Root Pose Features	root_pos	3
	root_vel	3
	root_vel_norm	1
	root_acc	3
	root_acc_norm	1
	root_rot	3
	root_rot_norm	1
Center of Mass (CoM) Features	com_pos	3
	com_vel	3
	com_vel_norm	1
	com_acc	3
	com_acc_norm	1
End Effector Features	end_effectors_pos	$4 \cdot 3 = 12$
	end_effectors_vel	$4 \cdot 3 = 12$
	end_effectors_vel_norm	4
	end_effectors_acc	$4 \cdot 3 = 12$
	end_effectors_acc_norm	4
Angular Momentum Features	angular_momentum	3
	angular_momentum_norm	1

Table I provides an overview of all the considered features. We have categorized the features into the six groups shown in the first column of Table I. In the next subsections, we will explain each of them in more detail following the same order.

B. Marker Features

The feature *marker_pos* describes the Cartesian positions of the markers attached to the human subject. From the position $\mathbf{p}_t^i$ of the marker i at time step t given by *marker_pos*, we can approximate the velocity vector $\mathbf{v}_t^i$ using the common central difference for numerical differentiation:

$$\mathbf{v}_t^i = \frac{\mathbf{p}_{t+1}^i - \mathbf{p}_{t-1}^i}{2\Delta t} \qquad (1)$$

in which Δt represents the duration of one time step. The feature *marker_vel* then describes the Cartesian velocity vectors of the 56 markers. *marker_vel_norm* combines all velocity vectors into a scalar value using the Euclidean norm. In a similar way, from *marker_vel* the features *marker_acc* and *marker_acc_norm* are computed, which describe the accelerations of the markers.

[1] https://motion-database.humanoids.kit.edu/marker_set/

C. Joint Features

The *joint_pos* feature consists of the joint values for the 40 joints of the MMM reference model that are used to represent human whole-body motion (e.g. fingers and eyes are excluded). *joint_vel* describes the joint angle velocities, calculated from the joint trajectories using equation 1. *joint_vel_norm* denotes the norm of all joint velocities, providing an indication for the current total movement in joint space. In addition to the velocities, joint accelerations (*joint_acc*) and acceleration norm (*joint_acc_norm*) are computed as well.

D. Root Pose Features

Together with the joint values, the motion of the MMM model also provides us with the 6D pose (location and rotation) of the model root, which is located inside the hip of the model. *root_pos* describes the Cartesian position of the model root, *root_vel* its velocity vector calculated using equation 1, and *root_vel_norm* the norm of the velocity vector, providing an indication for how fast the subject is moving. Similarly, *root_acc* provides the acceleration vector and *root_acc_norm* the acceleration norm. The rotation of the model root is described by *root_rot* as roll-pitch-yaw angles. The feature *root_rot_norm* represents the norm of *root_rot*, giving an indication of how much the model root is turned.

E. Whole-Body Center of Mass (CoM) Features

Given the motion in the MMM format defined by joint angles and 6D root pose, additional features can be derived from the MMM model and the associated kinematic and dynamic specifications for the human body.

The dynamic model allows us to compute the CoM of the human body, which is represented by the feature *com_pos*. At first glance, the CoM seems very similar to *root_pos*, however, unlike the CoM, the root position is fixed to a specific point inside the hip of the model. Therefore, depending on the motion type, the CoM trajectories differ strongly from the model root trajectories. Take, for example, upper-body dominant motions like waving, where the model root does not move significantly, in contrast to the CoM. For bowing, the model root fixed moves slightly backward, while the CoM moves in the opposite direction, i.e. forward and downward. Velocity and acceleration vectors and the corresponding scalar speed and acceleration values of the CoM (*com_vel*, *com_vel_norm*, *com_acc*, and *com_acc_norm*) are approximated using equation 1.

