algorithms

MDPI

Article

Real-Time Arm Gesture Recognition Using 3D Skeleton Joint Data

Georgios Paraskevopoulos [1], Evaggelos Spyrou [2,3,*], Dimitrios Sgouropoulos [2], Theodoros Giannakopoulos [2] and Phivos Mylonas [4]

[1] School of Electrical and Computer Engineering, National Technical University of Athens, Iroon Polytechneiou 9, 157 73 Zografou, Greece; georgepar.91@gmail.com

[2] Institute of Informatics and Telecommunications, NCSR Demokritos, Neapoleos 10, 153 41 Ag. Paraskevi, Greece; dsgou@iit.demokritos.gr (D.S.); tyianak@iit.demokritos.gr (T.G.)

[3] Department of Computer Science and Telecommunications, University of Thessaly, 3rd km Old National Rd. Lamia-Athens, 351 00 Lamia, Greece

[4] Department of Informatics, Ionian University, Platia Tsirigoti 7, 491 00 Corfu, Greece; fmylonas@ionio.gr

* Correspondence: espyrou@iit.demokritos.gr; Tel.: +30-210-650-3175

Received: 29 March 2019; Accepted: 15 May 2019; Published: 20 May 2019

Abstract: In this paper we present an approach towards real-time hand gesture recognition using the Kinect sensor, investigating several machine learning techniques. We propose a novel approach for feature extraction, using measurements on joints of the extracted skeletons. The proposed features extract angles and displacements of skeleton joints, as the latter move into a 3D space. We define a set of gestures and construct a real-life data set. We train gesture classifiers under the assumptions that they shall be applied and evaluated to both known and unknown users. Experimental results with 11 classification approaches prove the effectiveness and the potential of our approach both with the proposed dataset and also compared to state-of-the-art research works.

Keywords: gesture recognition; Kinect; skeleton joints; machine learning

1. Introduction

Both poses and gestures may be used for communication, since they are able to convey some kind of meaning. Pose and gesture recognition aim to recognizing such meaningful expressions performed by a human. These expressions typically involve the motion of hands, arms, head, facial expressions and in some cases the whole body. Their recognition often aims to facilitate the interfacing between humans and computers. Current applications range from sign language, to gaming, medical applications and even virtual reality. Typically, the user stands in front of a camera and some parts of his/her body are detected and tracked. Features are then extracted and compared to models so as to be translated into some predefined gestures.

In this work we propose an arm gesture recognition approach which introduces a novel set of features. More specifically we use the Kinect sensor and its software development kit (SDK) to extract and track the skeletal joints. Using a subset of these joints, meaningful for the set of gestures that we aim to recognize, we extract their 3D coordinates. Then we calculate a set of statistical features on the aforementioned set of coordinates and for the whole set of video frames that depict the specific gesture. In the following, the term "gesture" shall refer to arm gestures that may involve hands, wrists and elbows. We investigate the use of some well-known machine learning algorithms on these features. Moreover, we adopt two real-life

scenarios. We consider the cases where such an approach may be used by (a) a limited set of known users and (b) an unlimited set of both known and unknown users. The former scenario corresponds to the case where all users of the system are a-priori known; i.e., the system shall be designed to work only on them. The latter scenario is the case where the system is pre-trained with a set of available users but also needs to work with previously unseen ones. To our belief, both cases may be involved in real-life scenarios, since they may fit real-life user requirements, thus both need to be addressed and evaluated.

We also present a novel dataset of gestures, used for the evaluation of the aforementioned algorithms and scenarios. This set of features has been defined in order to be semantically meaningful, to reflect small gesture changes to small feature changes and to require constant time (in terms of complexity). Moreover, we investigate whether the manual filtering of a dataset from "mischievous" users that fail (intentionally or not) to provide reliable gestures leads to significant improvement of the results while is worth the effort spent. Since our ultimate goal is to extract features and classify the corresponding gestures in real-time, using trained models, we perform an evaluation of our approach in common hardware architectures and more specifically in terms of the time required for the recognition of a gesture. We discuss which algorithms are suitable for adoption into real-life, real-time applications and the trade-offs between precision and responsiveness. Finally, we perform extensive comparisons with the state-of-the-art and discuss research works that share similar goals to those of this paper. We should clarify that our approach does not perform any temporal segmentation to detect the beginning and the ending of a possible gesture; instead we consider this problem as solved, i.e., we use pre-segmented videos and only aim to recognize gestures per segment. Note that each segment may contain at most a sole gesture.

The rest of this paper is organized as follows: Section 2 presents related research works in the area of gesture recognition using skeletal data. Section 3 presents the Kinect SDK and the proposed feature extraction approach. The dataset and the experimental results are presented in Section 4, where we also include discussion and comparisons to the state-of-the-art. Finally, in Section 5, we draw our conclusions and discuss plans for future work.

2. Related Work

During the last few years, many research efforts have focused on the problem of gesture recognition. In this section we present works that attempt to recognize similar gestures, to those that are recognized in this work and focus on those that include (but are not limited to) arm gestures.

Traditional machine learning approaches such as artificial neural networks (ANN), support vector machines (SVM), decision trees (DT) or K-nearest neighbor classifiers (KNN) have been widely used. Bhattacharya et al. [1] used an exemplar gesture to mark the beginning of the recognition process and avoid temporal segmentation. Their approach used SVMs and DTs on 3D skeletal joint coordinates. Lai et al. [2] selected the spine joint as reference, computed distances from elbows and hands and used them as features. For recognizing gestures, they used KNN classifier. Mangera et al. [3] used 3D joints calculated on extracted key-frames of sequences and cascades of ANNs to firstly classify the gesture side (left/right) and then recognize gestures, accordingly. Miranda et al. [4] used SVMs to recognize distinctive key poses. They modelled gestures as sequences of key poses and used decision forests to recognize them. Ting et al. [5] used 4D quaternions to describe joints and SVMs to recognize gestures.

Since in general, gestures are temporal sequences that vary in speed (e.g., when performed by different users), the dynamic time warping algorithm (DTW) [6] has been exploited in many works, as it is able to tackle precisely this problem. Celebi et al. [7] used dynamic time warping (DTW) on 3D points normalized per person. Reyes et al. [8] proposed a feature weighted variation of DTW on 3D joints, using the neck as a reference point. Ribó et al. [9] presented a similar approach which relies on DTW, KNN classifiers and

some heuristics to improve performance. Ibañez et al. [10] used 3D joints and compared DTW with an HMM, acquiring the same precision.

Hidden Markov models (HMM) and Gaussian Mixture Models (GMM) have also been frequently used, due to their ability to solve temporal pattern recognition problems. Anuj et al. [11] reduced the problem of 3D tracking to 2D, by projecting the tracked joints to a coordinate system attached to the user and aligned to his/her orientation. Their rule-based approach was based on HMM and adaptive thresholds. The technique proposed by Gonzalez–Sanchez and Puig [12] relied on background subtraction and head/hands detection. The 3D positions of hands and head were used with a GHMM. Gu et al. [13] used a HMM on clustered sets of 3D joints. Tran and Trivedi [14] detected head and hands and then trained GMMs by imposing kinematic constraints in the process. Finally, Yin and Davins [15] tracked hands and derived feature vectors that represented the hand shape based on HOG and motion features. Then they used a hierarchical model of HMMs for real-time recognition.

