Human Activity Recognition using Max-Min Skeleton-based Features and Key Poses

Urbano Miguel Nunes, Diego R. Faria, and Paulo Peixoto

Abstract—**Human activity recognition is still a very challenging research area, due to the inherently complex temporal and spatial patterns that characterize most human activities. This paper proposes a human activity recognition framework based on random forests, where each activity is classified requiring few training examples (i.e. no frame-by-frame activity classification). In a first approach, a simple mechanism that divides each action sequence into a fixed-size window is employed, where max-min skeleton-based features are extracted. In the second approach, each window is delimited by a pair of automatically detected key poses, where static and max-min dynamic features are extracted, based on the determined activity example. Both approaches are evaluated using the Cornell Activity Dataset [1], obtaining relevant overall average results, considering that these approaches are fast to train and require just a few training examples. These characteristics suggest that the proposed framework can be useful for real-time applications, where the activities are typically well distinctive and little training time is required, or to be integrated in larger and sophisticated systems, for a first quick impression/learning of certain activities.**

Index Terms—**Human Daily Activity Recognition, Random Forest, Max-Min Skeleton-based Features, Key Poses, Static and Dynamic Features**

I. Introduction

ROBOT perception is still an open area of research mainly due to the complexity that characterize the dynamic environment that surrounds the robot on real-world application scenarios. This is particularly true when the robot needs to interact with humans, like in the case of assistive robots, which should be able to quickly assess and react to potential critical situations. So, human activity recognition should play an important role on any autonomous robot perception module. In this context, this paper contributes with the proposal of two approaches for human activity recognition, both thought for real-time application scenarios, where characteristics like the number of training samples and the time needed for training play an important role. Simple max-min features are extracted, within a defined activity window, to train a random forest classifier. Few training examples are used to train this classifier. The main contributions of this work are the following:

- Two simple and effective approaches to extract extremal skeleton information, based on max-min features;
- Very fast training, requiring few training examples, properties that are real-time oriented.

These contributions, as well as the relevant performance obtained, when evaluated using *state of the art* dataset, may serve as a solid human activity recognition framework for real-time applications. The remainder of the paper is organized as follows: section II briefly describes the relevant related work, highlighting the contributions that the proposed approaches provide; section III explains both developed approaches for features extraction; section IV presents the results of the experimental procedures used to evaluate the proposed methods; section V summarizes the key ideas proposed, as well as to ongoing work, introducing some new lines of research for the near future.

II. Related Work

Human activity recognition has been a very active topic of research. Typically, recent approaches for human activity recognition rely on two sensing modalities: depth data and 3D joints position data. In [2], a method based on depth images and temporal ordering of unique poses is presented. In [3], a method to predict in real-time 3D body joints position from a single depth image from a RGB-D sensor was proposed. The most representative works based on 3D skeleton joints position data are the following: interaction of a subset of human joints [4]; eigenjoints descriptor, which incorporates static postures, motion and overall dynamics [5]; temporal key poses, based on the skeleton kinetic energy [6]; a dynamic Bayesian mixture model for classification [7]; spatio-temporal evolution of 3D postures [8]; self-organizing growing when required networks to learn spatio-temporal dependencies [9]; key poses association, using clustering algorithms without the need of a learning algorithm [10]; multi-layer codebooks of key poses and atomic motions, representing patterns of a certain human activity [11]. Random forests have also been used to classify human activities [12], as well as human gestures [13].

The two approaches presented in this paper, extract a set of features from 3D skeleton joints position data and use a random forest algorithm for classification. The first approach uses a fixed-size window to extract a set of features that describe each human activity, while the second one uses a variable size window, delimited by a pair of automatically detected key poses. The rationale for using the key poses is that they depict extreme points in the motion path of each joint, where most of the discriminative properties of each action are encoded. An overview of the proposed approach is shown in

Urbano Miguel Nunes and Paulo Peixoto are with Institute of Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra, Polo II, 3030-290 Coimbra, Portugal (emails: urbanomiguel.g.nunes@ieee.org, peixoto@isr.uc.pt).

Diego Faria is with the System Analytics Research Institute, Aston University, Birmingham, UK (email: d.faria@aston.ac.uk).

