J Ambient Intell Human Comput
DOI 10.1007/s12652-011-0065-z

ORIGINAL RESEARCH

Multi-sensor system for detection and classification of human activities

Roberto Ugolotti · Federico Sassi · Monica Mordonini · Stefano Cagnoni

Received: 23 January 2011 / Accepted: 29 July 2011

Abstract This paper describes a novel system for detecting and classifying human activities based on a multi-sensor approach. The aim of this research is to create a loosely structured environment, where activity is constantly monitored and automatically classified, transparently to the subjects who are observed. The system uses four calibrated cameras installed in the room which is being monitored and a body-mounted wireless accelerometer on each person, exploiting the features of different sensors to maximize recognition accuracy, improve scalability and reliability. The algorithms on which the system is based, as well as its structure, are aimed at analyzing and classifying complex movements (like walking, sitting, jumping, running, falling, etc.) of potentially multiple people at the same time. Here, we describe a preliminary application, in which action classification is mostly aimed at detecting falls. Several instances of a hybrid classifier based on Support Vector Machines and Hierarchical Temporal Memories, a recent bio-inspired computational paradigm, are used to detect potentially dangerous activities of each person in the environment. If such an activity is detected and if the person "in danger" is wearing the accelerometer, the system localizes and activates it to receive data and then performs a more reliable fall detection using a specifically trained classifier. The opportunity to turn on the accelerometer on-demand makes it possible to extend its battery life. Besides and beyond surveillance, this system could also be used for the assessment of the degree of independence of elderly people or, in rehabilitation, to assist patients during recovery.

Keywords Multi-sensor systems · Hierarchical Temporal Memory · Support Vector Machines · Human activity monitoring · Fall detection

R. Ugolotti · F. Sassi · M. Mordonini · S. Cagnoni
Department of Information Engineering, Intelligent Bio-Inspired Systems (IBIS) Laboratory, University of Parma,
Via G.P. Usberti 181/a, 43100 Parma, Italy
e-mail: rob_ugo@ce.unipr.it

M. Mordonini
e-mail: monica@ce.unipr.it

S. Cagnoni
e-mail: cagnoni@ce.unipr.it

F. Sassi (✉)
Henesis s.r.l., Viale Dei Mille, 108, 43125 Parma, Italy
e-mail: fsassi@ce.unipr.it; federico.sassi@henesis.eu

1 Introduction

Over the past few years the technological advances, the miniaturization and diffusion of devices and sensors (Monekosso et al. 2009) and the increase of CPU performance have stimulated research on intelligent environments, with particular regard to monitoring people and their interactions with the surrounding environment. This work is about the observation and classification of activities of people, aimed at assisting them by detecting potentially dangerous situations.

Many techniques can be used to analyze human activities. A large number of such systems are exclusively based on cameras (Ji and Liu 2010; Turaga et al. 2008), while other systems use different kinds of sensors, like wearable (e.g. accelerometers, Kern et al. 2003; Khan et al. 2010; Zhu and Sheng 2009) or microphones (Ward et al. 2006) or passive ones (Rutishauser et al. 2005). On the one hand, ambient sensors, like cameras, have the advantage of being able to monitor multiple subjects at the same time without requiring that they wear any special devices. Nevertheless,

Published online: 21 August 2011

their main drawback is the need for a structured environment equipped with at least one calibrated camera per room. Moreover, analyzing images may be very complex, depending on lighting conditions, furniture placement, position of cameras or crowdedness of the environment.

On the other hand, wearable sensors, like wireless accelerometers, have several advantages over cameras: they are person-specific, while the sensed data are independent of the conditions of the environment, as well as are usually available in a format which is easier to process. However, while they need a less structured environment than other sensors, just requiring a non-calibrated receiver within the transmission range, they also impose that every subject being monitored wear the sensor. Another important downside is the need of frequent service to replace exhausted batteries.

Recently, there has been great interest in developing architectures that use different kinds of sensors, in order to acquire more robust data and to overcome the drawbacks of single-sensor architectures, such as sensitivity to device failures due to their physical limits or environmental problems. For example, Lester et al. (2005) use a very small sensing unit that includes eight different sensors, Leone et al. (2008) try to detect falls using three independent systems, based on a 3D time-of-flight range camera, an accelerometer and a microphone, respectively. However, neither system aims at detecting human activities using visual and wearable sensors at the same time.

Another possible field of research is the creation of an intelligent ambient which understands people's actions and behavior to assist them in accomplishing daily routine tasks. For instance, the work of Rashidi et al. (2011) exploits a highly sensorized ambient, but without cameras, to assess people's behaviors in a very detailed way.

The general, long-term goal of our research is to design intelligent systems which, while not interfering with everyday life's activities, are able to detect events in a timely manner and to provide reliable predictions by which traumatic events may be anticipated and avoided. The system we describe could also be used for monitoring elderly people in an institutional care facility to detect possibly dangerous events or to assess the degree of independence of patients during recovery.

In this paper, we focus on a preliminary application of such a system, mostly aimed at detecting falls. In particular, the architecture we present relies on two different action detectors: the former is based on four calibrated cameras, while the latter uses data acquired by a body-mounted accelerometer.