In Section 5 we summarize the aforementioned approaches, emphasizing on features used, learning algorithm(s) applied, gestures tackled and accuracy achieved on the data set adopted at each. As it will be seen, within this work we evaluate traditional machine learning algorithms, such as SVMs, KNN, HMMs, GMMs and tree models. However, we should herein mention that during the last few years, research efforts have shifted towards deep-learning based approaches. The latter are able to learn features, instead of computing them using an algorithm. Lin et al. [16] proposed the use of deep spatio–temporal features that are extracted by 3D convolutional neural networks (CNNs) and a model that was built with skeleton information and was based on a long short term memory network (LSTM). Wang and Wang [17] used two-stream recurrent neural networks, in order to model the temporal dynamics and the spatial configurations of actions. Mathe et al. [18] used a CNN on raw skeletal data, while Zhang et al. [19] used 3D CNNs and convolutional LSTMs. However, as we shall later discuss in Section 5, deep learning approaches typically lead to more complex models that require more computational resources when compared to the simpler models that result from feature-based approaches such as the one proposed in this work.

3. Arm Gesture Recognition

It is well-known that both poses and gestures may be used to communicate meaning. We may define a pose as the temporary suspension of movement, where body joints assume a distinct configuration in space. On the other hand, a gesture may have a more complex definition. According to the Merriam–Webster dictionary (https://www.merriam-webster.com/dictionary/gesture), a gesture may be either defined as "a movement usually of the body or limbs that expresses or emphasizes an idea, sentiment, or attitude," or "the use of motions of the limbs or body as a means of expression." Within the scientific literature, there exist several definitions of a gesture. In the fields of computer vision and pattern recognition, a generic definition of a gesture by Mitra and Acharya [20] has been formed as: "Gestures are expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head, face, or body with the intent of: (1) conveying meaningful information or (2) interacting with the environment." In the context of this work we focus on a subset of gestures, which may be referred to as "arm gestures" [21]. More specifically, we may define an arm gesture as the transfer of a (sub)set of arm joints (i.e., hands, wrists, elbows) from point A to point B, using a vaguely predefined trajectory.

3.1. The Microsoft Kinect SDK

In order to extract the skeletal data, we use the Microsoft Kinect [22] camera, which provides real time RGB, depth and skeletal tracking information and is based on the work of Shotton et al. [23], providing a robust algorithm for every person, in any background, without the need of a calibration step. A human

is described by a structured graph of joints, representing its main body parts (e.g., arms, legs, head, shoulders etc.). For each joint, the corresponding 3D coordinates are extracted in real time. In Figure 1 we depict the human skeleton joints, as they are extracted with the use of the Kinect SDK. We should note that in this work we are only interested in the left and right elbows, hands, and wrists. As we have already discussed, these data are comprised of the id and the 3D coordinates of each joint. We should note that joints are organized in a hierarchical structure, where a parent–child relationship is implied; e.g., the root is the hip center, its children are the spine, the left heap and the right heap and so on.

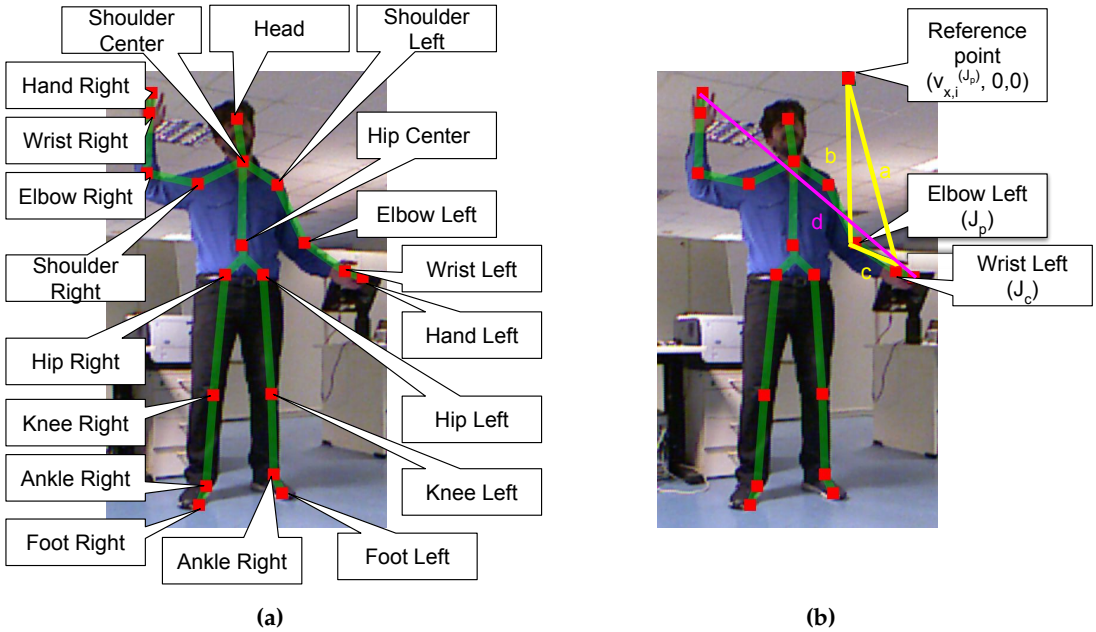

Figure 1. (**a**) Extracted human skeleton 3D joints using the Kinect software development kit (SDK). (**b**) Visual representation of a given node J, its parent J_p and its child J_c; a, b, c, d, used in Equations (2)–(6) and the reference point $(v_{x,i}^{J_p}, 0, 0)$.

3.2. Gesture Recognition

The proposed set of features, is partially inspired to those proposed by Sheng [24], whose work extended the one of Rubine [25] from the 2D to the 3D space. More specifically, let J denote a given joint. Then, by J_c and J_p we denote its child and parent joints, accordingly. Also let F_i, $i = 1, 2, \ldots, N$ denote a given video frame and $\mathbf{v}_i^{(J)} = \left(v_{x,i}^{(J)}, v_{y,i}^{(J)}, v_{z,i}^{(J)}\right)$ a vector corresponding to the 3D coordinates of J at frame F_i. Also, let $\mathcal{V}^{(\mathcal{J})}$ be the set of all $\mathbf{v}_i^{(J)}$. By $B(\mathcal{V}^{(\mathcal{J})})$ we denote the 3D bounding box of $\mathcal{V}^{(\mathcal{J})}$, by $a_{B(\mathcal{V}^{(\mathcal{J})})}$ and $b_{B(\mathcal{V}^{(\mathcal{J})})}$ the two different lengths of its sides. The extracted features are depicted in Table 1. Note that each feature is calculated for a given joint J and may involve a different subset of frames. The notation is summarized in Table 2.

More specifically, we extracted the spatial angle between the first two frames (i.e., F_1 and F_2), the last two frames (i.e., F_{N-1} and F_N) and the first and the last frame (i.e., F_1 and F_N), in all cases using the inner product of the corresponding vectors. Moreover, we extracted the total and the squared total vector angles, between F_1 and F_N using the inner product of the corresponding vectors, i.e., $\mathbf{v}_1^{(J)}$ and $\mathbf{v}_N^{(J)}$. The total vector displacement is the magnitude of the difference of the vectors corresponding to the initial and the final positions of J, while the total displacement is the sum of the magnitudes of all differences between

the positions of J within any two given consecutive frames. Similarly, the maximum displacement is the magnitude of the maximum difference between the positions of J within any two given frames. Finally, we built a 3D bounding box for the set of all positions of the joint within all frames and estimate accordingly the length and the angle of its diagonal. Note that from the skeletal data we used only a subset of joints (i.e., their 3D coordinates). Also, note that, $J \in \{$ ElbowLeft, ElbowRight, HandLeft, HandRight, WristLeft, WristRight$\}$. Features marked with $*$, are calculated using only HandLeft and/or HandRight.