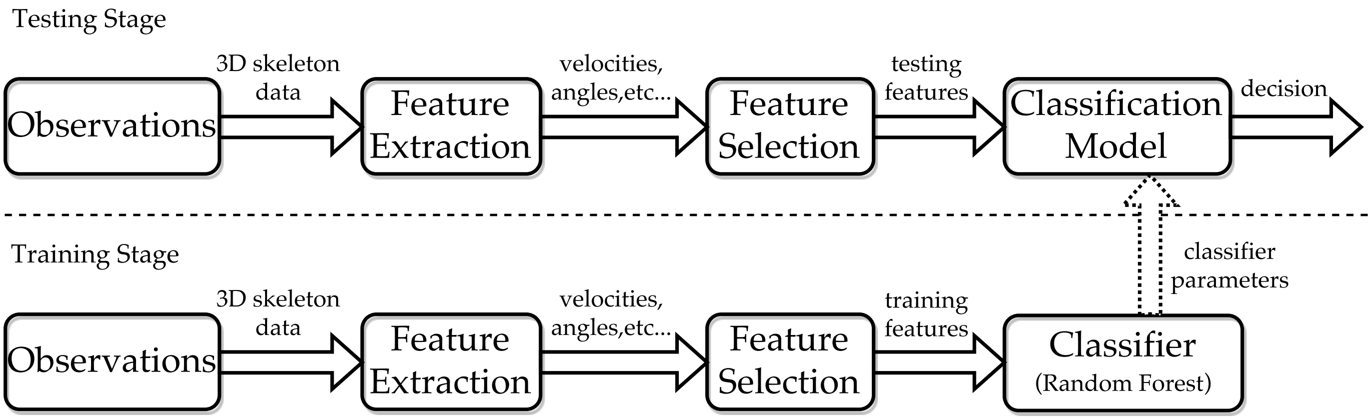

Fig. 1. Overview of the proposed approach. First, a human activity scene is observed and 3D skeleton data is collected. Then, the considered features are extracted (e.g. velocities, distances between joints). The "Feature Selection" block is generic for both developed approaches, where max-min features and static and max-min dynamic features (approach I and II, respectively) are obtained. In the training stage, a classifier model is built using a random forest as the classification algorithm, where each decision tree is constructed with the CART algorithm. The obtained classification model is then used during the testing stage.

Fig. 1. Both approaches aim to be fast to train, requiring few training examples and low computational cost, with relevant accuracy and precision, compared to other *state of the art* methods.

III. Proposed Approach

Let p_{jd}^t be the value of the coordinate $d \in \{x, y, z\}$ of the joint $j \in \{1, ..., m\}$ (m is the number of skeleton joints), at frame t. Therefore, $P_j^t = \left(p_{jx}^t, p_{jy}^t, p_{jz}^t\right)$ is the 3D vector that contains the information about the position of the joint j at frame t. The coordinates system (x, y, z) is defined as follows, relatively to the camera (see Fig. 2): x corresponds to the width; y corresponds to the height; z corresponds to the depth.

A. Preprocessing of 3D Skeleton Data

Before the process of feature extraction is performed, a preprocessing step is applied to the raw 3D skeleton data in order, not only to attenuate noise introduced by the sensor, but also to normalize the data to accommodate for different user's height, limb length, orientation and position. This preprocessing stage consists on the following steps:

1) **Translation**, to define the same origin of coordinates system for all frames; the selected one corresponds to the *torso* of the human skeleton;
2) **Normalization**, to reduce the influence of different user's height and limb length; first, the height of the subject is calculated; then all skeleton 3D coordinates are normalized according to the calculated value;
3) **Rotation**, to guarantee that the activity is always observed from the same point of view, independently of the initial pose of the subject with regard to the sensor; the rotation of the skeleton is performed in the y axis, considering the plane formed by the *torso*, *right* and *left hip* in relation to a fronto-parallel plane to the sensor;
4) **Symmetrization**, to disambiguate between right and left-handed people; since the skeleton is already in the same fronto-parallel pose in relation to the camera, it is just necessary to consider a new sample based on a mirrored version of the original 3D skeleton data.