F. End Effector Features

From the MMM model, we can also derive the locations of the extremities as a virtual tool center point located in the center of the hand palm or the posterior of the foot respectively. *end_effectors_pos* provides the positions for all four extremities of the human body. Similarly to before, *end_effectors_vel*, *end_effectors_vel_norm*, *end_effectors_acc*, and *end_effectors_acc_norm* can be computed using equation 1.

G. Whole-Body Angular Momentum Features

The angular momentum is a physical measure that represents the rotational equivalent to linear momentum of a system in 3D space. The characteristic and well-studied progression of the angular momentum for certain motion types like walking [24], [25] makes this measure a potentially very promising feature for the recognition of these motions.

The *angular_momentum* feature represents the three-dimensional whole-body angular momentum $\mathbf{L}$ along the three axes of the model coordinate system. From the N segments of the MMM model, it can be computed as follows:

$$\mathbf{L} = \sum_{i=1}^{N} \Big[m_i(\mathbf{r}_i^c \times \mathbf{v}_i^c) + \mathbf{I}_i^c \boldsymbol{\omega}_i \Big] \;\;, \;\mathbf{L} \in \mathbb{R}^3 \qquad (2)$$

with

$$\begin{aligned} \mathbf{r}_i^c &= \mathbf{r}_{CoM_i} - \mathbf{r}_{CoM} \\ \mathbf{v}_i^c &= \dot{\mathbf{r}}_{CoM_i} - \dot{\mathbf{r}}_{CoM}. \end{aligned}$$

The first part of the sum in equation 2 considers the angular momenta created by the orbital rotation of each segment around the CoM. m_i describes the mass of segment i, $\mathbf{r}_i^c$ its position, and $\mathbf{v}_i^c$ its velocity, both given with respect to the CoM. × denotes the cross product. The spin of each segment is taken into account by the second part of the sum, which is computed by forming the product of its inertia tensor $\mathbf{I}_i^c$ and its angular velocity $\boldsymbol{\omega}_i$. The *angular_momentum_norm* feature represents the norm of the angular momentum, which provides an indication for the total spin of the human body.

H. Feature Processing

Additional processing steps are necessary for the computation of the features mentioned above to make these features useful for motion recognition.

1) Smoothing: Marker trajectories from the motion capture system and the basic MMM features provided by the MMM motion reconstruction algorithm might contain noise or jitter. Since such errors are amplified by the repeated differentiation used in equation 1 to calculate velocities and accelerations, we use a moving average filter with a window size of 3 to smooth the data.

2) Normalization: It is important to note that motion features defined in Cartesian space need to be normalized to be useful. For example, trajectories of the model root or the CoM for the exactly same motion look very different when observed in the global coordinate system if the starting location or orientation is different. Therefore, these features are normalized with respect to the initial root pose of the motion. Given the 4×4 transformation matrix $\mathbf{T}_0$ of the model root pose for the first frame of the motion, the Cartesian feature $\mathbf{x}$ given in 4D homogeneous coordinates is normalized as:

$$\hat{\mathbf{x}}_t = \mathbf{T}_0^{-1} \mathbf{x}_t.$$

In a similar way, *end_effectors_pos* is normalized with respect to the current root pose of the model.

3) Scaling: When comparing joint angles in radians and Cartesian measures in millimeters, it is obvious that the described features live on different scales. As we will see in Section III, this can become a problem when *k-means clustering* is used to initialize the emission distribution parameters of an HMM, since the Euclidean distance measure used by the clustering algorithm does not consider the different extent of the features across dimensions. Therefore, the scalar x for each dimension of a feature is scaled roughly to the range $[-1, 1]$:

$$\hat{x}_t = 2 \cdot \frac{x_t - x_{min}}{x_{max} - x_{min}} - 1.$$

For each dimension of each feature, x_{min} and x_{max} are constant and determined from a sufficiently large set of training data.