Table 1. Proposed features, extracted from the skeletal joints (features marked with $*$, are calculated using only HandLeft and/or HandRight).

Feature Name	Frames Involved	Equation
Spatial angle	F_2, F_1	$\arccos \frac{\mathbf{v}_2^{(J)} \cdot \mathbf{v}_1^{(J)}}{\left\|\mathbf{v}_2^{(J)}\right\| \cdot \left\|\mathbf{v}_1^{(J)}\right\|}$
Spatial angle	F_N, F_{N-1}	$\arccos \frac{\mathbf{v}_N^{(J)} \cdot \mathbf{v}_{N-1}^{(J)}}{\left\|\mathbf{v}_N^{(J)}\right\| \cdot \left\|\mathbf{v}_{N-1}^{(J)}\right\|}$
Spatial angle	F_N, F_1	$\arccos \frac{\mathbf{v}_N^{(J)} \cdot \mathbf{v}_1^{(J)}}{\left\|\mathbf{v}_N^{(J)}\right\| \cdot \left\|\mathbf{v}_1^{(J)}\right\|}$
Total vector angle	$F_1, \ldots, F_N$	$\sum_{i=1}^{N} \arccos \left(\frac{\mathbf{v}_i^{(J)} \cdot \mathbf{v}_{i-1}^{(J)}}{\left\|\mathbf{v}_i^{(J)}\right\| \left\|\mathbf{v}_{i-1}^{(J)}\right\|} \right)$
Squared total vector angle	$F_1, \ldots, F_N$	$\sum_{i=1}^{n} \arccos \left(\frac{\mathbf{v}_i^{(J)} \cdot \mathbf{v}_{i-1}^{(J)}}{\left\|\mathbf{v}_i^{(J)}\right\| \left\|\mathbf{v}_{i-1}^{(J)}\right\|} \right)^2$
Total vector displacement	F_N, F_1	$\left\|\mathbf{v}_N^{(J)} - \mathbf{v}_1^{(J)}\right\|$
Total displacement	$F_1, \ldots, F_N$	$\sum_{i=1}^{n} \left\|\mathbf{v}_i^{(J)} - \mathbf{v}_{i-1}^{(J)}\right\|$
Maximum displacement	$F_1, \ldots, F_N$	$\max_{i=2,\ldots,N} \left(\left\|\mathbf{v}_i^{(J)} - \mathbf{v}_{i-1}^{(J)}\right\| \right)$
Bounding box diagonal length *	$F_1, \ldots, F_N$	$\sqrt{a^2_{B(\mathcal{V}^{(\mathcal{J})})} + b^2_{B(\mathcal{V}^{(\mathcal{J})})}}$
Bounding box angle *	$F_1, \ldots, F_N$	$\arctan \frac{b_{B(\mathcal{V}^{(\mathcal{J})})}}{a_{B(\mathcal{V}^{(\mathcal{J})})}}$

Table 2. Symbols used throughout this paper and their description.

Symbol	Definition
J	a given joint
J_c, J_p	child/parent joint of J, respectively
F_i	a given video frame, $i = 1, \ldots, N$
$\mathbf{v}_i^J$	vector of 3D coordinates of J at F_i
$v_{x,i}^{(J)}, v_{y,i}^{(J)}, v_{z,i}^{(J)}$	the 3D coordinates of $\mathbf{v}_i^J$
$\mathcal{J}$	the set if all joints
$\mathcal{V}^{\mathcal{J}}$	the set of all vectors $\mathbf{v}_i^J, J \in \mathcal{J}, i = 1, 2, \ldots, N$
$B(\bullet)$	a 3D bounding box of a set of vectors
$a_{B(\bullet)}, b_{B(\bullet)}$	the lengths of the sides of $B(\bullet)$

Also between two parent and child joints i.e., $\{J_p, J\}$, (or equivalently $\{J, J_c\}$) we calculate the initial and final (i.e., at F_1 and F_N, respectively), mean and maximum angle (i.e., for $F_1, \ldots, F_N$). The angle θ_{pc} used in these calculations is

$$\theta_{pc} = \cos^{-1}\left(\frac{a_{pc}^2 + b_{pc}^2 - c_{pc}^2}{2a_{pc}b_{pc}}\right), \tag{1}$$

where

$$a_{pc}^2 = \left(v_x^{(J)} - v_x^{(J_c)}\right)^2 + \left(v_y^{(J)} - v_y^{(J_c)}\right)^2, \tag{2}$$

$$b_{pc} = v_x^{(J)} \text{ and} \tag{3}$$

$$c_{pc}^2 = \left(v_x^{(J_p)}\right)^2 + \left(v_y^{(J)} - v_y^{(J_p)}\right)^2. \tag{4}$$

Note that for the formulation of the triangle we use a reference point with coordinates $(v_{x,i}^{J_p}, 0, 0)$ (any given point may be used). A visual representation of this triangle is illustrated in Figure 1b, where a_{pc}, b_{pc} and c_{pc} have been drawn. Finally, between HandLeft (HL) and HandRight (HR) we extract the max

$$d_{\max} = \max_{i,j}\left\{d\left(\mathbf{v}_i^{\mathrm{HR}}, \mathbf{v}_j^{\mathrm{HL}}\right)\right\}, \tag{5}$$

and the mean distance,

$$d_{\mathrm{mean}} = \frac{1}{F^{(J)}}\sum_{i,j} d\left(\mathbf{v}_i^{\mathrm{HR}}, \mathbf{v}_j^{\mathrm{HL}}\right), \tag{6}$$

within the gesture. By d we denote the Euclidean distance and $F^{(J)}$ is the number of frames for each gesture, which is also used as a feature. The distance d between HL and HR is illustrated in Figure 1b.

According to [24] the extracted features should be defined so as to satisfy the following criteria: (a) they should require constant time; (b) small gesture changes should be reflected to small changes to the extracted features; and (c) features should be semantically meaningful. As for (a), the aforementioned extracted features allow for very fast feature extraction, because the calculations require constant time (i.e., $O(1)$) updates per frame. Criteria (b) and (c) are met because we chose to use angle and displacement dependent features, able to discriminate between slight joint movements. In addition they comprise a good semantic representation of the way joints move to perform a gesture i.e., as a set of interconnected, yet discrete points.

4. Experimental Results

4.1. Dataset

For the sake of the evaluation of the proposed method we constructed a real-life dataset of 10 users (seven male and three female), all between 22 and 36 years old of several heights ranging from 155 cm to 190 cm. We defined a set of "swipes" (swipe up/down/in/out) for both hands, thus resulting to eight different arm gestures. In the "swipe-up" gesture, the user starts with both arms close to the hips. Then, one of the arms is moving vertically, with the palm facing up, until it reaches the head and then returns back. In the "swipe-in" gesture, the user starts with both arms closed to the hips. Then one of the arms is moving vertically, with the palm facing up until it reaches approx. the middle of the shoulder center and the hip center. Then, it turns towards the direction of the other hand and performs a horizontal movement, with the palm facing it, until it is aligned with the imaginary line joining the shoulder center and the hip center. Then, with a diagonal movement, it returns to its original position. In Figure 2 we illustrate a sequence of RGB and skeletal frames for "swipe-in" and "swipe-up" gestures, performed by the same user,

with his right hand. "swipe-out" and "swipe-down" and also left hand gestures may be easily conceived, since they are symmetric to the aforementioned. Note that the orientation of the palm does not matter, however, users were instructed to perform the aforementioned changes, so that the produced gestures would be more "natural."