B. Spatio-Temporal Features

The considered features may be divided into two categories: static (e.g. geometrical) and dynamic (e.g. temporal) features. The static features are intended to give information about *key poses*, which are obtained in frames where the pose has a kinetic energy equal to zero [6]. These poses represent extremal positions of a skeleton, which may be used to segment and recognize activities. This information is explored in subsequent sections. Additionally, dynamic features are intended to describe skeleton movements between key poses.

1) Static Features:

- **Projected distances** between two joints

$$\delta_{ab}^t = \sqrt{\sum_d \left(p_{ad}^t - p_{bd}^t\right)^2}, \tag{1}$$

where d belongs to one of the following sets $\{x, y\}$, $\{y, z\}$ and $\{z, x\}$ for each projection considered;

- **Projected angles** based on three joints

$$\theta_{id_p}^t = \arccos\left(\frac{(\delta_{ab}^t)^2 + (\delta_{bc}^t)^2 - (\delta_{ac}^t)^2}{2 \cdot \delta_{ab}^t \cdot \delta_{bc}^t}\right), \tag{2}$$

where δ corresponds to the Euclidean distance between two joints, given by (1), where $d_p \in \{xy, yz, zx\}$ for each projection considered;

- **Normal vector to triangles** formed by three joints

$$\Delta_k^t = \frac{(P_a^t - P_b^t) \times (P_a^t - P_c^t)}{||(P_a^t - P_b^t) \times (P_a^t - P_c^t)||}; \tag{3}$$

- **Sum of log-cov energy entropy** based on the global skeleton joints positions

$$(l_{\text{cov}p})^a = \sum_i \mathrm{U}_i \left\{\log\left(\text{cov} A_p^a\right)\right\}^2 \tag{4}$$

and based on the considered angles

$$(l_{\text{cov}\theta})^a = \sum_i \text{U}_i \left\{\log\left(\text{cov}A_\theta^a\right)\right\}^2, \tag{5}$$

where A_p^a and A_θ^a are matrices containing the values of skeleton joints positions and angles, respectively, associated to an activity a; cov represents the covariance matrix; log is the matrix logarithm and $\text{U}(\cdot)$ returns the upper triangular matrix elements. The idea of using the log-covariance is based on the work of Guo [14] and on its application to human activity recognition, in a way similar to the approach followed by Faria *et al.* [7].

2) Dynamic Features:

- **Velocities of joints coordinates**

$$v_{jd}^t = \left(p_{jd}^t - p_{jd}^{t-1}\right) \cdot f_r, \tag{6}$$

where f_r is the frame rate;

- **Projected angular velocities**

$$\omega_{id_p}^t = \left(\theta_{id_p}^t - \theta_{id_p}^{t-1}\right) \cdot f_r. \tag{7}$$

C. Feature Normalization

The features described above are combined along time to form the set of features matrix F', where each row corresponds to a feature vector and each column represents the variation of each feature along time. Next, the set of training and testing features matrices (F'_{tr} and F'_{te}, respectively) are normalized accordingly to:

$$f_{ij} = \frac{f'_{ij} - \min(F'_{\text{tr}\cdot j})}{\max(F'_{\text{tr}\cdot j}) - \min(F'_{\text{tr}\cdot j})}, \tag{8}$$

where f'_{ij} is the current value being normalized, f_{ij} is its respective value normalized and $F'_{\text{tr}\cdot j}$ refers to the column j of the matrix F'_{tr}. Two sets of normalized features are obtained: F_{tr} and F_{te} (training and testing sets, respectively). From this point on, these sets are generically referred as F.

D. Approach I - Max-Min Skeleton-based Features with Fixed-Size Window

Given a fixed-size window, used to observe a certain activity, the objective of this approach is to extract the maximum and minimum local values of considered features. The main reasons behind this approach are its low computational cost, since just maximum and minimum local values are needed to be computed, as well as the assumption that an activity can be discriminated just by considering the extreme movements (given by dynamic features) or poses (given by static features), since many activities are composed by repetitive sequences. The loss of temporal information is assumed as a possible limitation of this method.