The main software component of this system is a hierarchy of classifiers composed by a Hierarchical Temporal Network (HTM) (Hawkins and Blakeslee 2004) and a Support Vector Machine (SVM) (Hsu et al. 2003) applied to each sensor in the system: one for each camera and one for the accelerometer.

The Hierarchical Temporal Memory computational model takes inspiration from the structure and functional properties of the neocortex and derives from a more general theory, called Memory-Prediction Framework (Hawkins and Blakeslee 2004). A HTM can learn "invariants" when exposed to spatial-temporal data. In this particular application, during the training phase, the HTM learns invariant movement sequences while, during the inference phase, it recognizes movements by matching them with the patterns it has stored. The HTM receives as input data representing movements and computes as output the probability distribution of the match between them and its memory content.

SVMs are a set of methods for supervised learning mainly used for classification purposes. They minimize the empiric error while, at the same time, maximizing the geometric margin between different classes (Burges 1998). In our system the SVM processes the HTM output in order to classify it into a number of categories. A more detailed presentation of these techniques will be provided in the following section.

The computational architecture of the system integrates data acquisition from the sensors with classification using a multi-thread parallel approach to minimize its response time. The system relies on the specific features of different sensors to maximize classification accuracy and minimize interference with ordinary life. In particular, the system continuously analyzes the output of the cameras to detect potentially dangerous activities and, when this happens, selectively turns on the accelerometer of a specific person. This permits to extend the wearable sensors' battery life and to lower the service cost of the system without sacrificing accuracy or coverage.

The paper is organized as follows: in Sect. 2 we introduce the classification methods and algorithms while, in Sects. 3 and 4, we describe the two main modules by which our architecture is composed, with details on the hardware setup, the algorithms and the experimental results which have been obtained in our tests. In Sect. 5 we describe how the different modules are integrated into a reliable fall detection system and, finally, in Sect. 6 we draw some conclusions and describe future work.

2 Classification hierarchy

2.1 Hierarchical Temporal Memory

The Hierarchical Temporal Memory is a computational paradigm inspired by the internal structure of the mammalian neocortex. A Hierarchical Temporal Memory is

structured as a hierarchical network of nodes where the sensory data enter at the bottom level while the outputs of the network are output by the top nodes, which represent the possible "causes" of the input. As Mountcastle first described in Mountcastle (1978), the various parts of the neocortex are topologically very similar, so it is very likely that the whole neocortex implements a single common algorithm running at different levels of a hierarchy of processes. In agreement with this theory, every node in a HTM network runs the same algorithm, regardless of its location or of the problem the network has to deal with.

Every node in a Hierarchical Temporal Memory network (George and Jaros 2007) detects repeating spatial-temporal patterns (invariants) in its input and groups them together as "causes". A HTM network is trained in an unsupervised manner, in the classical meaning, but time is considered to be an implicit supervisor: if two events (inputs) often occur consecutively, it is very likely that they belong to the same "cause" (output). As in other connectionist approaches, nodes at the bottom of the hierarchy receive inputs from the sensors and their output becomes the input of the nodes that are located in the layer above them. Nodes on lower levels receive input from a limited set of the whole sensory data, while nodes on higher levels have a broader "view" of the sensory input, as processed and synthesized by the other nodes. This means that lower-level nodes find causes belonging only to a limited time and input (space) scale while higher-level nodes are able to find causes at a larger scale. In other words, every node has an internal representation of possible "causes" (spatial-temporal events) and its output is an array whose elements represent the probability of the corresponding "cause" to be active. For instance, if a node's memory is composed by five "causes", a node's output [0.6,0.2,0.1,0.9,0.3] means that at the given time, the fourth "cause" is the most likely to be happening, but even the first one may be active with a lower probability; a similar example may be found in Fig. 1. According to this model, the output of the top level node can be assimilated to the distribution of the probability that the current input corresponds to the causes memorized in such a node.

Figure 1 represents a general HTM network where the sensor nodes are in the bottom layer and the signals flow from them upwards to the top node through the network.

To assess the classification performance of a network over a set of data, as provided by the output node of a HTM, it is necessary to add a supervised classifier (like a SVM or a K-Nearest Neighbors classifier) on top of it. When used with this setup, it is possible to consider the HTM as a spatial-temporal filter applied to the data. These characteristics make HTMs a very powerful technology while, at the same time, causing their performance to be affected on the quality of data used to train them, as happens with virtually all machine learning techniques.