III. Motion Recognition with HMMs

To recognize human motion, we are building upon Hidden Markov Models (HMMs), which have been proven to be very suitable for the modeling of time series data, such as human motion [26], [27]. The dynamics of the stochastic process underlying the HMM is based on a single unobservable latent variable that can take on K discrete states. Each of these states is associated with a probability distribution and the emission distribution that models the observation. The transition probabilities between the K states and the parameters of the K emission distributions are learned when the HMM is trained. A more in-depth discussion of HMMs and their use can be found in existing literature like [28], [29].

In this work, we selected hyper-parameters for the HMMs that are popular in literature where HMMs are already used to represent human motion [7], [30], and verified this choice by own experiments. More concretely, we are using HMMs with observations modelled as Gaussian distributions, a left-to-right (Bakis) topology and $K = 8$ states. The covariance matrices of the HMMs are constrained to be diagonal. For training, means and covariances of the emission distributions of the HMMs are initialized using the *k-means algorithm* to cluster the data into K clusters, corresponding to HMM states. Transition and start probabilities of the HMMs are initialized uniformly. To learn the HMM parameters, we run the Baum-Welch algorithm for 10 iterations, or until convergence (Δlog-likelihoods $< 10^{-2}$).

To solve the multi-class classification problem of motion recognition, where one motion should be assigned to exactly one of the learned motion classes, we are training one HMM for each motion class using the motions associated with that class. To classify an unseen motion, we then determine its log-likelihood under each of the trained HMMs and assign the motion to the class which belongs to the HMM with the highest log-likelihood.

IV. Feature Selection

In literature from machine learning, approaches for feature selection are commonly grouped into three different classes: *embedded methods*, *filter methods*, and *wrapper methods* [1]. *Embedded methods* perform feature selection as part of the training process and are therefore specific to the learning algorithm. While solutions for certain classifiers such as Support Vector Machines are well-established, we do not know of any *embedded method* that can readily be used for an HMM with time series data. *Filter methods* use a given filter measure to rank individual features without actually training the classifier, which makes them potentially very efficient. A very large number of such filter measures is available and selecting a good one is crucial when using such a method [31]. However, the choice of the filter measure can be challenging as it highly depends on the classifier to be used subsequently and the data set. Also, since *filter methods* consider individual features, they cannot recognize relationships between features, which can result in the selection of redundant features. Then, there are *wrapper methods* [32] that treat the learning algorithm as a black box, evaluating the usefulness of a feature set by the classification performance of the classifier trained with this feature set. A search strategy must be chosen to efficiently search the space of feature subsets (FSS), because considering all possible FSS is computationally rarely possible. *Wrapper methods* have the advantage that they can consider relationships between given features when determining good subsets of features, e.g. only using one feature from a set of highly correlated, therefore more or less redundant, features. They have the disadvantage of a high computational effort because every evaluation requires model training, and the risk of overfitting if not enough data is provided for the given set of features.

Due to the considerations given above, we are using the *wrapper method* for feature selection in this paper. Of course, given the numbers of features available, it is computationally not feasible to perform an exhaustive search in the space of possible FSS, since $2^N - 1$ possible FSS can be constructed from N features. Thus, more sophisticated metaheuristics are necessary to explore this space. Traditional works for feature selection often employ a strategy of *forward selection*, that starts with an empty FSS and iteratively adds the feature to the FSS that increases recognition performance most. This approach however has two downsides. First, the algorithm does not consider the dimensionality of the available features, and will prefer a very high-dimensional feature over a low-dimensional one, even if the difference in terms of score improvement is very small. Hence, the resulting FSS usually corresponds to a high-dimensional feature vector. Second, this approach represents a *greedy algorithm* in that only the single best FSS is considered in each iteration, which makes it likely that the globally best FSS may be missed.