Assuming that the examples contained in matrix F correspond to several activities, this matrix can be rewritten as:

$$F = \begin{bmatrix} F^1 & F^2 & \dots & F^a & \dots \end{bmatrix}^T, \tag{9}$$

where F^a is a sub-matrix that describes each activity a. Each of these sub-matrices are then sub-sampled into activity examples of n_{f_r} fixed-size number of frames (i.e. the window size)

$$F^a = \begin{bmatrix} F_1^a & F_2^a & \dots & F_n^a \end{bmatrix}^T. \tag{10}$$

In this first approach, only the following features are considered, based on experimental tests:

$$F_n^a = \begin{bmatrix} v_{jd}^t & \theta_{id_p}^t & \delta_{ab}^t \end{bmatrix}. \tag{11}$$

The size of each activity example matrix is $n_{f_r} \times 3((m - m_{\text{extd}}) + n_\theta + n_\delta)$, where m_{extd} is the number of joints not considered on the feature vector (e.g. *torso*) and n_θ and n_δ are the number of considered angles and distances between joints, respectively. From each activity example matrix F_n^a, a feature vector is constructed by computing the $n_{\max}$ maximum and $n_{\min}$ minimum values for each considered feature:

$$f_n^a = \begin{bmatrix} f_n^{\max} & f_n^{\min} & (l_{\text{cov}p})_n^a & (l_{\text{cov}\theta})_n^a \end{bmatrix}, \tag{12}$$

where $f_n^{\max} = \begin{bmatrix} v_{jd}^{\max} & \theta_{id_p}^{\max} & \delta_{ab}^{\max} \end{bmatrix}$ and $f_n^{\min} = \begin{bmatrix} v_{jd}^{\min} & \theta_{id_p}^{\min} & \delta_{ab}^{\min} \end{bmatrix}$. The length of this vector, with the additional features $l_{\text{cov}p}$ and $l_{\text{cov}\theta}$ associated to the example activity, which improved the overall results, is $[(n_{\max} + n_{\min}) \cdot (3((m - m_{\text{extd}}) + n_\theta + n_\delta)) + 2]$.

E. Approach II - Max-Min Skeleton-based Features and Key Poses

The first approach may be limited by the fact that the possible optimal fixed-size window may vary, depending on the activity. On the other hand, a certain activity may take different times to be executed in real-time. Therefore, it may not be possible to fix a window size to implement the first approach. The second approach aims to solve this issue, considering variable-size windows. The concept of *key poses*, based on the pose kinetic energy, was introduced in [6]. The key poses represent extreme points in the motion path of each joint, where most of the discriminative properties of each action are encoded, so they can be used to determine the size of the analysis window. In other words, instead of a fixed-size window, a window is determined by considering two consecutive key poses. A key pose is characterized by having a kinetic energy equal to zero. From [6], the pose kinetic energy is defined as

$$E^t = \frac{1}{2} \sum_{j=1}^{m} \sum_d \left(v_{jd}^t\right)^2, \tag{13}$$

where $d \in \{x, y, z\}$ and the key poses must satisfy

$$E^t < E_{\min}, \tag{14}$$

where $E_{\min}$ is a tuned threshold. Although the noise of the skeleton data was attenuated in the preprocessing stage, it is important to set an upper threshold E_{u} (i.e. hysteresis behavior) after a key pose is identified, so that another key pose may be determined, disregarding the possible noisy kinetic energy values in the neighborhood of the first. This guarantee that only a key pose is identified in a neighborhood, depending on the fact of the mentioned threshold is passed. As in the previous approach, based on experimental tests, matrices F_n^a describing activity examples are obtained, but this time

$$F_n^a = \begin{bmatrix} v_{jd}^t & \omega_{id_p}^t & \theta_{id_p}^t & \Delta_k^t \end{bmatrix}, \tag{15}$$

with size of $n_a \times [3((m - m_{extd}) + n_\theta + n_\Delta) + n_\omega]$, where n_a is the size of the respective window, n_Δ is the considered number of normals to triangles formed by three joints and n_ω is the considered number of projected angular velocities.