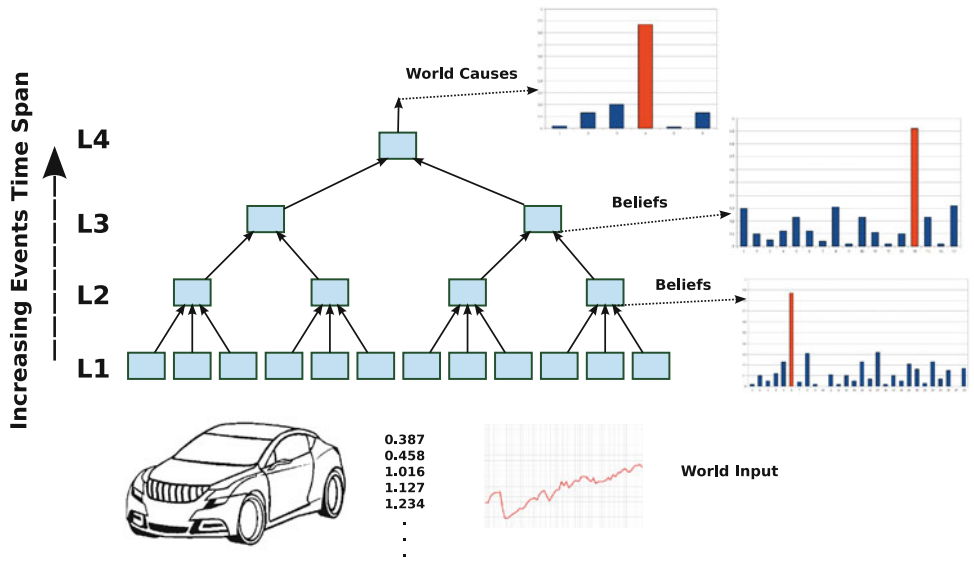

Fig. 1 A HTM network. The network is composed of a hierarchy of nodes. The nodes on level L1 receive data from the world. These data can be of any type, the only mandatory request being that they represent events that develop over time. The L1 nodes look for invariants in the received data and output a probability distribution that represents the "beliefs" learned by the HTM algorithm. The L2 nodes, as every other node in the network up to the top level, perform the very same algorithm on the "beliefs" received from all the children. The time span and the complexity of the events analyzed by the network increases with the level in the hierarchy. This means that, while lower levels learn small portions of the events input to the network, higher levels integrate this information to generate a more complete knowledge of the world. The last level's beliefs, termed "causes of the world", represent the "explanation" that the network gives about the current input

Hierarchical Temporal Memories forward propagation of signals up in the hierarchy is very similar to Bayesian Belief Propagation, but HTMs add spatial and temporal clusterings and the hierarchical structure, similar to connectionist models.

The training of a HTM network proceeds sequentially level by level, from the bottom to the top level. Once the nodes on the lowest level have been trained, they switch to inference mode and the nodes on the second level start their learning phase using the results of the inference of the nodes below. These operations are repeated until all levels in the network are fully trained.

Suppose the nodes on the first level have been trained. The outputs of these nodes corresponding to the patterns in the training set then become the input of the nodes at the second level. These nodes will now perform spatial and temporal operations in order to form new groups. These groups are a composition of low level temporal groups, forming a "sequence of sequences", that represents an event over a larger time scale. In this way, going up the HTM's hierarchy, the inputs are seen at a larger time scale until, at the top of the network, the output of the HTM allows a classifier layer to reach a decision about which "cause" generated the inputs.

A basic HTM node is modeled as comprising two parts: the SpatialPooler and the TemporalPooler. Every part of the node must be trained before being able to perform its computation.

The SpatialPooler quantizes multi-dimensional input data in order to reduce noise and dimensionality. The SpatialPooler extracts a limited set of "quantization centers" (coincidences) from the data during the training phase. After training (during the inference phase) such nodes describe each input pattern as a probability distribution associated to each quantization center.

The TemporalPooler receives in input the output of one or more SpatialPoolers and groups together coincidences that are likely to occur over time during the training phase. This node creates a temporal adjacency matrix (TAM) that stores the transitions between coincidences received as input. A state is defined by the current coincidence and the transition from the previous ones. The number of states is not known a-priori and depends on the dataset and on the processing done at the lower levels of the hierarchy. The more the data is time-dependent, the more states will be created. After observing all the training data, the TAM is full and the "grouping algorithm" computes state chains composed by consecutive temporal states, to form groups. Each group then represents a different "cause" in the input data. After the training phase the number of states and their connections (groups) are fixed.

During the inference phase this part of the node maps each occurring coincidence to the corresponding state and then assigns a likelihood score to the possible sequences (groups) that are currently active depending on the past history and the current state.

The fundamental paper in which the HTM model and its biological mapping are extensively discussed is the work of George and Hawkins (2009).

HTMs have been successfully used in visual recognition problems (Csapo et al. 2007; Farahmand et al. 2009), human gesture recognition (van Doremalen and Lou 2008) and activity classification tasks (Zhang et al. 2008).

2.2 Support Vector Machines

A SVM receives as input a vector of data that is mapped, using a kernel function, to a multi-dimensional space (hyperspace). Then, all data belonging to the same category are separated from data belonging to other classes by hyperplanes defined by the so-called "support vectors". Finally, the set of hyperplanes that maximizes the distance between the class-delimiting hyperplanes is computed. Therefore, every SVM is defined by its support vectors. In the training phase, the SVM maps data to the hyperspace and searches for an optimal partitioning. After this stage, the classification of a new element is made by mapping it to the hyperspace through the kernel function to determine the region of the hyperspace to which it belongs (Hsu et al. 2003). In the last years, SVMs have become a standard classification technique and their ability to classify human motion and activities has been successfully demonstrated (Kellokumpu et al. 2005; Kim and Ling 2009; Niebles and Fei-Fei 2007; Schuldt et al. 2004; Wu et al. 2005).