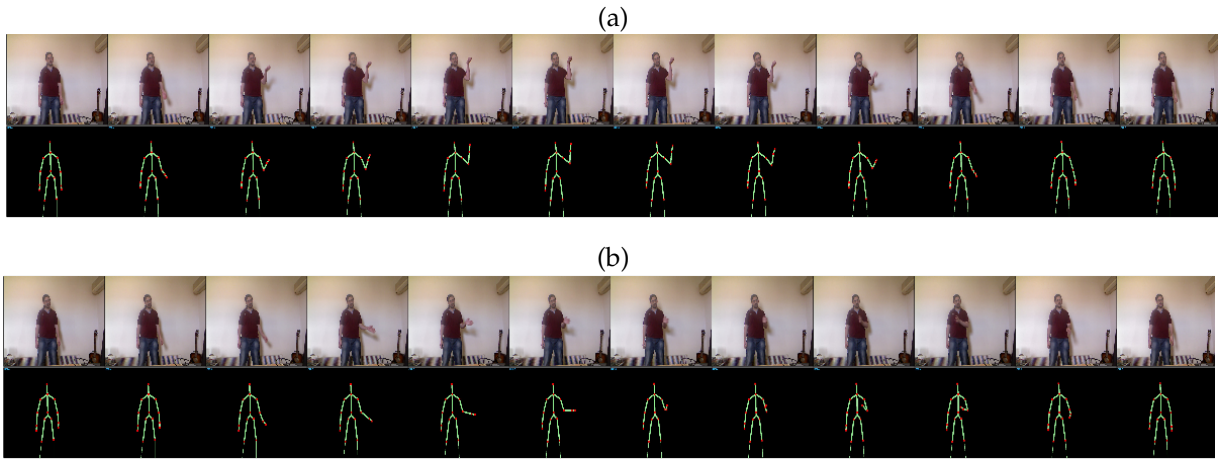

Figure 2. Aligned RGB and skeletal images of (**a**) swipe-up and (**b**) swipe-in gestures, performed by the same user.

Users were sitting in front of the Kinect; their distance was approx. 2 m to ensure that the skeleton would be extracted. From each sequence corresponding to a gesture we extracted the RGB-D raw data and also the skeletal data. Note that RGB-D data have been used only for visualization purposes, since the proposed approach relied solely on the skeletal data. We first demonstrated these gestures to the users so that they would become familiar before the gesture collection process. Within this process, we asked all users to perform each gesture at least 10 times and then we manually cleaned the data set by letting users to decide to "drop" some of their performed gestures, when they felt that it was wrongly performed. One of the users was "mischievous"; he was asked to introduce errors in the dataset generation process, since we wanted to observe the effect of those errors. More than 50% of his gestures were intentionally performed far from the correct way that he was instructed (e.g., incomplete, very fast/slow, wrongly performed etc.). This process resulted to a total of 820 gestures with approx 7.5% being a result of the mischievous behaviour of the aforementioned user. Note that none of the users meant to harm the experiment, apart from the mischievous one, who was explicitly instruct to behave this way.

To deal with "noisy patterns", before/after each gesture and to avoid a costly temporal segmentation step, users were equipped with a toggle button, which when pressed initialized/ended the recording of a gesture, accordingly. We let each user familiarize with this procedure, by testing it for 5–10 times. This way we were able to isolate gestures from other movements. For each gesture, we recorded the relevant skeletal data and extracted the sets of features described in Section 3.2 for all joints involved in these gestures, i.e., elbows, wrists and hands.

4.2. Experiments

We have experimented with a set of widely used classifiers, deriving from the Scikit-learn toolbox [26]: (a) support vector machines [27], both linear (LSVM) and non-linear (RBFSVM, i.e., with the radial basis function kernel); (b) k-nearest neighbor (KNN) classification [28]; (c) naïve Bayes (NB) [29]; (d) quadratic discriminant analysis (QDA) [30]; (e) linear discriminant analysis (LDA) [30]; (f) decision trees (DT) [31]; (g) random forest (RF) classifiers [32]; (h) extremely randomized trees (ET) [33] (which are ensembles that

are trained using a random threshold for each feature and while tend to produce more complex trees, they require smaller training time); (i) AdaBoost [34] with DT (ABDT) and ET (ABDT) classifiers.

For the experimental evaluation of the proposed methodology, we performed two series of extensive experiments. The goal of the first experiment was to investigate the performance for a known set of users. In other words, we applied the learning approach to a "closed" set of users, thus we were able to use training and testing sample gestures from all users (of course a sample gesture may not appear in both sets). The goal of the second experiment was to investigate the performance for an unknown set of users, i.e., training and testing sets were comprised from sample gestures of different users. This way, we aimed to study the generalization of our approach. The experiments were performed using the scikit-learn API [26].

Within each experiment, classifiers were trained using an optimal parameter value set, which was determined using cross-validation. As for the first experiment, we evaluated the aforementioned algorithms with a stratified K-fold cross validation procedure ($K \in \{3, 5, 7, 10, 13, 15, 17, 20\}$). The corresponding results are depicted in Figure 3, while the optimal parameter sets in Table 3. We may observe that best performance was achieved with the ET classifier followed by the RF. This indicates that an ensemble Tree Bagging approach is optimal for separating samples in the feature space constructed by our features. Nevertheless the rest of the classifiers still give adequate performance, which gives us the flexibility of tradeoffs (e.g., classification accuracy for computational complexity).

For our second experiment, we selected the most accurate approach of the first one, i.e., the ET. We experimented with training sets consisting of 1 to 9 users, while the samples of the rest were kept for the construction of the test set. Since we did not have a good heuristic to split the users in train and test sets, we considered every possible combination and calculated the mean accuracy over every combination with a given number of users in the training set. This experiment has two goals: (a) to estimate the generalization capability of our approach to unseen users; and (b) to investigate which is the minimum number of users so as to get usable results in real-life applications.

We begun this series of experiments with the leave one (user) out case. For this experiment we retained the samples of a user for the test set while training the algorithm on the samples of the rest of the users. In Table 4 we see the F_1 score of the gesture classification for every unknown user. We list the F_1 score for each gesture in detail and the average over the gestures.

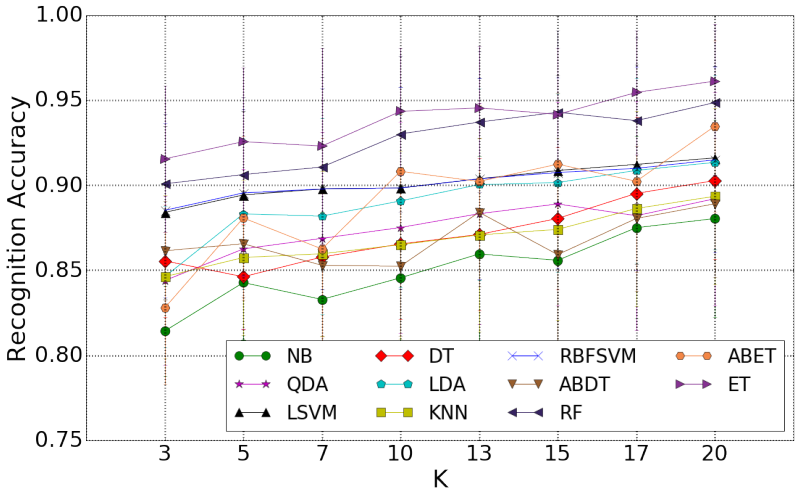

Figure 3. *K*-fold cross validation results for all machine learning approach and for several values of *K*.