Therefore, in this work, we use a modified forward exploration strategy that is given in pseudocode in Algorithm 1. Our algorithm iterates from 1 to a given maximum dimensionality (line 3), which can be as high as the cumulative number of dimensions of all available features (702 in our case). In iteration k, our algorithm considers only FSS with k dimensions. To build such FSS, we are taking previously evaluated lower-dimensional FSS from all past iterations and add a single not yet contained feature to them in order to create FSSs of

Algorithm 1 N-Best Feature Subset Exploration

```
 1: $possibleFeatures \leftarrow \{...\}$ (see Table I)
 2: $fssForDim[0] \leftarrow \{\emptyset\}$
 3: for $curDim \leftarrow 1$ to $maxDim$ do
 4:     // Build feature subsets of dimensionality $curDim$
 5:     $fssToEvaluate \leftarrow \{\}$
 6:     for $pastDim \leftarrow 0$ to $curDim - 1$ do
 7:         for all $oldFss \in fssForDim[pastDim]$ do
 8:             for all $feature \in possibleFeatures$ do
 9:                 if $pastDim + \dim(feature) = curDim$
                        and $feature \notin oldFss$ then
10:                     $newFss \leftarrow oldFss + \{feature\}$
11:                     add $newFss$ to $fssToEvaluate$
12:                 end if
13:             end for
14:         end for
15:     end for
16:
17:     // Evaluate feature subsets in $fssToEvaluate$
18:     for all $fss \in fssToEvaluate$ do
19:         for $round \leftarrow 1$ to $3$ do
20:             split data set into training and test folds
21:             train HMMs with data from training folds
22:             classify motions from test fold with HMMs
23:         end for
24:         compute and save weighted average of the
                $F_1$ scores for all HMM classifiers
25:     end for
26:
27:     // Keep n-best feature subsets of dim. $curDim$
28:     $fssForDim[curDim] \leftarrow$ n-best($fssToEvaluate$)
29: end for
```

dimensionality k (lines 6-15). Then, the performance of these newly created FSS for motion recognition is determined (lines 18-25). For this purpose, we are considering the F_1 score, which is a well-established measure to estimate classification performance based on *precision* and *recall* of a classifier [33]. For the evaluation of a given FSS, a stratified 3-fold cross validation is used, in which the data set is equally divided into three subsets, the folds (line 20). In three successive rounds, each of the three folds is used once as the test set, with the other two folds being used to train the model (lines 21-22). Stratification ensures that each fold contains a distribution of the motion classes that is representative for the whole data set [34]. At the end of the three rounds, every motion in the data set has been used for testing exactly once and the F_1 score is calculated using the classification results from all rounds. Since we are training one HMM per motion type, we compute the weighted average of the F_1 scores for each HMM to get a scalar measure for the recognition performance of the evaluated FSS (line 24). The contribution of each F_1 score to this average is proportional to the *support* of the corresponding motion type, i.e. the number of representative motions for this type contained in the data set. Similarly to the F_1 scores themselves, this measure is from the range $[0, 1]$, with higher values indicating better recognition performance. The best $n = 10$ of the evaluated FSS are kept in each iteration (line 28) as part of the result of the algorithm and to create higher-dimensional FSS in future iterations.

TABLE II
MOTION DATA SET USED FOR EVALUATION (IDS CORRESPOND TO THE RECORDS IN THE KIT WHOLE-BODY HUMAN MOTION DATABASE)

Motion Type	# Rec.	ID(s)
Walk	49	318, 362, 395, 452, 467
Run	41	324, 364, 399, 426, 533
Turn	59	326, 327, 402, 403, 445, 446
Pushed from Behind	9	476, 477, 478
Throw	10	573, 581
Kneel	5	515
Bow	10	582, 609
Kick	20	610, 611, 612, 613
Squat	5	616
Punch	10	617, 618
Stomp	10	619, 620
Jump	25	621, 622, 623, 624, 625
Golf Putt	5	626
Golf Drive	6	627
Tennis Smash	10	628, 629
Tennis Forehand	10	630, 631
Wave	15	633, 634, 635
Play Guitar	11	636, 637
Play Violin	10	638, 639
Stir	11	640, 641
Wipe	11	642, 643
Dance Waltz	6	644
Dance Cha-Cha-Cha	5	645
— Total —	**353**	

