Human Activity Recognition Process Using 3-D Posture Data

Salvatore Gaglio, *Member, IEEE*, Giuseppe Lo Re, *Senior Member, IEEE*, and Marco Morana

***Abstract*—In this paper, we present a method for recognizing human activities using information sensed by an RGB-D camera, namely the Microsoft Kinect. Our approach is based on the estimation of some relevant joints of the human body by means of the Kinect; three different machine learning techniques, i.e., K-means clustering, support vector machines, and hidden Markov models, are combined to detect the postures involved while performing an activity, to classify them, and to model each activity as a spatiotemporal evolution of known postures. Experiments were performed on Kinect Activity Recognition Dataset, a new dataset, and on CAD-60, a public dataset. Experimental results show that our solution outperforms four relevant works based on *RGB-D image fusion*, *hierarchical Maximum Entropy Markov Model*, *Markov Random Fields*, and *Eigenjoints*, respectively. The performance we achieved, i.e., precision/recall of 77.3% and 76.7%, and the ability to recognize the activities in real time show promise for applied use.**

***Index Terms*—Human activity recognition, kinect.**

I. Introduction

HERE, we present a novel technique to perform user activity recognition by means of an unobtrusive motion sensor device. In particular, we adopt the Microsoft Kinect as a motion sensor mainly due to its reliability, competitive cost, and its usage for user tracking. The output of the framework proposed here (i.e., the probability of the recognized activity) represents one of the inputs of a more general activity recognition system, which reasons about different information coming from the sensing infrastructure.

Human activities can be described as spatiotemporal evolutions of different body postures. We model the human body as a set of *joints* connecting some relevant body parts (e.g., arms or legs), and then, the most significant configurations of joint positions are used to define recurrent *postures*.

Our solution uses three different machine learning techniques. First, a set of body joints is detected by means of the Kinect. Then, such a set is clustered by applying the K-means algorithm in order to discover the postures involved in each activity. The obtained postures are validated by support vector machines (SVMs) and hidden Markov models (HMMs) are finally applied to model each activity as a sequence of known postures.

For more widespread applicability, we chose to connect the Kinect to a miniature fanless computer, which is able to process the scene with minimum levels of obtrusiveness and low power consumptions (about 7 W).

Our current work includes three contributions. Our first contribution is to design an activity recognition method able to guarantee *an acceptable accuracy*, *real-time processing*, *low power consumption*. The second contribution is the release of the public Kinect Activity Recognition Dataset (KARD), which contains 18 Activities, divided into ten gestures and eight actions, each performed three times by ten different subjects. The third contribution is the validation of the proposed method against a well-known public dataset.

This paper is organized as follows. Related work is outlined in Section II. The system architecture is described in Section III. Section IV presents the experimental scenario and the results for two different datasets. Conclusions are presented in Section V.

II. Related Work

First, we review some related activity recognition works based on RGB or RGB-D streams. Then, we provide a brief description of existing activity datasets.

A. Activity Recognition Methods

Early techniques focused on the processing of color images captured by traditional RGB cameras. In [1], the human body was represented in terms of silhouettes, extracted from RGB images, which were used as input to a framework based on HMM. Silhouettes and discrete HMMs are also used in [2], where authors applied Fourier analysis to describe the human silhouettes and SVMs [3] to classify them into different postures. The general weakness of the methods based on RGB data is that the complexity of the processing chain (e.g., background removal, vector quantization, image normalization), required to obtain adequate silhouette features, limits real-time use. Moreover, such systems are not robust enough to be applied in unconstrained situations, e.g., environments with complex backgrounds or low lighting conditions.

Dense approaches, as those based on salient points, which do not require segmentation, have been also proposed. The authors of [4] addressed the problem of activity recognition by analyzing the appearance of some points that are salient both in space and time. Each image sequence is represented in terms of spatiotemporal salient points and classified by means of K-nearest neighbor and relevance vector machines classifiers.

Manuscript received March 7, 2014; revised June 20, 2014, August 26, 2014, and November 13, 2014; accepted November 25, 2014. Date of publication December 18, 2014; date of current version September 14, 2015. This work was supported in part by the PO FESR 2007/2013 Grant G73F11000130004 funding the SmartBuildings project. This paper was recommended by Associate Editor F. Florez-Revuelta.

S. Gaglio is with the DICGIM, University of Palermo, 90128 Palermo, Italy, and also with the ICAR-CNR, National Research Council of Italy, 90128 Palermo, Italy (e-mail: salvatore.gaglio@unipa.it).

G. Lo Re and M. Morana are with the DICGIM, University of Palermo, 90128 Palermo, Italy (e-mail: giuseppe.lore@unipa.it; marco.morana@unipa.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/THMS.2014.2377111

An efficient technique based on a dense set of scale-invariant spatiotemporal features is proposed in [5]. The use of temporal scale-invariant features helps to recognize actions performed at different speeds, but also leads to errors when the speed is relevant to distinguish between similar actions (i.e., running versus walking). These approaches are useful to capture the most relevant cues of moving objects; thus, they perform well if the observed scene is made of a single subject acting in front of static background. Such a limit can be overcome by considering advanced devices capable of capturing both visual and depth information.

Some works addressed the problem of activity recognition by using intrusive sensors, i.e., wearable sensors [6], [7]. Such sensors provide more accurate information about the movements of the body; however, totally unobtrusive sensors, e.g., video sensors, are generally preferable to prevent users from wearing any electronic equipment and dealing with its maintenance.

Following these considerations, our perspective is to consider the Kinect as the primary sensor to transparently gather observations about users' behavior [8].

The vision system of the Microsoft Kinect is composed of two cameras (i.e., an RGB camera and an IR camera) with 640×480 resolution, and an IR projector that is responsible for shooting infrared rays toward the environment. The distortion degree of each ray projected against the scene is used to estimate a depth map in which each pixel value represents the distance of a specific 3-D point from the sensor.

Here, we review activity recognition approaches based on data provided by the Kinect. In [9], human bodies are modeled as a set of kinematic joints, and actions are defined by the interactions that occur between subsets of these joints. The authors proposed a new feature, called local occupancy feature (LOP), to describe each 3-D joint and introduced the concept of *actionlet* to define a particular conjunction of LOP features. Due to the great number of possible *actionlets*, a data mining technique is used to discover the most discriminative ones and represent an action as an *Actionlet Ensemble*, i.e., a combination of *actionlets*.

A posture-based approach for action recognition is presented in [10]. The authors represent salient postures as a bag of 3-D points obtained by projecting and sampling the depth maps onto three orthogonal planes. Each posture is then associated with a specific node of an action graph, which is used to model the dynamics of different actions. This method yields better results than those based on 2-D silhouettes; however, 3-D projections obtained from the depth maps are usually quite noisy due to low resolution of the sensor. Thus, further interpolation steps are generally required to repair corrupted projections, and this compromises the overall recognition time.

A histogram-based representation of human postures is presented in [11]. In this representation, the 3-D space is partitioned into n bins using a spherical coordinate system so that each of the 12 considered joints belongs to a bin with a certain level of uncertainty. Linear discriminant analysis (LDA) for C classes is performed to reduce the dimensions of the feature space from n to $C - 1$, and the obtained features are clustered into K visual words. The activities are then represented as sequences of visual words and recognized using discrete HMM classifiers. The features are detected in real time using a C language program, while activity recognition is simulated in MATLAB. The main limitations of this approach are the adoption of a complex model for representing the joints and the consequent need for reducing the dimensionality of the feature vectors by means of LDA. In [8], we observed that if the feature space already contains an optimal set of features, the attempt of further reducing such a space by means of principal component analysis or LDA does not increase the overall performance of the system, but may instead prevent the achievement of real-time processing.