Table 3. Optimal classifier parameters (α: learning rate; n: number of neighbors; e: number of estimators; s: search algorithm; d: max depth; m: metric between point p and q; f: max number of features; r: regularization parameter).

Classifier	Parameters
ABDT	$e = 103, \alpha = 621.6$
ABET	$e = 82, \alpha = 241.6$
DT	$d = 48, f = 49$
ET	$d = 17, f = 70, e = 70$
KNN	$n = 22, s = kd\_tree, m = \sum_{i=1}^{n}(\lvert p_i - q_i \rvert)$
LSVM	$C = 0.0091$
QDA	$r = 0.88889$
RBFSVM	$C = 44.445, \gamma = 0.0001$
RF	$d = 27, f = 20, e = 75$

Table 4. F_1 score for each gesture separately and mean F_1 score for all gestures for leave one (user) out experiment.

	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10
LH-SwipeDown	0.76	0.83	1.00	0.82	1.00	0.80	1.00	1.00	1.00	0.96
LH-SwipeIn	0.38	0.92	0.84	1.00	1.00	0.92	1.00	1.00	1.00	1.00
LH-SwipeOut	0.61	0.93	0.86	1.00	1.00	0.89	1.00	1.00	0.97	1.00
LH-SwipeUp	0.69	0.90	1.00	0.84	1.00	0.83	1.00	1.00	0.97	0.96
RH-SwipeDown	0.78	1.00	0.95	-	1.00	1.00	0.92	1.00	0.87	1.00
RH-SwipeIn	0.64	1.00	0.67	-	1.00	1.00	1.00	1.00	0.89	0.96
RH-SwipeOut	0.61	1.00	0.80	-	1.00	1.00	0.95	1.00	1.00	0.95
RH-SwipeUp	0.40	1.00	0.95	-		1.00	1.00	1.00	0.96	1.00
Average	**0.62**	**0.94**	**0.88**	**0.92**	**1.00**	**0.92**	**0.99**	**1.00**	**0.96**	**0.97**

For the next experiment we calculated the mean classification accuracy over the number of users in the training set. In Table 4 we may observe that as expected, the mischievous user had considerably lower classification scores, and we repeated the experiment with this user completely filtered out from the dataset. The results of both experiments are compared in Figure 4. As expected the overall accuracy of the system without the mischievous user was considerably higher, but as we included more users the difference between the performances of the aforementioned case, it decreased. It is important to note that we were able to train a classifier with adequate performance using only 3–4 users. In Figures 5 and 6 we plot the mean classification accuracy for each gesture separately as a function of the number of users used for training with and without the mischievous user, respectively. This information can be used to design more effectively a natural user interface (NUI) e.g., by assigning gestures with better recognition rate to most common operations.

Finally, since the ultimate goal of the presented approach was to recognize gestures in real time, we performed a series of experiments with different architectures, considering two cases where the classifier may run (a) on powerful desktop processor and (b) on a system with limited resources, like e.g., a Raspberry Pi. We constructed a benchmark where 80% of the dataset is used as training data and 20% as test data. We performed classification for every sample in the test data 1000 times and calculated the average time each classifier needed for a sample. Results are illustrated in Figure 7. We may observe that the most accurate classifiers, i.e., ET and RF, were considerably slower than the rest with approx. 100 ms classification time on a desktop system and 300 ms on a Raspberry Pi 2. However, we believe that although

these classification times are slower that the rest of the algorithms, the ET classifier can still be used in a real time scenario without having an impact on the user experience.

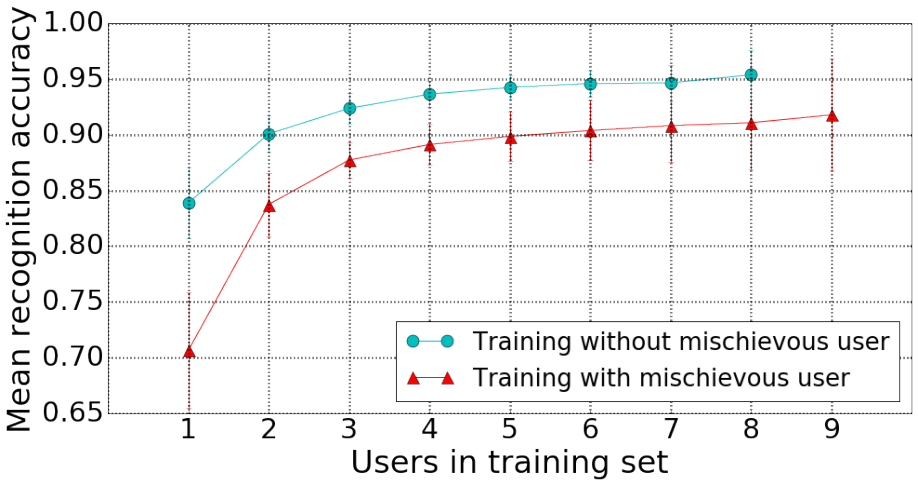

Figure 4. Mean accuracy vs. number of users in training set, with/without the mischievous user.

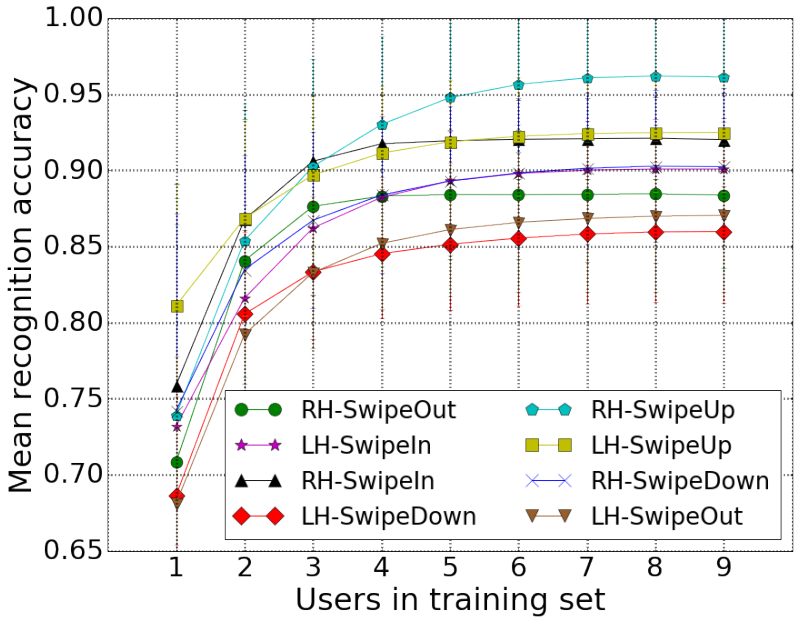

Figure 5. Mean accuracy per gesture vs. number of users in training set, with the mischievous user.