V. EVALUATION

A. Data Set

The data set used for evaluation comprises 353 different motion recordings captured from 9 different subjects (6 male, 3 female), and has been recorded using a passive optical Vicon MX motion capture system [35] consisting of ten T10 cameras. Table II provides an overview of the different motion types used for evaluation with the corresponding number of recordings available. Fig. 1 shows manually selected key frames from some exemplary motions. We tried to execute motions in different modalities, e.g. switching between hands for single-handed motions like *punch*, *golf*, or *tennis*. For periodic motions such as stirring or dancing, we varied the number of repetitions across the recorded trials.

All motions in our data set can be freely retrieved from the KIT Whole-Body Human Motion Database [19], available as raw marker data in the C3D file format and motions in the MMM format, as well as the corresponding video recordings. To alleviate finding them, the direct URL `https://motion-database.humanoids.kit.edu/details/motions/<ID>/` can be used in conjunction with the respective IDs given in Table II.

TABLE III
BEST FEATURE SUBSETS PER DIMENSIONALITY FROM ALGORITHM 1

D.	Feature Subset	F_1 Score	Accuracy
1	marker_vel_norm	51.67%	54.45%
2	com_vel_norm, marker_vel_norm	82.28%	82.20%
3	angular_momentum_norm, com_vel_norm, marker_vel_norm	88.66%	88.48%
4	com_vel, joint_vel_norm	94.76%	94.76%
5	com_acc_norm, com_vel, marker_vel_norm	94.18%	94.24%
6	com_vel, com_vel_norm, marker_vel_norm, root_vel_norm	94.24%	94.24%
7	angular_momentum_norm, com_vel, joint_vel_norm, marker_vel_norm, root_vel_norm	94.27%	94.24%
8	com_vel, end_effectors_vel_norm, joint_vel_norm	95.77%	95.81%
9	com_vel, end_effectors_vel_norm, joint_vel_norm, root_vel_norm	95.28%	95.29%
10	com_acc_norm, com_vel, com_vel_norm, end_effectors_vel_norm, joint_vel_norm	95.75%	95.81%
11	com_pos, com_vel, end_effectors_vel_norm, joint_vel_norm	95.97%	96.34%
12	angular_momentum, com_vel, end_effectors_vel_norm, marker_vel_norm, root_acc_norm	96.32%	96.34%
13	com_vel_norm, end_effectors_vel	97.32%	97.38%
14	end_effectors_vel, marker_vel_norm, root_rot_norm	97.87%	97.91%
15	com_acc_norm, com_vel_norm, end_effectors_vel, root_vel_norm	97.89%	97.91%

B. Experimental Results

As described in Section IV, the evaluation of a given FSS consists of three cross-validation rounds. In each round, HMMs for each of the motion types are learned from the training fold, and then used to compute the log-likelihoods under each HMM for the motions in the respective test fold. Depending on the dimensionality of the feature vector, such an evaluation takes approximately between 15 and 90 seconds. Computation of all results presented in this section then takes around eight hours. In the following, in addition to the F_1 scores used as the measure for feature selection, we also provide values for the recognition accuracy, defined as the percentage of motions that are assigned to the correct motion type by the system.

Table III shows the results of our feature selection algorithm and provides the best found FSS for every given dimensionality of the feature vector up to 15 dimensions. We verified the validity of our approach by running an exhaustive search considering all FSS with at most five dimensions, and can confirm that Table III indeed provides the best possible FSS for these dimensionalities. Since this exhaustive search already requires 16055 FSS to be evaluated and thus takes half a day even in a parallelized implementation, it is not feasible to perform such a validation for higher-dimensional FSS.