An improved spherical angular representation is used in [12], where a gesture recognition for natural user interface is described. Different poses are defined according to the position of nine joints (six torso joints are discarded), each represented by a pair of spherical angles. A multiclass classifier is applied to identify relevant poses; then, gesture recognition is performed by means of a decision tree whose nodes represent key poses and leafs are associated with gestures. The main limitation of this approach is the need for designing and training the set of key poses, which is often infeasible in dynamic environments occupied by occasional users, e.g., offices.

The authors of [13] addressed the problem of reconstructing valid movements from incomplete, i.e., noisy, postures captured by the Kinect. In particular, broken postures are corrected by searching through a motion database for similar postures, which are kinematically valid. Although the method improves wrongly detected postures, it assumes that the motion database always contains postures similar to the ones performed by the user, which is not always true in practical situations.

A method to obtain silhouettes from depth information only is presented in [14]. This solution is motivated by the fact that depth images are intensity invariant and then more robust to appearance variations of the human body than RGB ones. The authors trained their system by creating a codebook of body poses so that a new human pose can be represented by its most similar codeword. The major issue of this approach is related to the background removal routine, which needs background images to be known previously, or users to be located away from the background. Such constraints are not always applicable to real contexts.

The authors of [15] proposed an algorithm based on hierarchical maximum entropy Markov model (MEMM) to represent a single activity as a composition of a set of subactivities. Each subactivity is initially modeled by analyzing about 700 features extracted from RGB and depth images; then, it is associated with a high-level activity by means of a two-layer MEMM. In [16], the problem of understanding human activities and their association with object affordances was addressed. Activity recognition was performed by means of Markov random fields whose nodes represent objects and subactivities and edges represent their mutual relations. A comparison with [15], [16], and other methods using the CAD-60 dataset is reported in Section IV-E.

The framework proposed in [17] aims to demonstrate that using both depth and grayscale data can improve the performance of recognizing complex activities, e.g., users interacting with objects in the environment. Experimental results show that promising recognition and localization accuracies can be obtained, but a computation time analysis is missing. Therefore, suitability for real-time applications is unknown.

The effectiveness of using both color and depth information for activity recognition is also reported in [18]. The authors collected a dataset, called RGBD-HuDaAct, which contains 12 activities performed by 30 different subjects at a distance of about 3 m from a Kinect device. Results obtained by applying multimodal feature representation, i.e., combining color and depth information, are compared to the unimodal counterparts; however, neither an evaluation of time consumption nor a comparison with other approaches is provided.

B. Activity Recognition Datasets

In [19], it is shown that to achieve good recognition rates, collected data should ideally contain both correct examples (correctness) and the set of the natural variations of the movements associated with a gesture (coverage). We next provide an overview of some public activity datasets.

The MSRC-12 dataset [19] consists of 12 gestures performed by 30 people. The gestures are organized into two abstract classes: *iconic gestures* that have a correspondence between the gesture and the reference (*crouch or hide, put on night vision goggles, shoot a pistol, throw an object, change weapon, kick*), and *metaphoric gestures* that represent an abstract concept (*start system/music/raise volume, navigate to next menu/move arm right, wind up the music, take a bow to end music session, protest the music*, and *move up the tempo of the song/beat both arms*). The provided files contain the coordinates of 20 joints captured at a sample rate of 30 frames/s; however, some sequences are not useful since the estimation of the joints is not accurate.

The MSRDailyActivity3D dataset [20] contains 16 activities performed in front of the Kinect sensor: *drink, eat, read book, call cellphone, write on a paper, use laptop, use vacuum cleaner, cheer up, sit still, toss paper, play game, lie down on sofa, walk, play guitar, stand up*, and *sit down*. Each activity was performed twice, once in standing position and once in sitting position, by ten different subjects. Three channels were recorded: depth maps (.bin), skeleton joint positions (.txt), and RGB video (.avi). However, the RGB and depth channels were recorded independently; therefore, they are not strictly synchronized. Another lack of this dataset is that for each action, only two different sequences (acquired in standing/sitting positions) are provided; therefore, it is difficult to train and test a robust classifier having just these few examples.

The MSRAction3D dataset [10] contains 20 actions repeated three times by ten different subjects: *high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, sideboxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, and pickup and throw*. Its main limitation is that it was recorded by means of a depth sensor (similar to the Kinect), which was not able to capture RGB information.

The dataset presented in [21], called LIRIS, was used for the ICPR 2012 human activities and localization competition which focused on the problem of recognizing complex human behaviors involving several people. LIRIS was captured by means of two different cameras: a Kinect device mounted on a mobile robot mobile delivering grayscale and depth images, and a consumer camcorder delivering high-resolution videos. The actions available in the dataset are: *discussion of two or several people, a person gives an item to a second person, an item is picked up or put down, a person enters or leaves a room, a person tries to enter a room unsuccessfully, a person unlocks a room and then enters it, a person leaves baggage unattended, handshaking of two people, a person types on a keyboard*, and *a person talks on a telephone*.

In [22], a multimodal dataset (Multimodal Human Action Database MHAD—MHAD) is proposed. The MHAD database contains 11 actions, performed by 12 individuals, captured by means of an optical motion capture system based on 43 LED markers, 12 multiview stereo vision cameras, two Microsoft Kinect cameras, six three-axis wireless accelerometers, and four microphones. The method proposed in [10] has been applied to model the action sequence captured by each modality, while Gehler and Nowozin [23] was used to combine various modality (e.g., motion capture and Kinect, motion capture and accelerometers and Kinect, motion capture and accelerometers, and Kinect and audio). Results show that using multimodal data increases the recognition rate because multimodal features usually compensate for each other. However, hardware costs and the needs for continuous maintenance (i.e., preserving both geometric calibration and temporal synchronization) limit use for real-world activity recognition purposes.

The Cornell Activity Dataset (CAD-60) [15] contains 60 RGB-D videos of four subjects performing 12 activities (*rinsing mouth, brushing teeth, wearing contact lens, talking on the phone, drinking water, opening pill container, cooking-chopping, cooking-stirring, talking on couch, relaxing on couch, writing on whiteboard*, and *working on computer*) in five different environments (*office, kitchen, bedroom, bathroom*, and *living room*). The authors also maintain a website [24] with reported results of activity recognition techniques.

In [25], an activity recognition method based on a bag-of-words model is proposed. The authors used SVMs with dynamic time warping (DTW) kernel functions to restore temporal relationships within time series of codeword histograms. Tests were performed on three different datasets including ReadingAct, a novel dataset (not yet available for download) captured by means of two Kinect devices, which contains 19 actions performed by 20 subjects. Results show that the DTW-SVM approach slightly improves the results on long actions sequences, while it performs as other state-of-the-art methods in general.