2.3 Hybrid HTM/SVM classifier

The output of a HTM network's top node is not different from the signals that flow inside the network. In order to map these "causes" to a known number of categories, a supervised classifier is needed. This step is necessary because the HTM learning algorithm is unsupervised and the number of groups formed may not always be directly related to the desired categories.

We classify the HTM outputs adding a SVM as a supervised top layer, which means that the input that the SVM has to process is a vector of floating-point numbers that represent the "causes" found by the underlying HTM network. The SVM clusters these "causes" and maps them onto the categories considered for the input patterns. Therefore, a HTM can be considered a pre-processing filter which adds temporal information with respect to a "stateless" SVM-based classification.

It has been demonstrated that a combination of HTM and SVM works much better than a single SVM,

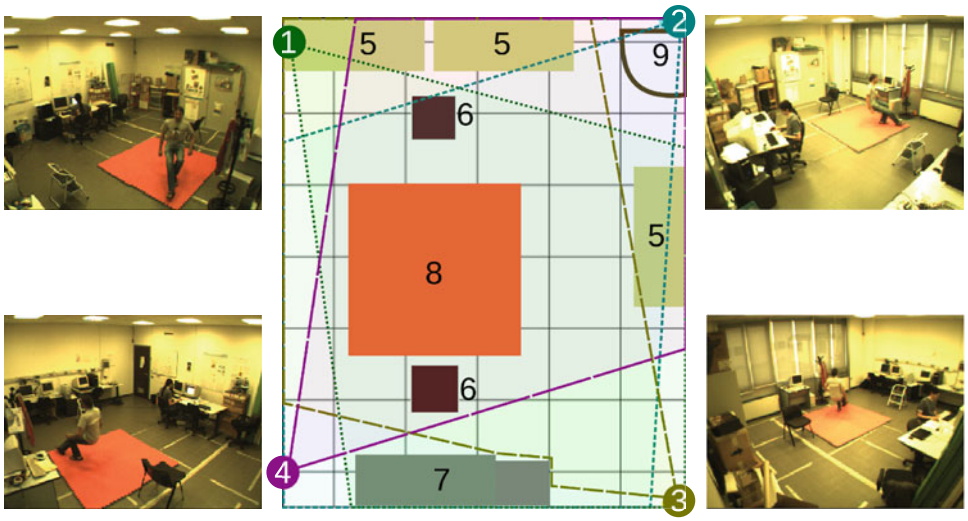

Fig. 2 Floor plan of the laboratory environment where our experiments took place. The *four circles* (numbers *1–4*) at the corners of the room represent the cameras and the corresponding *lines* represent their fields of view. For every camera, the figure shows an instance of a captured frame. Desks, low-height obstacles, are denoted by number *5*; number *6* are the chairs where the subjects may sit; number *7* are *closets*, high obstacles that limit the visual field; number *8* is a "tatami" to soften the falls of the subjects; number *9* is the door

particularly in the presence of noisy signals (Sassi et al. 2009) or missing values.

In the following two sections we introduce the two subsystems that compose the multi-sensor architecture finally presented in Sect. 5.

3 Camera-based system

The first subsystem is based on four cameras but can be extended to a higher number of cameras. The system is designed to operate in an indoor environment (even if it is possible to apply it also to outdoor scenes), such as a room. In particular, the room we used for implementing and testing the system is 5.4 m wide and 6.3 m long. A representation of the environment is shown in Fig. 2.

3.1 Hardware setup

The four cameras (Firefly MV-03MTC by Point Grey[1] equipped with lenses with an aperture of F1.4 and focal length of 2.8 mm) are located in the four corners of the room at an average height of 2.3 m. This placement has been chosen in order to extend as much as possible the field of view of the cameras without distorting too much the images of the people being monitored. The cameras are connected to a central server via a FireWire IEEE-1394 S400 bus. Because of the limits imposed by the signal bandwidth, every camera can capture images having a resolution of 640 × 480 pixels at a frequency of 15 fps per camera.

The calibration of the four cameras is an important preliminary step. This phase is absolutely required for merging different views of the same person correctly, especially if several people are in the room. The Camera Calibration Toolbox for Matlab developed by Jean-Yves Bouguet[2] was used in this phase. After this operation, the relative position of the cameras with respect to the room is fully determined.

3.2 Image processing

The goal of the image processing step is to identify and track people over time and extract movement descriptors that can be used by a classifier to detect their actions. The operations are organized into a modular system, in which the images from every camera are analyzed independently in order to improve performance and make the system more robust to single camera failures or occlusions. After this, the results are merged to obtain a unique decision.