As explained in Section IV, our metaheuristics used for search is not strictly greedy, but considers the ten best FSS for every dimensionality to provide a broader exploration of the space of possible FSS. Fig. 2 shows the F_1 scores of these ten best FSS, again for feature vectors with up to 15 dimensions. As can be seen, starting with five dimensions, recognition performance does not significantly diminish within these ten best FSS, indicating that a larger number of "good" FSS exist that can provide a comparable performance.

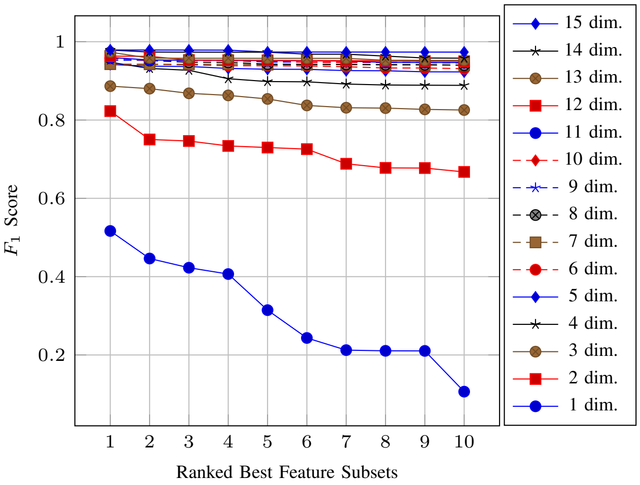

Fig. 2. F_1 scores for the ten best FSS of every dimensionality found by Algorithm 1, up to 15 dimensions. The X axis ranks the FSS of a given dimensionality from the best to the 10th best.

TABLE IV
COMPARISON OF SELECTED FEATURE SUBSETS, ALSO PROVIDING THE TIME NECESSARY TO TRAIN THE SYSTEM

Feature Subset	Dim.	F_1 Score	Accuracy	Tr. T.
com_vel, joint_vel_norm	4	94.76%	94.76%	19s
com_vel, joint_vel_norm, end_effectors_vel_norm	8	95.77%	95.81%	21s
root_pos, root_rot	6	87.17%	87.54%	20s
end_effectors_pos	12	95.22%	95.29%	25s
end_effectors_vel	12	94.15%	94.05%	25s
joint_pos	40	87.17%	88.10%	39s
joint_vel	40	86.84%	86.97%	48s
marker_pos	168	90.29%	90.37%	108s
marker_vel	168	94.44%	94.62%	120s

Table IV compares two of the FSS shown in Table III to manually selected FSS that are commonly used for motion recognition. In addition to F_1 score and accuracy, we also provide the time required for training, accumulated over the three cross-validation rounds. From the manually selected FSS, we can see that using joint angles as a feature does not provide a performance comparable to our selected FSS. However, considering the positions of the end effectors provides a feature that works quite well on our data set, which is not that surprising, given the nature of whole-body motion and the motion types used for evaluation. Although the results cannot be shown here for space reasons, an evaluation of all features even showed that *end_effectors_pos* is the best of all of our implemented features, if each feature is evaluated in isolation. It should be noted though that the normalization described in Section II-H2, i.e. representation of end effector positions with respect to the model coordinate system, is required to make this feature so useful. Looking at the first two low-dimensional FSS listed in Table IV that have been found by our search strategy, we can see that both offer a performance comparable to or even better than using end effector positions, the whole set of marker velocities, or all the other shown high-dimensional features. Because these FSS use less dimensions (4 and 8) to represent the motion, they take less time for training, and should also require less training data and exhibit a reduced risk of overfitting.

We noticed during our evaluation that, aside from end effector positions and velocities, the whole-body center of mass (CoM) seems to be a very significant feature for the recognition of human whole-body motion. Indeed, almost all of the FSS found by our exploration, which represent the ten best FSS per dimensionality, contain at least one of the features describing the velocity of the CoM (*com_vel* or *com_vel_norm*). In general, for vector quantities like velocity or acceleration vectors, it often seems to be sufficient to use the norm of the vector as a scalar feature instead of the vector itself, indicating that a large portion of the informative value of these features is already provided by the magnitude of motion and not its direction. Also, while the angular momentum does occur in some of the selected FSS, it does not seem to be as useful as we initially thought. One possible explanation for this is that only a minority of the motions used in our data set, e.g. *walking*, *golf drive*, or *tennis smash*, exhibit enough whole-body drive to render the angular momentum a useful feature.