III. Activity Recognition System

The system proposed here (see Fig. 1) aims at automatically inferring the activity performed by the user according to a set of known postures. The system can be decomposed into three components addressing three different aspects. The first is responsible for *features detection*, that is for the extraction of a set of points to be used for distinguishing different

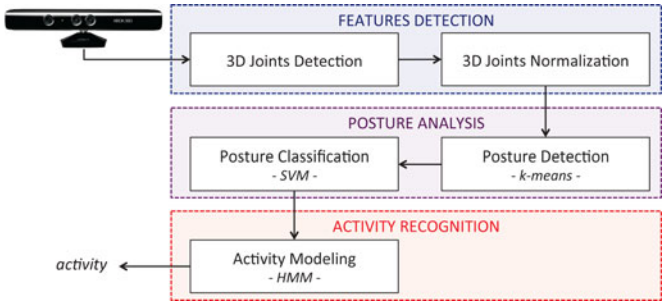

Fig. 1. Images captured by the Kinect are processed to detect a set of joints, which are subsequently normalized with respect to scale and position. These joints represent the features used to define a set of postures, which are detected by applying a K-means clustering and classified by means of SVMs. HMMs are finally used to model an activity in terms of postures and classify new sequences coming from the Kinect.

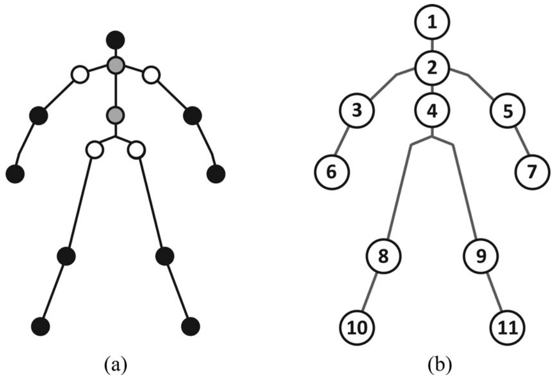

Fig. 2. (a) Fifteen joints detected by means of the Kinect. Reference joints (gray): *neck*, *torso*. Selected joints (black): *head*, *elbows*, *hands*, *knees*, *feet*. Discarded joints (white): *shoulders*, *hips*. (b) Eleven joints of the feature set.

body postures. The detection and classification of such postures is accomplished by the *posture analysis* techniques, based on K-means and SVM, and, finally, *activity recognition* is performed by means of HMMs built on the set of known postures.

A. Features Detection

The first processing step consists in identifying the features of interest. Since our goal is to understand what activity the user is performing at a given time, we need to track movements focusing on those body parts, which are mostly involved while executing a particular activity.

The human body consists of many interacting systems, none of which can work in isolation. In particular, we started from the musculoskeletal system, which is responsible for supporting the human body and enabling its movements in accordance with the stimuli provided by the nervous system. To describe the user's movements, we chose to track the human skeleton focusing on significant parts such as head, neck, torso, arms, legs, hands, and feet. The different parts of the human skeleton can be modeled as segments connected to each other by nodes, called *joints*, which limit the movement of each body part in the 3-D space.

Thus, the 3-D positions of some relevant *joints* can be used to describe different movements of the body. To extract these features, we adopted the Kinect device, which has been demonstrated to be an unobtrusive sensor to perform real-time detection (i.e., to determine the 3-D coordinates) of a number of body joints (see Fig. 2).

Unfortunately, due to the intrinsic noise of the sensor and the peculiarities of the human body, not all joints are equally informative; thus, a mechanism to select the most promising ones is required. In [26], this task is accomplished by means of an evolutionary algorithm, which determines the optimal subset of skeleton joints according to a specific training set. Although such an approach improves the recognition rate in the specific case, the joint selection process is too data centric, and any variation on the activity set causes the selection of different subsets of joints.

Since we are interested in a more general representation suitable for a dynamic environment, we performed some preliminary tests to measure the relevance of the set of joints provided by the Kinect. In [8], due to the sensitiveness of the IR sensor, some joints are misdetected if two segments overlap (e.g., hands touching other body parts), or not detected at all due to the presence of objects between the sensor and the user. For this reason, we evaluated the system by measuring the recognition rates achieved on a limited number of selected subset of joints.

Some noisy joints that are redundant (i.e., wrists, ankles) due to their closeness to other joints (i.e., hands, feet) or not relevant at all for activity recognition (i.e., spine, neck, hip and shoulders) have been discarded. The final set of joints we chose as features is shown in black in Fig. 2(a), while the joints we discarded are white.

Since the appearance of the skeleton depends on several factors, as, for example, the distance between the user and the sensor, the detected features need to be normalized for scale. For doing that, we moved the detected joints to a new coordinate system fixed at the torso (considering as the x-direction the left-right hip axis) and all features have been scaled according to a reference distance, h, between the neck and the torso joints. The reference joints are shown in gray in Fig. 2(a).

Thus, let $\mathbf{J_i}$ be one of the 11 joints detected by means of the Kinect, the feature vector $\mathbf{f}$ is defined as

$$\mathbf{f} = [\, \mathbf{j}_1, \mathbf{j}_2, \mathbf{j}_3, \mathbf{j}_4, \mathbf{j}_5, \mathbf{j}_6, \mathbf{j}_7, \mathbf{j}_8, \mathbf{j}_9, \mathbf{j}_{10}, \mathbf{j}_{11} \,] \tag{1}$$

where each $\mathbf{j}_i$ is the vector containing the 3-D normalized coordinates of the ith joint $\mathbf{J}_i$ detected by the Kinect. Thus

$$\mathbf{j_i} = \frac{J_i}{s} + T, \qquad 1 \leq i \leq 11 \tag{2}$$

being s the scale factor which normalizes the skeleton according to the distance, h, between the neck and the torso joints of a reference skeleton (detected offline)

$$s = \frac{\|J_4 - J_2\|}{h} \tag{3}$$

and T the translation matrix needed to set the origin of the coordinate system to the torso.

We do not normalize for rotation since some preliminary results showed that the angle between the user and the Kinect is an important cue for our method. This is mainly due to two aspects. The first is that the rotation of the user with respect to the Kinect is important for recognizing some full-body activities such as *walk* or *take umbrella*. In these cases, a rotation-invariant representation would produce flat poses, which could be more frequently misclassified. The second is that, even if it is usually convenient to adopt a rotation invariant representation, we can overcome this limitation by placing the Kinect within the office so that almost frontal activities are observed.

B. Posture Analysis

As already mentioned, our idea is that each activity can be considered as a sequence of different configurations of joints. In order to identify those configurations that are effectively related to meaningful users postures, a classification procedure is needed.

SVMs [3] are supervised learning models used for binary classification and regression, which aim to find the optimal separating hyperplane between two classes according to some labeled training samples. Unfortunately, building the training set on large-scale data is a costly operation, which may also lead to worse performance because of the presence of noise. Thus, a more effective way of building the training set could be to select the most informative samples, that is, in our case, the most recurrent joint configurations. We chose to perform such a selection process by means of a clustering algorithm. In particular, given the set of feature vectors $(f_1, f_2, \ldots, f_n)$, the K-means algorithm is applied to partition the n observations into k sets, $\mathbf{C} = (C_1, C_2, \ldots, C_k)$, so as to minimize the intracluster error

$$E = \sum_{j=1}^{k} \sum_{f_i \in C_j} \|f_i - \mu_j\|^2 \tag{4}$$

where μ_j is the mean value of the jth set, i.e., cluster, C_j.