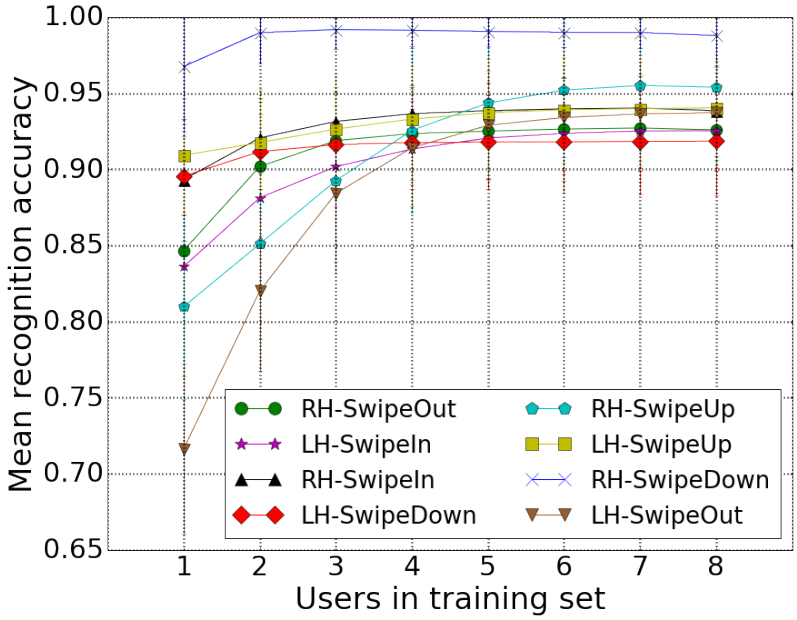

Figure 6. Mean accuracy per gesture vs. number of users in training set, without the mischievous user.

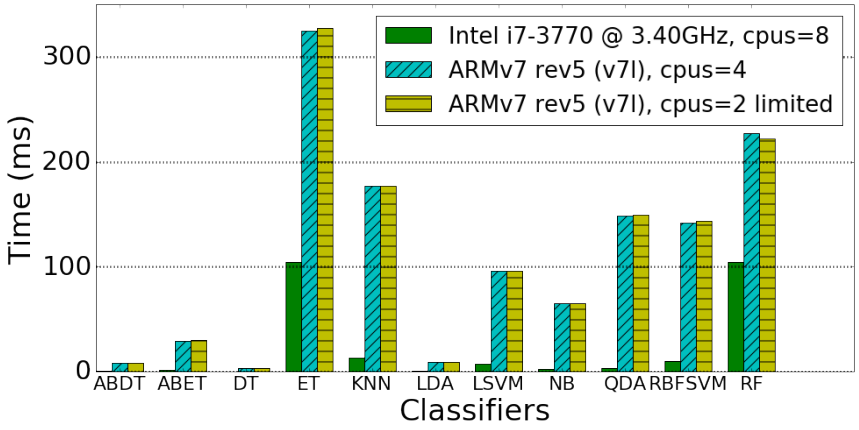

Figure 7. Comparison of average time required for the classification of a sample in several architectures.

4.3. Comparisons to the State-of-the-Art

In order to assess the efficiency of the proposed method, we performed a comparison using the MSR action dataset of [35]. This dataset consists of 20 gestures, performed by 10 subjects, each performing each action 2–3 times and captured with a depth sensor similar to the Kinect. We compare against the work of Li et al. [35], which are the authors of the dataset. We also compare against the aforementioned work of Miranda et al. [4], which only reported results on Test III. Results are depicted in Table 5. Based on the proposed evaluation procedure, three tests are performed, namely tests I, II and III, on three different datasets, namely AS1, AS2 and AS3. In tests I and II samples from all subjects are used both in training and testing (1/3–2/3 and 1/2–1/2 splits, respectively), while in test III half subjects are used for training and the remaining for testing. We may observe that our method performs very well using the AS3 dataset, has satisfactory results using AS1 while its performance drops when using AS2. This is due to the complexity of gestures involved in each dataset. AS3 is composed of simpler gestures that are closer to the ones we use in our dataset, while AS2 consists of more complex and nuanced gestures

that cannot be consistently and accurately recognized with only the use of aggregate features. Moreover, AS1 lies in the middle. We must also note that our method performs well on Test III, which evaluates performance on unseen users, which is a major issue when designing a NUI. Experiments indicate that our method outperforms the one of Li et al. [35] in both AS1 and AS3 and on average. However, the method of Li et al. achieved better results in tests I and II, mostly due the poorest performance of our approach in AS2. Of course, since in test III evaluation considers only unseen subjects, we feel that the superiority of our method therein compared to both [35] and also [4] is the most remarkable result of the evaluation with the MSR dataset.

We also performed a comparison to the publicly available dataset of [7], which has also been used for the evaluation of [3,9]. This dataset contains gestures similar to those of our dataset, which are also a lot simpler when compared to those of the MSR dataset. Corresponding results are depicted in Table 6. As it may be seen, our approach showed excellent performance, by recognizing all examples. Finally in Table 7 we compare our system to the results of [36]. This dataset consisted of gestures similar to ours and it was split in three subsets: RotationDB, RelaxedDB and Rotation/RelaxedDB. For the first subset, users inserted rotational distortion with their positioning in respect to the camera. For the second subset, users performed the gestures in a more relaxed way inserting movement of more joints than a gesture needs (for example scratching ones head while performing a swipe). The third subset contained both types of distortion. We may observe that our system outperforms the state-of-the-art approach of [36] for both types of injected distortions. Finally, we performed a comparison using our own dataset and the deep learning methodology of Mathe et al. [18] that uses a Convolutional Neural Network (CNN). Results for this case are provided in Table 8. As it may be seen, the proposed feature-based approach showed accuracy which was slightly increased compared to the CNN. Note, that in every case, the evaluation protocol of the dataset used was followed, so as to ensure fair comparisons among research works.

Table 5. Comparisons to state-of-the-art research works using the MSR action dataset of [35]. Results denote accuracy (%).

	Test I		**Test II**		**Test III**			**Avg.**	
	[35]	**Our**	[35]	**Our**	[35]	**Our**	[4]	[35]	**Our**
AS1	89.50	85.36	93.30	91.39	72.90	89.28	93.50	85.23	88.68
AS2	89.00	72.90	92.90	84.40	71.90	73.20	52.00	84.60	76.84
AS3	96.30	93.69	96.30	98.81	79.20	97.47	95.40	90.60	96.66
Avg.	**91.60**	83.98	**94.17**	91.53	74.67	**86.65**	80.30	84.24	**87.39**

Table 6. Comparison to state-of-the-art research works using the dataset of [7].

	[7]	[3]	[9]	**Our**
Acc. (%)	96.7	95.6	89.4	**100.0**

Table 7. Comparison to the state-of-the-art research work of [36].

	RotationDB		**RelaxedDB**		**Rotation/RelaxedDB**	
	[36]	**Our**	[36]	**Our**	[36]	**Our**
Acc. (%)	n/a	**98.01**	97.13	**98.74**	96.64	**97.98**

Table 8. Comparison to state-of-the-art research work of [18], using our own dataset.

	[18]	**Our**
Acc. (%)	91.0	**96.0**

5. Conclusions and Discussion

In this paper we have presented an approach that aims to recognize a set of simple gestures in real-time. We proposed a novel set of features extracted from the 3D positions of skeletal joints and evaluated them using several well-known machine learning approaches. We also introduced a dataset comprising of RGB, depth and skeletal data for several gestures.