C. Future Work

In the future, we are planning to evaluate our approach with larger data sets and to investigate whether our findings generalize to other problems such as multi-label classification, where one motion can be assigned to an arbitrary number of labels. Also, it would be interesting to consider a motion recognition system using Factorial Hidden Markov Models [9] with the sequential training algorithm presented in [3], although the higher training time might require a modified exploration strategy.

Since we considered marker positions and joint angles only as one single feature respectively, it remains an open question if there are markers or joints that are less important for motion recognition and can thus be left out to reduce the dimensionality. Furthermore, we want to evaluate additional features for the description of human motion, e.g. features based on support contacts [36] or features based on the effort keys presented in [16], which are derived from Laban Movement Analysis [37].

VI. Conclusions

In this paper, we showed that it is possible to achieve a high performance for whole-body human motion recognition with using a low-dimensional representation of the motion as the feature vector. As mentioned in the introduction, such a dimensionality reduction allows for computational advantages and also reduces the amount of training data needed. We started by defining 29 features with a total number of 702 dimensions that can be used to represent human whole-body motion. Then, we proposed an exploration strategy to search the space of all possible feature subsets for meaningful low-dimensional subsets of features. Our evaluations showed that there exist a large number of promising low-dimensional feature subsets and that, given the right choice of features, a feature vector consisting of just eight dimensions is sufficient to outperform existing higher-dimensional features for motion recognition tasks, like the positions or velocities of joints or markers.

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework Programme under grant agreement no. 611909 (KoroiBot) and from the European Union H2020 Programme under grant agreement no. 643666 (I-SUPPORT).

References

[1] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[2] C. Freeman, "Feature selection and hierarchical classifier design with applications to human motion recognition," Ph.D. dissertation, University of Waterloo, 2014.

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

[4] L. Ren, G. Shakhnarovich, J. K. Hodgins, H. Pfister, and P. Viola, "Learning silhouette features for control of human motion," *ACM Transactions on Graphics (ToG)*, vol. 24, no. 4, pp. 1303–1331, 2005.

[5] 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," *International Journal of Robotics Research*, vol. 31, no. 3, pp. 330–345, 2012.

[6] W. Takano and Y. Nakamura, "Statistical mutual conversion between whole body motion primitives and linguistic sentences for human motions," *International Journal of Robotics Research*, vol. 34, no. 10, pp. 1314–1328, 2015.

[7] W. Takano, K. Yamane, T. Sugihara, K. Yamamoto, and Y. Nakamura, "Primitive communication based on motion recognition and generation with hierarchical mimesis model," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 3602–3609.

[8] T. Asfour, P. Azad, F. Gyarfas, and R. Dillmann, "Imitation learning of dual-arm manipulation tasks in humanoid robots," *International Journal of Humanoid Robotics*, vol. 5, no. 2, pp. 183–202, 2008.

[9] D. Kulić, W. Takano, and Y. Nakamura, "Representability of human motions by factorial hidden markov models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 2388–2393.

[10] V. Krüger, D. L. Herzog, S. Baby, A. Ude, and D. Kragic, "Learning actions from observations," *IEEE Robotics & Automation Magazine*, vol. 17, no. 2, pp. 30–43, 2010.

[11] D. Herzog, A. Ude, and V. Krüger, "Motion imitation and recognition using parametric hidden markov models," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2008, pp. 339–346.

[12] A. D. Wilson and A. F. Bobick, “Parametric hidden markov models for gesture recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 884–900, 1999.