The k generated clusters are representative of recurrent postures and can be used to train a multiclass SVM classifier on the set $T = \{(C_1, L_1), (C_2, L_2), \ldots (C_k, L_k)\}$, where (C_k, L_k) is the kth pair *(cluster, cluster label)* produced by K-means.

A multiclass SVM is usually implemented by combining several binary SVMs according to three main strategies: one-versus-all, one-versus-one, and directed acyclic graphs SVM. Several studies addressed the issue of evaluating which is the best multiclass SVM method, and both studies [27] and [28] claimed that the one-versus-one approach is preferable to other methods. For a problem with k classes, this strategy consists in constructing $k(k-1)/2$ SVMs classifiers, which are trained to distinguish samples from two different classes. After all $k(k-1)/2$ classifiers are constructed, the classification is done according to a “max wins” voting strategy.

The process of classifying the detected features into k classes can be viewed as building a k-words vocabulary. Each posture can be represented as a single word of the vocabulary, i.e., cluster center, and therefore, each activity can be considered as an ordered sequence of vocabulary words.

Transforming sequences of joints configurations into sequences of k-words allows merging all repeated instances of a same posture, that is, we focus only on posture transitions. Thus, we have two advantages: the first is that a more compact representation of the sequences is obtained; and the second is that we overcome the problem of recognizing the same activities performed at different speeds. Moreover, the posture-based representation does not affect the capacity of the system to distinguish among different activities with different durations. In those cases, a greater number of postures would be involved making longer activities intrinsically different from the shorter ones. In Fig. 3, an example of the posture sequence extracted from one repetition of the “high arm wave” gesture is shown.

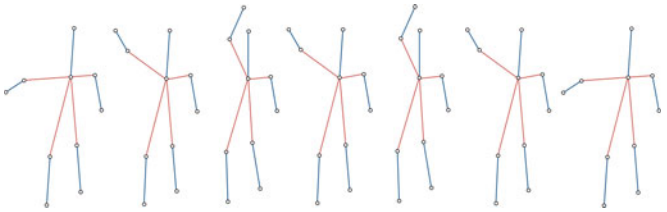

Fig. 3. Posture sequence from one repetition of the “high arm wave” gesture.

C. Activity Recognition

In order to fully satisfy the design requirements, the system should also correctly classify multiple instances of the same activity, which may generally involve different sequences of postures.

The activity recognition process is based on HMMs similarly to what is described in [11] and [29]. We modeled each activity using a discrete HMM, whose observed symbols are the postures we have previously extracted.

In a system whose instantaneous condition may be represented as belonging to one of N distinct states, we denote the different states as $S = \{S_1, S_2, \ldots, S_N\}$, and the state at time t as q_t.

Given the set of prior probabilities $\pi = \{\pi_i\}$

$$\pi_i = P[q_1 = S_i], \quad 1 \le i \le N \tag{5}$$

where π_i are the probabilities, assumed equiprobable, of S_i being the first state of a state sequence; the state transition probability $A = \{a_{ij}\}$, from the state S_i to the state S_j, is

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \quad 1 \le i, j \le N. \tag{6}$$

Let M be the number of distinct observation symbols per state, the individual symbols are $V = \{v_1, v_2, \ldots, v_M\}$, and the observation symbol probability distribution in state j, $B = \{b_j(k)\}$ is

$$b_j(k) = P[v_k \ \ at \ \ t \,|\, q_t = S_j], \quad 1 \le j \le N, \quad 1 \le k \le M. \tag{7}$$

The complete parameter set of the model is the triplet

$$\lambda = (A, B, \pi). \tag{8}$$

The idea is to encode each activity in terms of postures and build the corresponding HMM. Once each HMM has been trained on the posture sequences of each activity, a new (unknown) sequence is tested against the set of HMMs and classified according to the largest posterior probability. Otherwise,

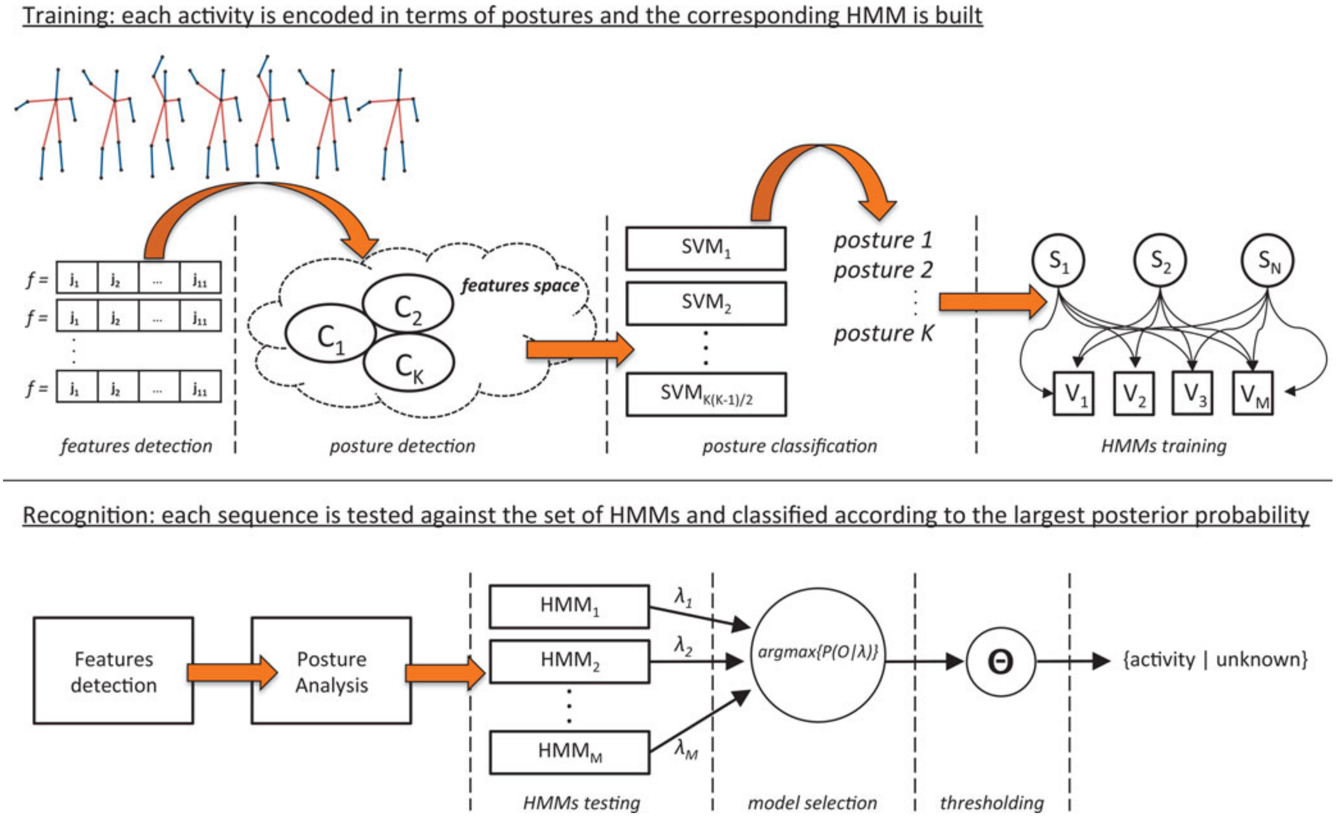

Fig. 4. Activity recognition process. During the training, each activity is analyzed to extract a set of postures, which are used to build an HMM. A new activity is recognized by testing the corresponding posture sequence against the set of HMMs and selecting the model with the largest posterior probability.

if such a probability is below a fixed threshold, the sequence is marked as "unknown."