At this point, it is important to notice the distinction made previously about the calculated features: static and dynamic. Since key poses are determined on frames where the velocities of the corresponding joints are close to zero, it does not make sense to consider dynamic features there. Therefore, for the key poses only static features are extracted (i.e. θ and Δ). The max-min dynamic features (i.e. v and ω) are extracted, in the same way as explained in the first approach, but this time in the dynamic window defined by the key poses. In other words, from a given activity, information about static postures and dynamic movements is extracted, in between key poses, which delimit a variable sized window of an activity. Thus, an example vector of the form

$$f_n^a = \begin{bmatrix} f^1_{\text{static}} & f_{\text{dynamic}} & f^{n_a}_{\text{static}} \end{bmatrix} \tag{16}$$

is obtained, where $f^t_{\text{static}} = \begin{bmatrix} \theta^t_{id_p} & \Delta^t_k \end{bmatrix}$ and $f_{\text{dynamic}} = \begin{bmatrix} v^{\max}_{jd} & \omega^{\max}_{id_p} & v^{\min}_{jd} & \omega^{\min}_{id_p} \end{bmatrix}$. The length of this vector is $[2 \cdot 3(n_\theta + n_\Delta) + (n_{\max} + n_{\min}) \cdot (3(m - m_{extd}) + n_\omega)]$.

F. Training Model - Random Forest

Breiman [15] first introduced the random decision forest (RF), which may be viewed as an ensemble of decision-tree classifiers. It consists of two phases, where each tree is grown to the largest extent possible, without pruning, or until some defined maximum depth is reached:

1) **Bootstrap Phase**: randomly select a subset of features, from the training set, which will be used for growing a tree; the remaining features form the out-of-bag (OOB) set, which is used to estimate the OOB-error of the training set;
2) **Growing Phase**: using classification and regression tree (CART) [16], for each node to be divided, select one feature, from the randomly selected subset; the parameters of each node of every tree are optimized.

This process is done until a defined number of maximum trees. In the experimental results, a maximum of 100 trees was used. The number of selected features used to form the random subset is given by $\text{int}\left(\log_2(n_{\text{features}}) + 1\right)$.

IV. Experimental Results

In order to assess the proposed method, the Cornell Activity Dataset (CAD-60) [1] was used. The implementation of both developed approaches was done in Matlab and the Weka Version 3-6-13 software [17] provided the RF training algorithm, in a 2.60 GHz Intel Core i5 CPU machine.

A. Cornell Activity Dataset

The CAD-60 consists of 3D skeleton's coordinates joints, acquired by a RGB-D sensor at a frame rate of 30 Hz. Figure 2 exemplifies the skeleton's data provided, as well as the assumed coordinates system. Table I shows the index considered for each joint. The dataset contains 12 human distinct activities plus 1 random action and 1 still posture, categorized into 5 environments (bathroom, bedroom, kitchen, living room and office), performed by 4 different subjects. The considered joint angles are defined in Table II and the normal to the triangles formed by groups of three joints are described in Table III. The considered distances between joints are presented in table IV. These considered features aim to provide a good discrimination between activities. In this sense, for example, distances between adjacent joints are not considered (e.g. δ_{12}), since they are the same, independently of the activity.

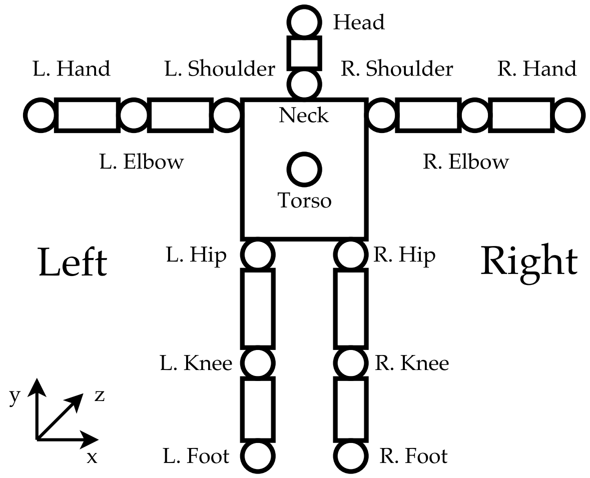

Fig. 2. Provided dataset: 3D skeleton data and respective coordinates system (after OpenNi [18]).