[13] A. Baca, “Methods for recognition and classification of human motion patterns – a prerequisite for intelligent devices assisting in sports activities,” *MATHMOD*, 2012.

[14] S. Calinon, F. Guenter, and A. Billard, “On learning, representing, and generalizing a task in a humanoid robot,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 2, pp. 286–298, 2007.

[15] K. Yamane, Y. Yamaguchi, and Y. Nakamura, “Human motion database with a binary tree and node transition graphs,” *Autonomous Robots*, vol. 30, no. 1, pp. 87–98, 2011.

[16] M. Kapadia, I.-k. Chiang, T. Thomas, N. I. Badler, J. T. Kider Jr *et al.*, “Efficient motion retrieval in large motion databases,” in *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D)*, 2013, pp. 19–28.

[17] G. W. Taylor and G. E. Hinton, “Factored conditional restricted boltzmann machines for modeling motion style,” in *International Conference on Machine Learning (ICML)*, 2009, pp. 1025–1032.

[18] G. W. Taylor, G. E. Hinton, and S. T. Roweis, “Modeling human motion using binary latent variables,” in *Advances in Neural Information Processing Systems*, 2006, pp. 1345–1352.

[19] C. Mandery, O. Terlemez, M. Do, N. Vahrenkamp, and T. Asfour, “The KIT whole-body human motion database,” in *International Conference on Advanced Robotics (ICAR)*, 2015, pp. 329–336.

[20] P. Azad, T. Asfour, and R. Dillmann, “Toward an unified representation for imitation of human motion on humanoids,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 2558–2563.

[21] O. Terlemez, S. Ulbrich, C. Mandery, M. Do, N. Vahrenkamp, and T. Asfour, “Master Motor Map (MMM) – framework and toolkit for capturing, representing, and reproducing human motion on humanoid robots,” in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2014, pp. 894–901.

[22] D. A. Winter, *Biomechanics and motor control of human movement*, 4th ed. Wiley, 2009.

[23] P. de Leva, “Adjustments to zatsiorsky-seluyanov’s segment inertia parameters,” *Journal of Biomechanics*, vol. 29, no. 9, pp. 1223–1230, 1996.

[24] H. Herr and M. Popovic, “Angular momentum in human walking,” *Journal of Experimental Biology*, vol. 211, no. 4, pp. 467–481, 2008.

[25] M. Popovic and A. Englehart, “Angular momentum primitives for human walking: biomechanics and control,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, 2004, pp. 1685–1691.

[26] T. Inamura, I. Toshima, H. Tanie, and Y. Nakamura, “Embodied symbol emergence based on mimesis theory,” *International Journal of Robotics Research*, vol. 23, no. 4–5, pp. 363–377, 2004.

[27] J. Yamato, J. Ohya, and K. Ishii, “Recognizing human action in time-sequential images using hidden markov model,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1992, pp. 379–385.

[28] 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.

[29] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[30] D. Kulić, W. Takano, and Y. Nakamura, “Incremental on-line hierarchical clustering of whole body motion patterns,” in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2007, pp. 1016–1021.

[31] C. Freeman, D. Kulić, and O. Basir, “An evaluation of classifier-specific filter measure performance for feature selection,” *Pattern Recognition*, vol. 48, no. 5, pp. 1812–1826, 2015.

[32] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” *Artificial Intelligence*, vol. 97, no. 1, pp. 273–324, 1997.

[33] C. J. V. Rijsbergen, *Information Retrieval*, 2nd ed. Butterworth-Heinemann, 1979.

[34] R. Kohavi *et al.*, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, 1995, pp. 1137–1145.

[35] “Vicon Motion Capture,” Website, available online at http://www.vicon.com.

[36] C. Mandery, J. Borràs, M. Jöchner, and T. Asfour, “Analyzing whole-body pose transitions in multi-contact motions,” in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 1020–1027.

[37] R. Laban and L. Ullmann, *The mastery of movement*. MacDonald & Evans, 1971.