The parameters (k, N) have been experimentally computed by performing an exhaustive search through a subset of values. In particular, a Grid Search [30] guided by a leave-one-out cross validation (LOOCV) [31] has been applied.

The activity recognition process is described in Fig. 4. The training phase consists of four steps: 1) for each activity, the features of interest are detected; 2) the features space is organized into k clusters, which represent the most significative postures; 3) the detected postures are refined by means of SVMs classification; and 4) an HMM which models the activity is built. To recognize an activity, we need to 1) detect the features, 2) detect and classify the postures involved in the activity, 3) test the posture sequence against all HMMs; 4) select the model which maximizes the posterior probability, and 5) compare such probability against a threshold to classify an activity as known or unknown.

IV. Results

A. Case Study

The activity recognition technique discussed here was developed as part of an AmI system [32] designed to perform timely and ubiquitous monitoring of a complex of buildings to optimize energy consumption [33]. From a logical point of view, the reference model of the AmI system is composed of three layers: the *sensing layer*, responsible for monitoring and controlling the environment by means of heterogeneous sensors and actuators [34]; the *middleware layer*, which provides a standard interface between physical sensors and AmI algorithms; the *intelligent layer*, which implements the AmI functionalities and produces the necessary actions to adapt the environment to the user requirements [35]. A prototype of the system was built at the Networking and Distributed Systems Lab of the University of Palermo.

The office is equipped with wireless and wired sensor nodes, which monitor the environment conditions and the status of the actuators, respectively [32]. For example, RFID readers are installed close to each office door providing information about the presence of a particular user, while software sensors are installed to detect the users' activities on their workstations. In this scenario, the Kinect is one among several sensors deployed in the office, and its specific assignment is to provide information about the activities performed by the user.

B. Data Analysis Apparatus

The activity recognition module ran on an Intel Atom Z530 1.6-GHz CPU and Linux OS with kernel 2.6.32. Such a small device guaranteed real-time processing of the observed scene with low levels of obtrusiveness and low power consumptions, demonstrating both the effectiveness of the solutions and the efficiency of the algorithms.

The results presented have been obtained by simulating the overall system in MATLAB on a desktop PC equipped with a 2.6-GHz dual-core microprocessor.

C. KARD—Kinect Activity Recognition Dataset

We collected a new dataset, called KARD, paying special attention to the correctness both of the acquired data itself and the ground truth [36]. KARD contains 18 activities, divided into ten gestures (*horizontal arm wave, high arm wave, two hand wave, high throw, draw x, draw tick, forward kick, side kick, bend*, and *hand clap*) and eight actions (*catch cap, toss paper, take umbrella, walk, phone call, drink, sit down*, and *stand up*).

The distinction between these two classes of activities is useful to better evaluate the performance of the system both on simple sequences, which separately involve specific parts of the body, i.e., gestures, and on complex actions where different parts of the body interact to each other.

Each activity was repeated three times by ten different individuals (nine males and one female) with ages ranging from 20 to 30 years and height from 150 to 185 cm. Instructions were given to the users about what activity to perform, e.g., *"clap your hands," "catch the cap,"* without providing information on *how* to perform it, so as to guarantee the naturalness of the movements.

The dataset was captured by means of a Kinect device placed about 2–3 m from the subject, in an office scene containing a desk, a phone, a coat rack, and a waste bin.

KARD is made of 540 sequences for about a total of 1 h of videos captured at a resolution of 640 × 480 pixels at 30 frames/s. For each sequence, we provide both the RGB and depth images, and the list of the detected joints in real world and screen coordinates.

D. Experiments on Kinect Activity Recognition Dataset

We investigated both the ability of the system to distinguish between similar activities and the scalability of our solution. Two different classes of tests are described. The former, called *model test*, aims to evaluate how the accuracy of our framework depends on the complexity of the chosen model, that is, how much the recognition rate is influenced by the model parameters (i.e., the number of postures and the hidden states). The latter, called *data test*, aims to evaluate if the accuracy is related to the properties of the training set, that is, how much the recognition rate is influenced by the number of the observed subjects and the characteristics of the performed activities.

1) Model Test: The first test aimed to find the best pair of values for the number of clusters k (i.e., the number of postures) and the number of the HMM states N. A Grid Search approach [30] was applied to search for the values of k, in the range $[15; 51]$, and N, in $[3; 17]$. The value (k, N) of each node of the grid was computed as the mean rate of an LOOCV [31] repeated ten times to overcome the randomness of the clustering algorithm.

We used 539 training sequences and one testing sequence. The results are shown in Fig. 5. The best recognition rate is obtained for $k = 39$ and $N = 5$, with a mean accuracy of 95% and standard deviation of 2.45 between the different runs of the LOOCV.

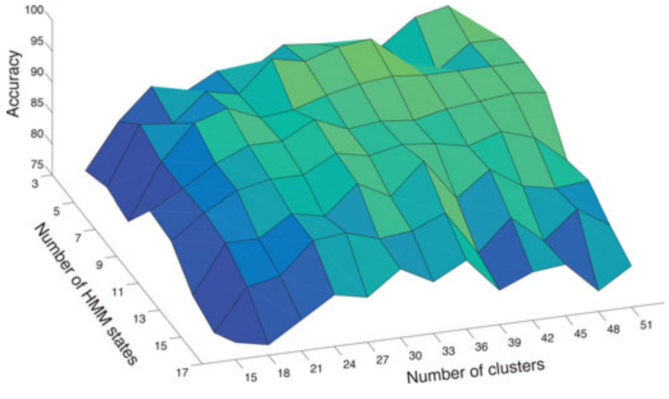

Fig. 5. Results of grid search for $k \in [15, 51]$ and $N \in [3, 17]$.

TABLE I
ACCURACY ON KARD TESTED FOR $k = 39$ AND $N = 5$

Gestures		Actions	
Horizontal arm wave	92%	Catch Cap	100%
High arm wave	96%	Toss Paper	90%
Two hand wave	96%	Take Umbrella	96%
High throw	80%	Walk	100%
Draw x	96%	Phone Call	96%
Draw tick	90%	Drink	86%
Forward Kick	96%	Sit down	100%
Side Kick	100%	Stand up	100%
Bend	96%		
Hand Clap	100%		

Table I shows the results obtained for the 18 activities. The highest recognition rate is 100% (*side kick, hand clap, catch cap, walk, sit down*, and *stand up*), while the worst is 80% (*high throw*). Since the recognition rate appears to be stable, we can conclude that there is no bias of the proposed method toward a particular activity or subset of activities. This indicates the effectiveness of both the chosen feature space and its representation, that is, the model we used is able to capture the key points of different kinds of activities, regardless of the parts of the body they involve.

In Table II, the confusion matrix for this experiment is shown. In some cases, the system failed in recognizing similar activities that involve similar postures, e.g., a few times *high throw* was recognized as *drink* since the performed movements are very similar, while only a few instances of five activities (i.e., *two hand wave*, *forward kick*, *take umbrella*, *bend*, *phone call*) were classified as "unknown."