The image processing step is schematized in Fig. 3 and comprises the following operations:

1. background computation using a running average and background subtraction (the result of the operation is shown in Fig. 3b) obtained by computing the difference with the background. We use HSV color coordinates for independence from lighting conditions (Cucchiara et al. 2001);
2. image segmentation, based on the result of the previous step, to compute the number and positions of the people in the room and remove objects that are definitely not humans, according to their size (Fig. 3c);
3. people tracking, based on a distance criterion between consecutive frames, to follow the subjects one frame after another and keep track of their identities;

[1] http://www.ptgrey.com/products/fireflymv/fireflymv.pdf equipped with lenses with an aperture of F1.4 and focal length of 2.8 mm.

[2] The calibration toolbox is available at http://www.vision.caltech.edu/bouguetj/calib_doc.

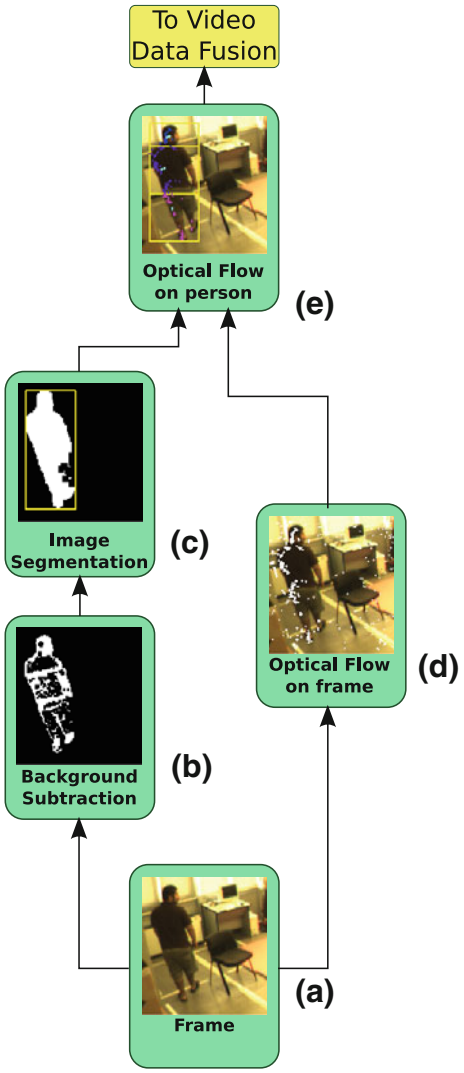

Fig. 3 Image processing schema: **a** the image captured by the camera, **b** background subtraction, **c** image segmentation, **d** optical flow on the whole image, **e** optical flow computed on a person

4. optical flow calculation on the whole image, using the Lucas-Kanade algorithm (Shi and Tomasi 1994) to detect and follow the movements that occur in the scene (Fig. 3d);
5. finally, the visual and positional information, obtained by background subtraction, segmentation, and optical flow computation, is merged. Only the pixels belonging to people are considered (Fig. 3e). The images of the people who have been identified in step 2 are divided in three parts (head, torso, legs) as shown in Fig. 4. For these three parts, the average shift with respect to the previous frame is calculated.

For every camera, the results of the image processing step are the number, position and movement representation (by optical flow) of all the people detected in the room. The

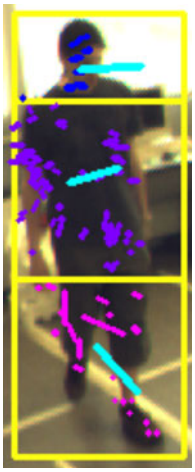

Fig. 4 Representation of a person. The three parts are visible: head (the upper 20% of the whole body), torso and legs (middle and lower 40%, respectively). For every part, the *light lines* show the shift with respect to the previous frame, magnified 10 times for better visibility

following step consists of merging the information obtained by the four cameras, in particular the data related with the same person in the different views. The lowest pixel of each person's image is found and is assumed to be the projection of the person on the floor. Then, thanks to cross-calibration, the 3D coordinates of this point in the room are determined. The four projections are associated to the corresponding subjects according to a proximity criterion. Obviously, each subject is not always visible to all four cameras, for example when he/she is hidden by another person or object. In any case, this affects mainly the classification phase, because the classifier has fewer data to process.

The results of the optical flow computed based on the images from the (up to) four cameras are the data that the video system must classify.

The raw data output of the optical flow consists of the displacements of each subject's head, torso and legs with respect to the previous frame. The encoding we propose is composed by two steps:

- conversion to polar coordinates: providing the classification system with a direct measure of direction and speed of movement (corresponding to the angle and intensity coordinates) is definitely more relevant and immediate than dealing with their cartesian projections;
- quantization, to prevent the search for invariants performed by HTM's SpatialPoolers from being influenced by small data variations and outliers. This pre-processing step is not mandatory but is advisable to simplify classification and make it more controllable.

After this pre-processing phase data samples related to each body part are encoded as two integers, of which one (in the range [0, 5]) represents the speed of the movement

and the other (in the range $[-2, 2]$) represents the direction. These values are then input into the classifier.

3.3 Image classification system

The image classification system has been designed to be flexible with respect to the number of the available cameras and of people in the environment.