TABLE I
CAD-60 Dataset Joints

j	Joint	j	Joint	j	Joint
1	Head	2	Neck	3	Torso
4	L. Shoulder	5	L. Elbow	6	R. Shoulder
7	R. Elbow	8	L. Hip	9	L. Knee
10	R. Hip	11	R. Knee	12	L. Hand
13	R. Hand	14	L. Foot	15	R. Foot

TABLE II
Considered Angles

Angle	Joints Triplet	Angle	Joints Triplet
i	(a,b,c)	i	(a,b,c)
1	(6,7,13)	2	(4,5,12)
3	(6,10,11)	4	(4,8,9)
5	(10,11,15)	6	(8,9,14)
7	(6,10,13)	8	(4,8,12)
9	(1,7,13)	10	(1,5,12)
11	(3,12,13)	12	(3,14,15)

TABLE III
Considered Normal to Triangles Formed by Three Joints

Normal	Joints Triplet	Normal	Joints Triplet
k	(a,b,c)	k	(a,b,c)
1	(4,5,12)	2	(6,7,13)
3	(8,9,14)	4	(10,11,15)
5	(1,12,13)	6	(3,12,13)
7	(5,8,12)	8	(7,10,13)

The performance indicators in terms of Precision (Prec) and

TABLE IV
CONSIDERED DISTANCES BETWEEN JOINTS

Joints	Joints	Joints	Joints	Joints
(a,b)	(a,b)	(a,b)	(a,b)	(a,b)
(1,4)	(1,5)	(1,6)	(1,7)	(1,8)
(1,9)	(1,10)	(1,11)	(1,12)	(1,13)
(1,14)	(1,15)	(2,5)	(2,7)	(2,8)
(2,9)	(2,10)	(2,11)	(2,12)	(2,13)
(2,14)	(2,15)	(3,5)	(3,7)	(3,9)
(3,11)	(3,12)	(3,13)	(3,14)	(3,15)
(4,7)	(4,9)	(4,11)	(4,12)	(4,13)
(4,14)	(4,15)	(5,6)	(5,8)	(5,9)
(5,10)	(5,11)	(5,13)	(5,14)	(5,15)
(6,9)	(6,11)	(6,12)	(6,13)	(6,14)
(6,15)	(7,8)	(7,9)	(7,10)	(7,11)
(7,12)	(7,14)	(7,15)	(8,12)	(8,13)
(9,12)	(9,13)	(10,12)	(10,13)	(11,12)
(11,13)	(12,14)	(12,15)	(13,14)	(13,15)

Recall (Rec) are presented, for each scenario [19]. A leave-one-out cross validation procedure is employed: the model is trained by three of the four subjects and tested using the remaining one. This strategy enables the conclusion about the generalization capability of the classifier using the proposed set of features.

B. Results and Analysis - Approach I

The results obtained for the first approach are presented in Table V. The overall average for precision was 87.62% and for recall was 84.82%. The following parameters were used:

- $n_{f_r} = 120$; this means that, if an activity has more frames, it is divided roughly into examples of size n_{f_r};
- $m_{extd} = 3$; the velocities of the *head*, *neck* and *torso* were not used;
- $n_\theta = 12$; the considered angles are shown in Table II;
- $n_\delta = 70$; the considered distances between joints are presented in Table IV;
- $n_{\max} = n_{\min} = 1$; only the most extreme values for each feature on the analysis window per activity were considered.

These results show that with few examples to train the classifier (the average number of examples to train the RF classifier is 386,7), allied to just max-min skeleton-based features (each example has 566 features), it is possible to discriminate between distinct human activities, with a good confidence and very fast training (the average training time was 611 ms). Based on these characteristics, this approach could be suitable for real-time applications.