The experiment was also repeated including all 15 joints depicted in Fig. 2, and we observed a reduction of the mean recognition rate of about 4%. This confirms that excluding joints that are not relevant improves the performance both in terms of accuracy and dimension of the representation space.

2) Data Test: The second class of tests aimed to measure the performance that the system can achieve while varying the training set. In particular, the goals are:

TABLE II
LOOCV CONFUSION MATRIX FOR THE KARD TESTED FOR $k = 39$ AND $N = 5$

	Horizontal arm wave	High arm wave	Two hand wave	Catch Cap	High throw	Draw X	Draw tick	Toss Paper	Forward Kick	Side Kick	Take Umbrella	Bend	Hand Clap	Walk	Phone Call	Drink	Sit down	Stand up	Unknown
Horizontal arm wave	.92					.04	.04												
High arm wave		.96			.02		.02												
Two hand wave			.96																.04
Catch Cap				1															
High throw					.80		.08									.12			
Draw X	.02					.96	.02												
Draw tick	.04				.02	.04	.90												
Toss Paper								.90				.10							
Forward Kick									.96										.04
Side Kick										1									
Take Umbrella											.96								.04
Bend											.02	.96							.02
Hand Clap													1						
Walk														1					
Phone Call															.96	.02			.02
Drink		.06			.04		.04									.86			
Sit down																	1		
Stand up																		1	

TABLE III
ACCURACY (%) FOR THE MODEL TEST CONSIDERING GESTURES AND ACTIONS SEPARATELY

	Gestures	**Actions**
Experiment A	86.5	92.5
Experiment B	93.0	95.0
Experiment C	86.7	90.1

TABLE IV
KARD ACTIVITIES ORGANIZED INTO THREE ACTIVITY SETS WITH DIFFERENT LEVELS OF DIFFICULTY

Activity Set 1	**Activity Set 2**	**Activity Set 3**
Horizontal arm wave	High arm wave	Draw Tick
Two hand wave	Side Kick	Drink
Bend	Catch Cap	Sit Down
Phone Call	Draw tick	Phone Call
Stand Up	Hand Clap	Take Umbrella
Forward Kick	Forward Kick	Toss Paper
Draw x	Bend	High throw
Walk	Sit Down	Horizontal arm wave

TABLE V
ACCURACY (%) FOR THE MODEL TEST USING THREE DIFFERENT ACTIVITY SETS

	Activity Set 1	**Activity Set 2**	**Activity Set 3**
Test A	95.1	89.9	84.2
Test B	99.1	94.9	89.5
Test C	93.0	90.1	81.7

1) To measure the recognition rate of the system for actions and gestures separately;
2) To measure the recognition rate when considering gestures or actions based on very similar postures.

The dataset is divided into subsets and each subset is tested three times similar to [10]:

1) *Experiment A*: One-third of the samples of each subject is used for training and the rest for testing.
2) *Experiment B*: Two-third of the samples of each subject is used for training and the rest for testing.
3) *Experiment C*: Half of the samples is used for training set and the rest for testing.

Each of the above experiments was repeated ten times, randomly choosing the sequences or subjects of the training and testing sets. Results are shown in Table III .

Our second goal was to measure the performance of the system in analyzing similar activities. Thus, we divided the data into three subsets with different levels of difficulty (see Table IV). In particular, the Activity Set 1 is made up of very different activities, the Activity Set 2 contains more similar activities than the previous one, and the Activity Set 3 is composed of very similar activities. The system performed as we expected, that is, better results are obtained on Activity Set 1, as shown in Table V. Test B showed better results over the three activity sets, while worst performances are obtained with Test C. Test A showed that the system performs well when it uses only one repetition of each activity per subject, that is, the system is able to capture a general model of the activity regardless to the user that performed it. This is also confirmed by the results of Test C, where it is shown that once the system has been trained, it can recognize activities performed by new subjects.

E. Experiments on CAD-60

The Cornell Activity Dataset, CAD-60, as described in Section II, contains data collected from four different people.

TABLE VI
STATE-OF-THE-ART PRECISION AND RECALL VALUES (%) ON CORNELL ACTIVITY DATASET CAD-60

	Precision	**Recall**
Sung *et al.* [15]	67.9	55.5
Koppula *et al.* [16]	80.8	71.4
Yang and Tian [37]	71.9	66.6
Ni *et al.* [17]	75.9	69.5
Gupta *et al.* [14]	78.1	75.4
Our method	**77.3**	**76.7**

TABLE VII
PRECISION (%) AND RECALL (%) OF OUR METHOD IN THE FIVE ENVIRONMENTS OF CAD-60

		"new person"	
Location	**Activity**	**Precision**	**Recall**
bathroom	rinsing mouth	98.3	97.8
	brushing teeth	97.0	96.4
	wearing contact lens	78.1	77.6
	Average	**91.1**	**90.6**
bedroom	talking on phone	72.7	73.5
	drinking water	63.4	61.3
	opening container	76.0	73.2
	Average	**69.7**	**69.3**
kitchen	cooking (chopping)	72.6	75.3 \
	cooking (stirring)	59.3	58.0
	drinking water	74.3	72.7
	opening container	78.1	75.8
	Average	**71.1**	**70.5**
living room	talking on phone	69.0	66.4
	drinking water	73.4	71.1
	talking on couch	78.2	76.9
	relaxing on couch	73.4	77.2
	Average	**73.5**	**72.9**
office	talking on phone	72.3	73.4
	writing on whiteboard	84.3	87.4
	drinking water	78.4	75.3
	working on computer	90.0	85.6
	Average	**81.3**	**80.4**
Overall Average		**77.3**	**76.7**

Results are expressed in terms of precision and recall measured according to the "new person" scenario, that is, by training the system on three of the four people from whom data were collected, and testing on the fourth. We selected five works whose precision and recall values are summarized in Table VI. The results are shown in Table VII. This test is useful to evaluate the performance of the system in analyzing activities which involve similar postures. The results of the overall system evaluation on CAD-60 are reported in Table VIII.

Some activities, characterized by postures which involve very similar subsets of joints, e.g., *brushing teeth* and *drinking water*, or *cooking (chopping)* and *cooking (stirring)*, are more difficult to be recognized, while others are correctly classified. The overall precision and recall of our method are 77.3% and 76.7%, respectively. Comparing such values with the works listed in Table VI, we outperform four out of five, while we achieve comparable results with [14].

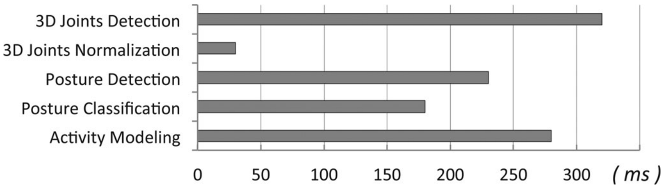

Fig. 6. Average processing time for each activity recognition step.

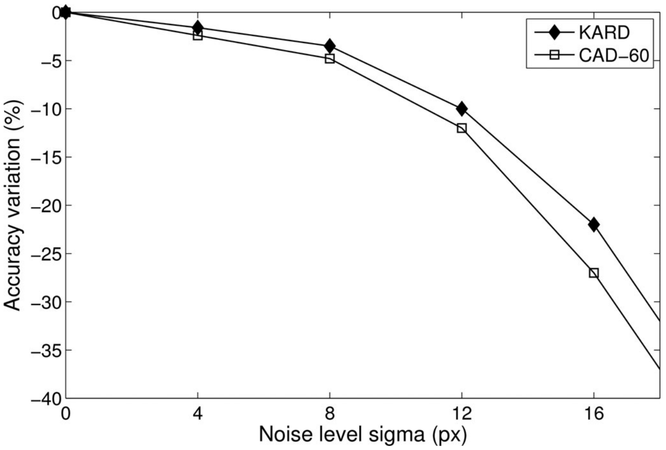

Fig. 7. Performance variations (%) measured by altering the joint estimation process with a Gaussian noise, centered on the joint with $\sigma \in [0, 20]$ pixels.