The basic classification unit, called "camera-classifier", processes each subject's data coming from one camera (see Fig. 5 on the left). This unit receives in input six integers that represent the speed and direction of movement of the head, torso and legs of the person. It is composed by a two-layer Hierarchical Temporal Memory having a SVM on top of it. The HTM's nodes at the lower level look for sequences of basic movements of each body part separately, while the node at the second level merges this information to create sequences of full body movements at a longer time scale. In particular, in our experiments, the node on the first level found 30 groups for the head and torso, and 25 groups for the legs. On the second level, the HTM found 150 groups describing composite movements.

The SVM on top clusters the HTM's output (probability distribution over the groups) into the following categories of human actions: walking, standing, sitting, getting up from a chair, falling, and rising up from the floor. This "camera-classifier" (CC) is trained using the data coming from all four cameras in the experimental environment, to make it independent of the view.

Every time a new subject is detected on the scene a new "*personal*-classifier" is created. It is composed by four CCs (one for each camera), finally merged and processed by a SVM (see Fig. 5). This SVM acts as a "settler" that computes the final movement classification based on the information received from the underlying classifiers. This "*personal*-classifier" must be immune to partial lack of data, in the sense of being able to take the correct decision even if a person is not visible to one or more cameras and, consequently, there is no output from the corresponding CCs. Experimental results showed that this solution offers good performance from this point of view.

The experimental setup includes four cameras; accordingly, this classifier is composed by four CCs. The classifier analyzes temporal information about the movements, relying on the system's tracking algorithms in order to always feed data from the same person to the corresponding "*personal*-classifier". Every time a new person appears on the scene a new "*personal*-classifier" is instantiated, which is destroyed when the person exits. This modular architecture lets the system achieve a high parallelization degree and enables it to work in real time with multiple subjects on a standard computer. More details on the implementation are presented in Sect. 3.

If a higher, or lower, number of cameras is desired it is sufficient to retrain the SVM layer (and not all the CCs) to handle the new set of lower-level classifiers.

3.4 Experimental results

Tests were performed using as a server a quad-core computer based on an Intel i7 2.67GHz processor, equipped with 4 GB RAM and a Nvidia FX 5800 Quadro graphics card.

The dataset used for the experiments comprises data acquired from 11 volunteers, 10 males and 1 female, who continuously performed several actions in the sensorized room, one at a time. Each subject performed all actions sequentially in a unique video stream. The durations of the video sequences depend on the time needed by the subject to perform all the requested actions and vary from 90 to 200 s. Every subject performed actions in no specific order. Obviously, since not all events are equally long (e.g. a walk usually lasts longer than a fall), some events occupy more frames than others. Data from four subjects were used as training set for the classifiers, while the other data were used as test set. Even if the system is designed to handle

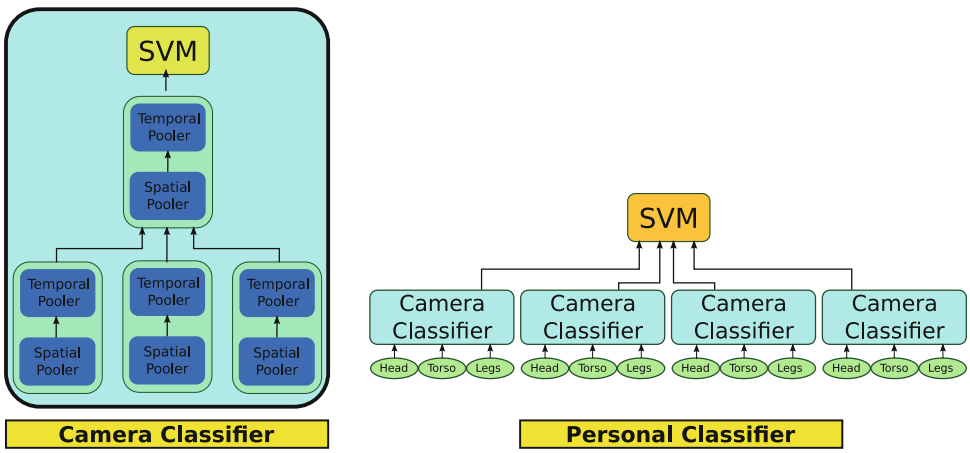

Fig. 5 Schema of the complete classifier used in the camera-based system (*right*). The blocks named "Camera Classifier" are instances of the hybrid HTM/SVM classifier shown on the *left*

Table 1 Percent classification accuracy of the camera-based system

Action/class	Stand	Walk	Sit	Get up	Fall	Rise	n.r.
Stand	83.7	0.0	0.0	4.1	2.0	6.1	4.1
Walk	3.5	91.2	0.0	0.0	0.0	3.5	1.8
Sit	9.5	0.0	76.2	0.0	0.0	0.0	14.3
Get up	0.0	0.0	4.8	71.4	0.0	4.8	19.0
Fall	0.0	4.8	4.8	0.0	85.6	0.0	4.8
Rise	0.0	0.0	0.0	5.0	0.0	70.0	25.0

Confusion matrix computed on the test set

more than one person at a time in the same room, all the tests presented in this work deal with a single person at one time. This restriction does not permit to finely verify the performance of the system in case of occlusions or bad tracking. Nevertheless, this kind of degradation may be modeled as the lack of one or more cameras.