C. Results and Analysis - Approach II

The results obtained for the second approach are presented in Table VI. The overall average for precision was 81.73% and for recall was 79.01%. The following parameters were used:

- $m_{extd} = 3$; the velocities of the *head*, *neck* and *torso* were not used;
- $n_\theta = 12$; the considered angles are summarized in table II;
- $n_\omega = 12$; the considered angular velocities are obtained based on their respective angles;

TABLE V
PERFORMANCE OF THE APPROACH I ON THE CAD-60

Location	Activity	Prec (%)	Rec (%)
Bathroom	random+still	86.55	92.12
	rinsing water	91.67	80.05
	brushing teeth	98.92	95.25
	wearing lens	93.75	83.02
	average	92.72	87.61
Bedroom	random+still	93.20	99.00
	talking on phone	88.00	77.12
	drinking water	77.50	84.47
	opening container	71.67	60.95
	average	82.59	80.38
Kitchen	random+still	91.47	96.75
	drinking water	99.07	100
	chopping	91.97	96.65
	stirring	99.07	92.22
	opening container	72.05	65.65
	average	90.73	90.25
Living room	random+still	95.75	98.87
	talking on phone	73.40	55.75
	drinking water	75.80	75.65
	talking on couch	89.00	94.22
	relaxing on couch	70.00	71.87
	average	80.79	79.27
Office	random+still	93.25	97.75
	talking on phone	79.40	66.95
	writing on board	96.50	94.90
	drinking water	87.27	73.40
	working on computer	100	100
	average	91.28	86.6
	Overall Average	**87.62**	**84.82**

- $n_\Delta = 8$; the considered normal to triangles formed by three joints are presented in table III;
- $n_{\max} = n_{\min} = 1$; only the most extreme values for each feature on the dynamic window per activity were considered;
- $E_{\min} = 0.0028$; this value was tuned empirically, based on experimental tests on the training data;
- $E_u^a = 2 \times \text{mean}\left(E^a\right)$, where mean$(\cdot)$ is the mean function and $E^a = \{\left(E^1\right)^a, \left(E^2\right)^a, ..., \left(E^t\right)^a, ...\}$ is the set of kinetic energy values of the activity a; this value was also tuned empirically, based on experimental tests on the training data.

It is important to notice some practical constrains, which may contribute to reduce the overall performance of the approach:

- It calculates the global skeleton's kinetic energy, based on the velocities of **all** joints; this means that for a key pose to occur, every joints must have zero velocity; lets consider the *drinking water* activity (which revealed the worsts results): only the upper joints are of interest in this activity; however secondary motions can interfere in the computation of the actual key poses (e.g. noisy data; leg movements);
- The $E_{\min}$ value, which was obtained based on tests for all activities, should not be a **fixed threshold**, since there are activities with different motion patterns; this means that more examples were obtained for less dynamic activities (e.g. *relaxing on couch*), compared to more dynamic ones (e.g. *brushing teeth*).

TABLE VI
PERFORMANCE OF THE APPROACH II ON THE CAD-60

Location	Activity	Prec (%)	Rec (%)
Bathroom	random+still	95.87	96.60
	rinsing water	81.57	68.32
	brushing teeth	96.05	92.97
	wearing lens	84.05	85.35
	average	89.38	85.81
Bedroom	random+still	97.87	99.62
	talking on phone	56.05	66.50
	drinking water	57.15	36.85
	opening container	100	94.35
	average	77.77	74.33
Kitchen	random+still	93.00	98.47
	drinking water	99.00	95.82
	chopping	83.77	92.50
	stirring	73.07	64.80
	opening container	100	86.32
	average	89.77	87.58
Living room	random+still	96.70	99.62
	talking on phone	59.82	75.07
	drinking water	58.15	33.42
	talking on couch	81.25	85.72
	relaxing on couch	75.00	62.50
	average	74.18	71.27
Office	random+still	94.60	96.90
	talking on phone	49.72	71.02
	writing on board	92.17	90.87
	drinking water	51.32	21.55
	working on computer	100	100
	average	77.56	76.07
	Overall Average	**81.73**	**79.01**

Nevertheless, these problems inspire some new ideas for future research, which are discussed in the next section. The training of the classifier using this approach is also fast (the average training time is 0.92 s), requiring an average of 630.30 training examples, each with 264 features. Given these characteristics, this approach is also suitable for being used on real-time applications.