Our approach was evaluated based on the assumptions that a gesture-driven NUI may be used by (a) a limited set of known users, i.e., both training and testing sets will be built using gesture examples from all users; and (b) unknown (previously unseen) users, i.e., training and testing sets will not have common users. The experimental results indicated that our approach, when using ET for classification, was able to achieve more than satisfactory results, outperforming state-of-the-art methods. Moreover, we investigated the effect of mischievous users, i.e., users which (intentionally or not) fail to successfully imitate the exemplar gestures. We showed that upon the sanitation of the dataset from such users, we were able to achieve significant accuracy improvement. Finally, we performed a comparison of prediction times of trained classifiers under different CPU architectures.

In Table 9 we summarize all related work of Section 2. As it can be observed, the majority of included research works do not propose a set of features on joints; rather they use joint coordinates and in many cases heuristics, which limit their capability to expand to other gestures. Also, performed evaluation in most cases is poor, most times limited to 3–5 subjects. High accuracy rates are achieved using custom and not publicly available datasets. On the other hand we proposed a scalable approach which uses features, does not rely on heuristics and is evaluated using a significantly larger set of users.

Future work will focus on expanding the set of gestures, by adding more complex ones and more samples of the existing ones. We also aim to investigate more machine learning approaches and extend our data set with more users. We also aim to use the approach in the context of the smart meeting room [37] developed within the IoT ecosystem of [38]. Therein it may act as an alternative means of controlling devices in the room, e.g., HVAC, lights, projector etc., and will be transformed into an exposed web service. Of course, in such a scenario, there will be the need for temporal segmentation or continuous recognition of performed arm gestures. Such a segmentation may be performed e.g., by finding the statit position of the actor [39], by applying sliding window volumes comprising of consecutive video frames [40], or even in a totally unsupervised manner by using clustering and keyframe extraction [41].

We should note that one way to avoid the aforementioned computationally expensive task of temporal segmentation, the initialization of the gesture recognition may be triggered e.g., with a predefined pose, which shall also act as a means for the selection of the device to be controlled. We plan to perform both quantitative and qualitative evaluations, i.e., also assess user experience when using such an approach within a smart environment. Finally, upon the completion of the aforementioned functionalities we plan an evaluation by real-life users. Among the issues we plan to investigate is the effectiveness of the proposed approach with users that cannot successfully perform (some of) the predefined gestures, i.e., people with limitations, disabilities or diseases such as Parkinson's.

Table 9. Reported results of the state-of-the-art (alphabetically). The 2nd column (Approach) summarizes features and learning approach used. In the 3rd column (Gestures), relevant gestures to those of our work are in bold. The 4th column (Acc.(s.)) presents the best accuracy achieved in the used dataset and the number of subjects in parentheses.

Ref.	Approach	Gestures	Acc.(s.)	Comments/Drawbacks
[11]	2D projected joint trajectories, rules and HMMs	**Swipe(L,R)**, Circle, **Hand raise**, Push	95.4 (5)	heuristic, not scalable rules, different features for different kinds of moves
[1]	3D joints, rules, SVM/DT	Neutral, T-shape, T-shape tilt/**pointing**(L,R)	95.0 (3)	uses an exemplar gesture to avoid segmentation
[7]	Norm. 3D joints, Weig. DTW	Push Up, Pull Down, **Swipe**	96.7 (n/a)	very limited evaluation
[12]	Head/hands detection, GHMM	**Up**/**Down**/**Left**/Stretch(L, R, B), Fold(B)	98.0 (n/a)	relies on head/hands detection
[13]	clustered joints, HMM	Come, Go, Sit, Rise, Wave(L)	85.0 (2)	very limited evaluation, fails at higher speeds
[10]	HMM, DTW	Circle, Elongation, Punch, Swim, **Swipe(L, R)**, Smash	96.0 (4)	limited evaluation
[2]	Differences to reference joint, KNN	**Swipe**(**L**, **R**, B), Push(L, R, B), Clapping in/out	97.2 (20)	sensitive to temporal misalignments
[3]	3D joints, velocities, ANN	S**wipe(L,R)**, **Push Up(L,R)**, **Pull Down(L,R)**, Wave(L,R)	95.6 (n/a)	not scalable for gestures that use both hands
[4]	Pose sequences, Decision Forests	Open Arms, **Turn Next/Prev. Page**, **Raise/Lower Right Arm**, Good Bye, Jap. Greeting, **Put Hands Up Front/Laterally**	91.5 (10)	pose modeling requires extra effort, limited to gestures composed of distinctive key poses
[8]	3D joints, feature weighted DTW	Jumping, Bending, Clapping, **Greeting**, Noting	68.0 (10)	detected begin/end of gestures
[9]	3D joints, DTW and KNN	**Swipe(L, R)**, **Push Up(L, R)**, **Pull Down(L, R)**, Wave(L, R)	89.4 (n/a)	relies on heuristically determined parameters
[5]	4D quaternions, SVM	**Swipe(L, R)**, Clap, Waving, Draw circle/tick	98.9 (5)	limited evaluation
[14]	Head/hands detection, kinematic constraints, GMM	Punch(L, R), Clap, Wave(L, R), Dumbell Curls(L, R)	93.1 (5)	relies on head/hands detection
[15]	Motion and HOG features of hands, hierarchical HMMs	S**wipe(L, R)**, Circle, Wave, Point, Palm Forward, Grab	66.0 (10)	below average performance on continuous gestures
our	novel set of features, ET	**Swipe Up/Down/In/Out(L,R)**	95.0 (10)	scalable, does not use heuristics

Concluding, we should note that research within the area of gesture recognition has been turned towards deep learning approaches which make use of raw RGB-D data, e.g., as in the work of Zhang et al. [19] and sophisticated deep network architectures. However, although such methods are able to achieve higher accuracy rates, often without the need of hand-crafted features, they typically result to more sophisticated models that need large amounts of data to train. Also such models may not be able to run inference in real time using low-end hardware such as a Raspberry Pi, which is often adopted in several IoT projects. If on the other hand the model runs on a more powerful server the network overhead of the raw data streams may be a bottleneck. Therefore, we feel that hand-crafted features combined with traditional machine learning approaches will be still of use within the forthcoming years for data or computationally constrained applications.

Author Contributions: Conceptualization, G.P. and E.S.; data curation, G.P.; investigation, D.S.; methodology, E.S.; supervision, E.S., T.G. and P.M.; Validation, D.S.; writing—original draft, G.P., E.S. and T.G.; writing—review and editing, G.P., E.S., T.G. and P.M.