In order to make a comparison with the results obtained on CAD-60, we repeated the evaluation of our method on KARD, according to the "new person" scenario. The corresponding confusion matrix is reported in Table IX. The overall precision and recall we achieved are 84.8% and 84.5%, respectively. When comparing these results with the ones obtained on CAD-60, we noticed that the proposed system performs better with KARD data. The main reason is that the 12 activities of CAD-60 are more complex in terms of the involved postures than those contained in KARD. Thus, the single pose estimation errors accumulate making the recognition process less reliable.

F. Performance

The JAVA implementation of the system allows us to capture the Kinect stream at 30 frames/s and perform the recognition of a sequence (i.e., posture analysis and activity recognition) in about 1 s, with a power consumption of about 7 W, that is just 1 W more than the 6 W consumed during idle time.

Fig. 6 shows the average processing time measured for each step involved in the activity recognition module. The most time consuming algorithms are those responsible for detecting the joints and modeling the activity by means of HMMs, while posture detection and classification take less than half of the overall processing time.

G. Limitations

Most recognition issues are mainly due to the intrinsic limitations of the tracking algorithm [38]. In particular, when a body

TABLE VIII
CONFUSION MATRIX FOR THE "NEW PERSON" TESTS ON CAD-60, IRRESPECTIVE OF DIFFERENT ENVIRONMENTS

	Brushing teeth	Rinsing mouth	Wearing contact lens	Working on computer	Cooking (chopping)	Cooking (stirring)	Talking on the phone	Drinking water	Opening pill container	Talking on couch	Relaxing on couch	Writing on whiteboard
Brushing teeth	.48						.33	.19				
Rinsing mouth	.04	.96										
Wearing contact lens			1									
Working on computer				1								
Cooking (chopping)					.79	.17			.04			
Cooking (stirring)					.43	.55			.02			
Talking on the phone	.22	.10					.43	.25				
Drinking water	.20	.04					.13	.63				
Opening pill container				.02				.20	.78			
Talking on couch										.73	.27	
Relaxing on couch										.15	.85	
Writing on whiteboard												1

TABLE IX
CONFUSION MATRIX FOR THE "NEW PERSON" TESTS ON KARD

	Horizontal arm wave	High arm wave	Two hand wave	Catch Cap	High throw	Draw X	Draw tick	Toss Paper	Forward Kick	Side Kick	Take Um-brella	Bend	Hand Clap	Walk	Phone Call	Drink	Sit down	Stand up
Horizontal arm wave	.83					.08	.07						.02					
High arm wave		.81		.05	.07		.07											
Two hand wave	.02		.83			.03	.05						.07					
Catch Cap		.05		.85	.10													
High throw					.75		.05								.07	.13		
Draw X	.10					.80									.10			
Draw tick		.12				.10	.76									.02		
Toss Paper								.84				.16						
Forward Kick									.90	.02				.08				
Side Kick									.02	.98								
Take Umbrella								.02			.92	.06						
Bend								.02			.06	.92						
Hand Clap	.06		.10										.84					
Walk									.05	.05				.90				
Phone Call				.05	.02										.83	.10		
Drink		.08			.10		.04								.04	.74		
Sit down																	.87	.13
Stand up																	.15	.85

part is misdetected (e.g., due to partial occlusions), the skeleton tracker tries anyway to estimate its position according to a global body model. However, such a compensation process produces a domino effect, which makes the detection of the whole skeleton unreliable.

In order to evaluate how much the system performance is dependent on noisy joints, some experiments were performed by adding a Gaussian noise to each joint and measuring the system accuracy for different noise levels.

The results obtained both on the proposed dataset and on CAD-60 (see Fig. 7) show that when the noise is characterized by a standard deviation less than 10 pixels, slight accuracy variations can be observed, while the performance drops significantly for greater noise levels. This trend is not surprising given that the Kinect sensor is affected by an intrinsic noise; thus, slow variations on the left side of the curves suggest that our model is quite robust as long as the combination of intrinsic and additive noise is below a certain critical value. Greater values of sigma correspond to what happens when partial occlusions occur; for example, if $\sigma = 20$, the position of a joint is estimated with a precision of about ±60 pixels, that is, similar to what we observed when legs are hidden behind a desk.

Finally, Fig. 7 also shows that accuracy variations on CAD-60 are greater than those observed on KARD. Our dataset appears to be more reliable being characterized by a lower noise level.

V. Conclusion

In this study, we presented a framework for human activity recognition using 3-D posture data. In particular, we referred to a scenario where the whole environment is equipped with a number of sensory nodes capable of unobtrusive monitoring of some raw measures such as temperature, humidity, and light level. In this context, the Kinect is responsible for gathering high-level information about what the user is doing.

In order to obtain a suitable representation of the human body, we detected 11 relevant joints and encoded a relevant set of joints into *postures*. Thus, since each posture represents a recurrent pattern of joints positions, an activity can be described as a sequence of known postures.

To support a real office environment, we mainly focused on a solution made of simple processing blocks, which are functional in the scenario we considered. Other approaches could perform better on single tasks, e.g., providing more reliable posture representation mechanisms or more complex activity models, but we aimed to develop a framework which can be easily integrated in a more general AmI system.

To this end, we evaluated the effectiveness of our technique using two different datasets. The first is KARD, a new public dataset we collected to overcome the unreliability of some other existing data collections. The second is CAD-60, which allowed comparison with some state-of-the-art techniques.

The experiments showed that our method is able to capture a general model of the activity regardless of the user. In particular, the activity models we built are independent of who performs the action, independent of the speed at which the actions are performed, scalable to large number of actions, and expandable with new actions. Moreover, since repeated sequences of the same posture are merged, the proposed method is able to recognize the same class of activities performed with different time durations.

Using the public Cornell Activity Dataset, we obtained an overall precision and recall of 77.3% and 76.7%, respectively, demonstrating that our framework outperforms four of the techniques we considered as reference.

Due to the requirements of the overall AmI system, we implemented a real prototype of the activity recognition module by connecting the Kinect to a miniature computer getting a real-time processing of the observed scene with minimum levels of obtrusiveness and low power consumptions.

Analogously to other approaches, the main limitations or our system are primarily related to the capacity of the Kinect of providing a stable video stream and, consequently, a reliable joint detection mechanism. In this regard, future work can concern the improvement of the pose estimation process in order to deal with frame loss and body occlusions, which are the main causes of misclassification.

References

[1] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden Markov model," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recog. Proc.*, 1992, pp. 379–385.

[2] M. P. V. Kellokumpu and J. Heikkila, "Human activity recognition using sequences of postures," in *Proc. IAPR Conf. Mach. Vision Appl.*, 2005, pp. 570–573.