The training of a "*personal*-classifier" classifier consisted of two training phases:

- the CCs were trained using the images from all cameras;
- the SVM in the "*personal*-classifier" was trained using the outputs of the four CCs.

Tables 1 and 2 report some of the results which were obtained on the test set, in classifying actions and in detecting falls.

For each frame in a video sequence, a classification was generated for each subject. The classification results were averaged over time windows 0.5 s long to avoid errors due to isolated misclassified samples. Each window was finally assigned the label corresponding to the event which had been most frequently detected within it. Table 1 shows the results of action recognition. The last column of the table reports events which were not recognized. An event was considered to include all time windows it occupies and was considered to have been recognized when more than 50% of its time windows had been assigned to the same class.

Table 2 shows the results of fall detection. A fall was considered to have been detected when at least one of the windows that compose the event was classified as a fall. This different behavior was chosen because of the shortness and criticality of this kind of event. To adapt to this detection criterion, we subdivided the test data into events differently for this test, even if the data included in the test set were the same. This is why the test appears to have been made on a different number of events.

Experimental results show that this system obtains good results in recognizing and tracking people and in classifying their actions, but produces some false alarms in fall detection. Results get worse when the number of available cameras decreases. Table 3 reports event classification accuracy versus number of cameras. These results have been obtained by re-evaluating the same video sequences deprived of the data coming from one or more cameras; this can be considered a simulation of what happens when a camera loses the subject because of bad tracking or occlusions. The classification results degrade significantly only when two or more cameras miss the subject.

Table 2 Percent results of the camera-based system

Event	Not fall	Fall
Stand	100	0
Walk	94.7	5.3
Sit	95.5	9.5
Get up	100	0
Fall	0	100

Fall detection problem on the test set

Table 3 Percentage of correct event classifications versus number of cameras

No. of cameras	Correct classification percentage
4	82
3	75
2	59
1	26

4 Accelerometer-based system

The second subsystem is based on a body-mounted low-cost wireless module, incorporating a triaxial high resolution accelerometer and an IEEE 802.15.4 RF transmitter (a.k.a. MAC level of a ZigBee network). The wireless sensor module was entirely designed and developed by Henesis s.r.l. and is being used for distributed sensing within the Henesis WISnP.[3] Figure 6 shows the module mounted on a chest-band.

[3] Wireless Sensor Network Platform: see http://wisnp.henesis.eu.

Fig. 6 Wireless accelerometer module developed by Henesis s.r.l

In this implementation of the system we use only one accelerometer per person firmly attached to the lower end of the sternum of the person to be tracked. This location has been chosen because it is very near to the body center of mass, and provides a good approximation for the movement of the whole person. The sampling rate of the wireless module has been set to 160 Hz, its range being ±2G and its resolution 1 mG. Figure 7 shows a plot of the acceleration data acquired during a walking sequence of about 3 s.

In order to make the system able to deal with any subject, regardless of body shape, height and type of motion, the data must be as generic as possible. For this reason, we do not use a kinematic approach, since this would involve the definition of a complex and specific model that, consequently, would impose limitations and constraints. Instead, we directly send the acceleration data to a central server which classifies them using a hybrid HTM/SVM classifier, shown in Fig. 8, which closely resembles the one used for video-based classification. This classifier searches for invariants in the acceleration data without relying on a-priori knowledge.

Every input of the classifier is an array which stores the acceleration along the three axes. When the accelerometer is placed correctly and the subject stands still (calibration) the array elements (x, y, z) must be (0, 1, 0) because of the gravity force along the y axis. Like the HTM used in the

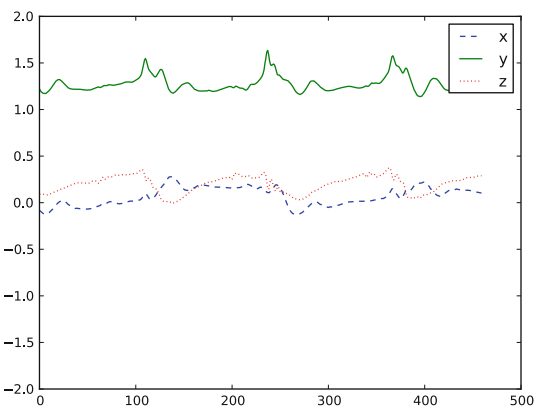

Fig. 7 Walking sequence recorded via the 3-axial accelerometer

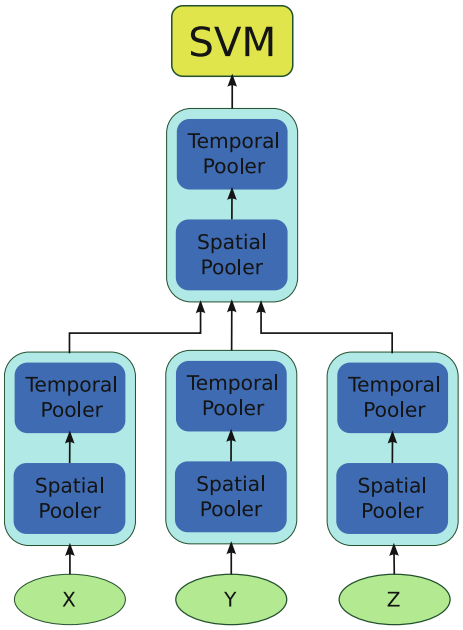

Fig. 8 The classifier used for processing accelerometer data

camera-based system, this HTM network is composed by two layers (see Fig. 8):