V. Conclusion and Future Work

The proposed approaches are based in max-min skeleton-based features. While in the first approach each example consists of a fixed-size window, in the second one, the considered window is delimited by key poses (no fixed-size window), determined by frames where the skeleton has zero kinetic energy. The second approach may suggest several new research directions:

- Instead of considering the global skeleton's kinetic energy, it may be possible to divide the human skeleton into several parts, calculate the kinetic energy for each part and then apply a similar method as approach II to those parts; this could be useful to differentiate parts that are more/less important to some activity;
- Train a specialized RF for each part, according to their correspondent features, having a higher-level classifier to discriminate between activities; it would also be interesting to assign different weights to each part, which could enable the detection of multiple activities at the same time (i.e. activities being performed by different body parts);
- Distinguish transitions between different activities, considering the computed key poses;
- Develop a parallel system to perceive the context of the activity (e.g. classification of used objects), allowing the extraction of context features.

Another interesting characteristic of the proposed method is its fast training, requiring few training examples. This could lead to a line of research, where the training algorithm could learn and incorporate new activities in real-time, conciliating all the previously mentioned directions of research.

References

[1] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Human Activity Detection from RGBD Images," *Plan, Activity, and Intent Recognition*, vol. 64, 2011.

[2] R. Gupta, A. Y.-S. Chia, and D. Rajan, "Human Activities Recognition using Depth Images," in *Proceedings of the 21st ACM international conference on Multimedia*. ACM, 2013, pp. 283–292.

[3] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time Human Pose Recognition in Parts from Single Depth Images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[4] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Learning Actionlet Ensemble for 3D Human Action Recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 5, pp. 914–927, 2014.

[5] X. Yang and Y. Tian, "Effective 3D Action Recognition using Eigenjoints," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 2–11, 2014.

[6] J. Shan and S. Akella, "3D Human Action Segmentation and Recognition using Pose Kinetic Energy," in *Advanced Robotics and its Social Impacts (ARSO), 2014 IEEE Workshop on*. IEEE, 2014, pp. 69–75.

[7] D. R. Faria, M. Vieira, C. Premebida, and U. Nunes, "Probabilistic Human Daily Activity Recognition Towards Robot-Assisted Living," in *Robot and Human Interactive Communication (RO-MAN), 2015 24th IEEE International Symposium on*. IEEE, 2015, pp. 582–587.

[8] S. Gaglio, G. L. Re, and M. Morana, "Human Activity Recognition Process using 3-D Posture Data," *Human-Machine Systems, IEEE Transactions on*, vol. 45, no. 5, pp. 586–597, 2015.

[9] G. I. Parisi, C. Weber, and S. Wermter, "Self-organizing Neural Integration of Pose-motion Features for Human Action Recognition," *Frontiers in Neurorobotics*, vol. 9, 2015.

[10] E. Cippitelli, S. Gasparrini, E. Gambi, and S. Spinsante, "A Human Activity Recognition System using Skeleton Data from RGBD Sensors," *Computational Intelligence and Neuroscience*, vol. 2016, 2016.

[11] G. Zhu, L. Zhang, P. Shen, and J. Song, "Human Action Recognition using Multi-layer Codebooks of Key Poses and Atomic Motions," *Signal Processing: Image Communication*, 2016.

[12] L. Gan and F. Chen, "Human Action Recognition using APJ3D and Random Forests," *Journal of Software*, vol. 8, no. 9, pp. 2238–2245, 2013.

[13] W. Liu, Y. Fan, T. Lei, and Z. Zhang, "Human Gesture Recognition using Orientation Segmentation Feature on Random Forest," in *Signal and Information Processing (ChinaSIP), 2014 IEEE China Summit & International Conference on*. IEEE, 2014, pp. 480–484.

[14] K. Guo, *Action Recognition using Log-Covariance Matrices of Silhouette and Optical-Flow Features*. Boston University, 2012.

[15] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[16] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. CRC press, 1984.

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: an Update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[18] "OpenNi SDK," Accessed: 2016-06-06. [Online]. Available: http://www.openni.org/

[19] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured Human Activity Detection from RGBD Images," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 842–849.