[3] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[4] A. Oikonomopoulos, I. Patras, and M. Pantic, "Spatiotemporal salient points for visual recognition of human actions," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 3, pp. 710–719, Jun. 2005.

[5] G. Willems, T. Tuytelaars, and L. Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proc. 10th Eur. Conf. Comput. Vision*, 2008, pp. 650–663.

[6] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, D. Howard, K. Meijer, and R. Crompton, "Activity identification using body-mounted sensors—A review of classification techniques," *Physiol. Meas.*, vol. 30, no. 4, pp. R1–R33, 2009.

[7] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive Computing* (ser. Lecture Notes in Computer Science, vol. 3001), A. Ferscha and F. Mattern, Eds. Berlin, Germany: Springer, 2004, pp. 1–17.

[8] P. Cottone, G. Lo Re, G. Maida, and M. Morana, "Motion sensors for activity recognition in an ambient-intelligence scenario," in *Proc. PerCom Workshops*, 2013, pp. 646–651.

[9] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Learning actionlet ensemble for 3D human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 914–927, May 2014.

[10] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3D points," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recog. Workshops*, 2010, pp. 9–14.

[11] L. Xia, C.-C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3D joints," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recog. Workshops*, 2012, pp. 20–27.

[12] L. Miranda, T. Vieira, D. Martnez, T. Lewiner, A. W. Vieira, and M. F. M. Campos, "Online gesture recognition from pose kernel learning and decision forests," *Pattern Recog. Lett.*, vol. 39, no. 0, pp. 65–73, 2014.

[13] H. Shum, E. Ho, Y. Jiang, and S. Takagi, "Real-time posture reconstruction for microsoft kinect," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1357–1369, Oct. 2013.

[14] R. Gupta, A. Y.-S. Chia, and D. Rajan, "Human activities recognition using depth images," in *Proc. 21st ACM Int. Conf. Multimedia*, 2013, pp. 283–292.

[15] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from RGBD images," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 842–849.

[16] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from RGB-D videos," *Int. J. Rob. Res.*, vol. 32, no. 8, pp. 951–970, Jul. 2013.

[17] B. Ni, Y. Pei, P. Moulin, and S. Yan, "Multilevel depth and image fusion for human activity detection," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1383–1394, Oct. 2013.

[18] B. Ni, G. Wang, and P. Moulin, "RGBD-HuDaAct: A color-depth video database for human daily activity recognition," in *Proc. IEEE Int. Conf. Comput. Vision Workshops*, Nov. 2011, pp. 1147–1153.

[19] S. Fothergill, H. M. Mentis, P. Kohli, and S. Nowozin, "Instructing people for training gestural interactive systems," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2012, pp. 1737–1746.

[20] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, Jun. 2012, pp. 1290–1297.

[21] C. Wolf, J. Mille, E. Lombardi, O. Celiktutan, M. Jiu, M. Baccouche, E. Dellandréa, C.-E. Bichot, C. Garcia, and B. Sankur, "The LIRIS human activities dataset and the ICPR 2012 human activities recognition and localization competition," Laboratoire d'Informatique en Images et Systmes d'Information, INSA de Lyon, Lyon, France, Tech. Rep. LIRIS RR-2012-004, 2012.

[22] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Berkeley MHAD: A comprehensive multimodal human action database," in *Proc. IEEE Workshop Appl. Comput. Vision*, Jan. 2013, pp. 53–60.

[23] P. Gehler and S. Nowozin, “On feature combination for multiclass object classification,” in *Proc. IEEE 12th Int. Conf. Comput. Vision*, Sep. 2009, pp. 221–228.

[24] (2012). [Online]. Available: http://pr.cs.cornell.edu/humanactivities/results.php

[25] L. Chen, H. Wei, and J. Ferryman, “ReadingAct RGB-D action dataset and human action recognition from local features,” *Pattern Recog. Lett.*, vol 50, pp. 159–169, 2014.

[26] A. A. Chaaraoui, J. R. Padilla-Lopez, P. Climent-Perez, and F. Florez-Revuelta, “Evolutionary joint selection to improve human action recognition with RGB-D devices,” *Expert Syst. Appl.*, vol. 41, no. 3, pp. 786–794, 2014.

[27] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.

[28] K.-B. Duan and S. Keerthi, “Which is the best multiclass SVM method? An empirical study,” in *Proc. 6th Int. Conf. Multiple Classifier Syst.*, N. Oza, R. Polikar, J. Kittler, and F. Roli, Eds. Berlin, Germany: Springer, 2005, pp. 278–285.

[29] L. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[30] S. S. Rao, *Engineering Optimization: Theory and Practice*, 3rd ed. New York, NY, USA: Wiley-Interscience, 1996.

[31] S. Arlot and A. Celisse, “A survey of cross-validation procedures for model selection,” *Statist. Surveys*, vol. 4, pp. 40–79, 2010.

[32] A. De Paola, G. Lo Re, M. Morana, and M. Ortolani, “An intelligent system for energy efficiency in a complex of buildings,” in *Proc. Sustainable Internet ICT Sustainability*, Oct. 2012, pp. 1–5.

[33] A. De Paola, M. Ortolani, G. Lo Re, G. Anastasi, and S. K. Das, “Intelligent management systems for energy efficiency in buildings: A survey,” *ACM Comput. Surv.*, vol. 47, no. 1, pp. 13:1–13:38, Jun. 2014.

[34] G. Lo Re, M. Morana, and M. Ortolani, “Improving user experience via motion sensors in an ambient intelligence scenario,” in *Proc. Pervasive Embedded Comput. Commun. Syst.*, 2013, pp. 29–34.

[35] A. D. Paola, M. L. Cascia, G. L. Re, M. Morana, and M. Ortolani, “Mimicking biological mechanisms for sensory information fusion,” *Biol. Inspired Cognitive Architect.*, vol. 3, pp. 27–38, 2013.

[36] (2014). [Online]. Available: http://www.dicgim.unipa.it/networks/ndslab/KARD/

[37] X. Yang and Y. Tian, “Effective 3D action recognition using eigenjoints,” *J. Visual Commun. Image Representation*, vol. 25, no. 1, pp. 2–11, 2014.

[38] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, “Real-time human pose recognition in parts from single depth images,” in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2011, pp. 1297–1304.

Salvatore Gaglio (M’78) received the Laurea degree in electrical engineering from the University of Genoa, Italy, in 1977 and the degree of “Master of Science in electrical engineering” from the Georgia Institute of Technology, Atlanta Georgia, U.S.A, in 1978. He is a full Professor of computer science and artificial intelligence with the University of Palermo, Palermo, Italy. His current research interests include the area of artificial intelligence and robotics.

Giuseppe Lo Re (SM’11) received the Laurea degree in computer science from the University of Pisa, Pisa, Italy, in 1990, and the Ph.D. degree in computer engineering from the University of Palermo, Palermo, Italy, in 1999.

He is currently an Associate Professor of computer engineering with the University of Palermo. His current research interests include the area of computer networks and distributed systems, broadly focusing on wireless sensor networks, ambient intelligence, and Internet of Things.

Marco Morana received the Laurea degree and the Ph.D. degree in computer engineering from the University of Palermo, Palermo, Italy, in 2007 and 2011, respectively.

He is currently a Postdoctoral Research Fellow with the University of Palermo. His current research interests include data fusion and reasoning in smart environments, intelligent data analysis for user profiling, and ambient intelligence.