Funding: We acknowledge support of this work by the project SYNTELESIS "Innovative Technologies and Applications based on the Internet of Things (IoT) and the Cloud Computing" (MIS 5002521) which is implemented under the "Action for the Strategic Development on the Research and Technological Sector", funded by the Operational Programme "Competitiveness, Entrepreneurship and Innovation" (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bhattacharya, S.; Czejdo, B.; Perez, N. Gesture classification with machine learning using kinect sensor data. In Proceedings of the 2012 Third International Conference on Emerging Applications of Information Technology, Kolkata, India, 30 November–1 December 2012.
2. Lai, K.; Konrad, J.; Ishwar, P. A gesture-driven computer interface using kinect. In Proceedings of the Southwest Symposium on Image Analysis and Interpretation (SSIAI), Santa Fe, NM, USA, 22–24 April 2012.
3. Mangera, R.; Senekal, F.; Nicolls, F. Cascading neural networks for upper-body gesture recognition. In Proceedings of the International Conference on Machine Vision and Machine Learning, Prague, Czech Republic, 14–15 August 2014.
4. Miranda, L.; Vieira, T.; Martinez, D.; Lewiner, T.; Vieira, A.W.; Campos, M.F. Real-time gesture recognition from depth data through key poses learning and decision forests. In Proceedings of the 25th IEEE Conference on Graphics, Patterns and Images (SIBGRAPI), Ouro Preto, Brazil, 22–25 August 2012.
5. Ting, H.Y.; Sim, K.S.; Abas, F.S.; Besar, R. Vision-based human gesture recognition using Kinect sensor. In *The 8th International Conference on Robotic, Vision, Signal Processing Power Applications*; Springer: Singapore, 2014.
6. Albrecht, T.; Muller, M. Dynamic Time Warping (DTW). In *Information Retrieval for Music and Motion*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 70–83.
7. Celebi, S.; Aydin, A.S.; Temiz, T.T.; Arici, T. Gesture Recognition using Skeleton Data with Weighted Dynamic Time Warping. In Proceedings of the VISAPP, Barcelona, Spain, 21–24 February 2013; pp. 620–625.
8. Reyes, M.; Dominguez, G.; Escalera, S. Feature weighting in dynamic timewarping for gesture recognition in depth data. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 6–13 November 2011.
9. Ribó, A.; Warcho, D.; Oszust, M. An Approach to Gesture Recognition with Skeletal Data Using Dynamic Time Warping and Nearest Neighbour Classifier. *Int. J. Intell. Syst. Appl.* **2016**, *8*, 1–8. [CrossRef]
10. Ibanez, R.; Soria, Á.; Teyseyre, A.; Campo, M. Easy gesture recognition for Kinect. *Adv. Eng. Softw.* **2014**, 76, 171–180. [CrossRef]
11. Anuj, A.; Mallick, T.; Das, P.P.; Majumdar, A.K. Robust control of applications by hand-gestures. In Proceedings of the 5th Computer Vision Fifth National Conference on Pattern Recognition, Image Processing and Graphics (NCVPRIPG), Patna, India, 16–19 December 2015.

12. Gonzalez-Sanchez, T.; Puig, D. Real-time body gesture recognition using depth camera. *Electron. Lett.* **2011**, 47, 697–698. [CrossRef]
13. Gu, Y.; Do, H.; Ou, Y.; Sheng, W. Human gesture recognition through a kinect sensor. In Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), Guangzhou, China, 11–14 December 2012.
14. Tran, C.; Trivedi, M.M. 3-D posture and gesture recognition for interactivity in smart spaces. *IEEE Trans. Ind. Inform.* **2012**, *8*, 178–187. [CrossRef]
15. Yin, Y.; Davis, R. Real-time continuous gesture recognition for natural human-computer interaction. In Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Melbourne, Australia, 28 July–1 August 2014.
16. Lin, C.; Wan, J.; Liang, Y.; Li, S.Z. Large-Scale Isolated Gesture Recognition Using a Refined Fused Model Based on Masked Res-C3D Network and Skeleton LSTM. In Proceedings of the 13th IEEE International Conference on Automatic Face and Gesture Recognition, Xi'an, China, 15–19 May 2018.
17. Wang, H.; Wang, L. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
18. Mathe, E.; Mitsou, A.; Spyrou, E.; Mylonas, P. Arm Gesture Recognition using a Convolutional Neural Network. In Proceedings of the 2018 13th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP), Zaragoza, Spain, 6–7 September 2018.
19. Zhang, L.; Zhu, G.; Shen, P.; Song, J.; Shah, S.A.; Bennamoun, M. Learning Spatiotemporal Features Using 3DCNN and Convolutional LSTM for Gesture Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3120–3128.
20. Mitra, S.; Acharya, T. Gesture recognition: A survey. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2007**, *37*, 311–324. [CrossRef]
21. Wang, S.B.; Quattoni, A.; Morency, L.P.; Demirdjian, D.; Darrell, T. Hidden conditional random fields for gesture recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; Volume 2, pp. 1521–1527.
22. Zhang, Z. Microsoft kinect sensor and its effect. *IEEE Multimedia* **2012**, *19*, 4–10. [CrossRef]
23. Shotton, J.; Sharp, T.; Kipman, A.; Fitzgibbon, A.; Finocchio, M.; Blake, A.; Cook, M.; Moore, R. Real-time human pose recognition in parts from single depth images. *Commun. ACM* **2013**, *56*, 116–124. [CrossRef]
24. Sheng, J. *A Study of Adaboost in 3d Gesture Recognition*; Department of Computer Science, University of Toronto: Toronto, ON, Canada, 2003.
25. Rubine, D. *Specifying Gestures by Example*; ACM: New York, NY, USA, 1991; Volume 25, pp. 329–337.
26. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
27. Vapnik, V.N. *Statistical Learning Theory*; Wiley: Hoboken, NJ, USA, 1998.
28. Cover, M.T.; Hart, E.P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
29. Domingos, P.; Pazzani, M. On the optimality of the simple Bayesian classifier under zero-one loss. *Mach. Learn.* **1997**, *29*, 103–130. [CrossRef]
30. McLachlan, G. *Discriminant Analysis and Statistical Pattern Recognition*; Wiley: Hoboken, NJ, USA, 2004.
31. Breiman, L.; Friedman, J.; Stone, C.J.; Olshen, R.A. *Classification and Regression Trees*; Taylor & Francis: Abingdon, England, 1984.
32. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
33. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3–42. [CrossRef]
34. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory*; Springer: Berlin/Heidelberg, Germany, 1995; pp. 23–37.
35. Li, W.; Zhang, Z.; Liu, Z. Action recognition based on a bag of 3d points. In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, San Francisco, CA, USA, 13–18 June 2010.

36. Arici, T.; Celebi, S.; Aydin, A.S.; Temiz, T.T. Robust gesture recognition using feature pre-processing and weighted dynamic time warping. *Multimedia Tools Appl.* **2014**, 3045–3062. [CrossRef]
37. Sfikas, G.; Akasiadis, C.; Spyrou, E. Creating a Smart Room using an IoT approach. In Proceedings of the Workshop on AI and IoT (AI-IoT), 9th Hellenic Conference on Artificial Intelligence, Thessaloniki, Greece, 18–20 May 2016.
38. Pierris, G.; Kothris, D.; Spyrou, E.; Spyropoulos, C. SYNAISTHISI: An enabling platform for the current internet of things ecosystem. In Proceedings of the Panhellenic Conference on Informatics, Athens, Greece, 1–3 October 2015; ACM: New York, NY, USA, 2015.
39. Peng, X.; Wang, L.; Cai, Z.; Qiao, Y. Action and gesture temporal spotting with super vector representation. In *European Conference on Computer Vision (ECCV)*; Springer: Cham, Switzerland, 2014.
40. Camgoz, N.C.; Hadfield, S.; Koller, O.; Bowden, R. Using convolutional 3d neural networks for user-independent continuous gesture recognition. In Proceedings of the 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016.
41. Hachaj, T.; Ogiela, M.R. Full body movements recognition–unsupervised learning approach with heuristic R-GDL method. *Digit. Signal Process.* **2015**, *46*, 239–252. [CrossRef]