1. the first layer includes three nodes (each composed by a SpatialPooler and a TemporalPooler). Each of them analyzes one axis separately, so every node on this level can learn and recognize patterns occurring in time only along the corresponding axis. In this particular case we know that the range of data that every SpatialPooler receives as input is ±2G. Therefore, we did not use training data to set the quantization center, but we set them so that the range would be uniformly sampled;
2. the second layer receives as input the output of the three nodes in the first layer to learn invariants related to full body movements at a larger time scale and sends its results to the SVM.

Finally, the SVM classifies the output of the HTM. Every sample is classified independently of the previous one because the temporal information is already taken into account by the HTM.

The events this system has been trained to detect are: standing up, walking, sitting down, getting up from sitting, falling, rising up from the ground, sitting, and lying down. The dataset includes accelerometer recordings acquired during the same events which were included in the training set for the video system described in Sect. 4.

Table 4 shows the results of event recognition, which appear to be in general much less accurate than those obtained by the video-based system.

However, this system has very good performance in detecting falls. If it is used as a fall detector according to the

Table 4 Percent results of the accelerometer-based system

Action/class	Stand	Walk	Sitdown	Get up	Fall	Rise	Sit	Lie	n.r.
Stand	25.0	58.3	0.0	0.0	0.0	0.0	16.7	0.0	0.0
Walk	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Sit down	0.0	9.5	47.6	19.1	0.0	23.8	0.0	0.0	0.0
Get up	0.0	14.3	33.3	47.6	0.0	4.8	0.0	0.0	0.0
Fall	0.0	0.0	0.0	0.0	90.5	9.5	0.0	0.0	0.0
Rise	0.0	0.0	5.0	0.0	25.0	55.0	0.0	5.0	10.0
Sit	14.3	38.1	0.0	0.0	0.0	0.0	42.8	0.0	4.8
Lie down	0.0	6.2	0.0	0.0	25.0	12.5	6.3	50.0	0.0

Event recognition on the test set

Table 5 Percentage results of the accelerometer-based system

Event	Not fall	Fall
Stand	100	0
Walk	100	0
Sit	100	0
Get up	100	0
Fall	0	100

Fall detection on test set

same classification strategy used for the vision-based system, it obtains a perfect score on our test set, as shown in Table 5. It can be seen that this system works better than the previous one on this particular problem. Data were organized as for the corresponding test of the vision-based system.

The experimental results highlight how, in this case, fall detection is more accurate than with the previous system. However, this solution has some "practical" drawbacks that hamper its use in a real-world environment:

- an accelerometer which continuously transmits data requires a considerable amount of power and has a short battery life;
- the data rate needed for a reliable movement classification requires a large bandwidth, which allows just one accelerometer to send data at one time.

In the following section we present an architecture that integrates the two systems to overcome these problems.

5 Multi-sensor system

This section describes a multi-sensor architecture, obtained by combining the video-based and accelerometer-based systems presented in the previous sections. The main goal of this new architecture is to create a real-time and reliable fall detector and to overcome the drawbacks of the single-sensor systems, in particular:

- the dissatisfactory results of the camera-based system in recognizing falls;
- the physical limits of the accelerometer-based system, like battery life and limitations in the number of subjects which can be monitored at the same time.

When analyzing performance of a multi-sensor system it is necessary to correctly match the data coming from different sensors, in order for it to be able to coherently analyze the behavior of a person when all systems are active at the same time.

In our hybrid architecture, the video system constantly monitors the actions of all the people in the room and turns on the accelerometer of a specific person only when a possibly dangerous event, which may drive to a fall, has been detected.

To achieve this goal, a new subsystem, named "Coordinator", has been created. The new multi-sensor architecture is represented in Fig. 9. The goal of the "Coordinator" is to keep the correct association and synchronization between the subjects identified by the video system and the corresponding accelerometers.

To properly merge the two subsystems the cross-localization of the people in the room is necessary. It could be easily achieved if, ideally, both systems were able to localize each subject within the room correctly. As shown before, the camera-based sub-system already includes an accurate tracking method which can localize people in the room. Regarding the accelerometer-based system, in the following section we shortly introduce the problems which must be tackled in localizing radio sources in a small environment as a room, before describing how we implemented the "Coordinator".

5.1 Localization based on wireless transmission

The localization of a wireless node has been vastly examined in the literature (e.g. in Grossmann et al. 2007) but, at present, no robust method for reliably computing a position has been designed yet. Most techniques are based

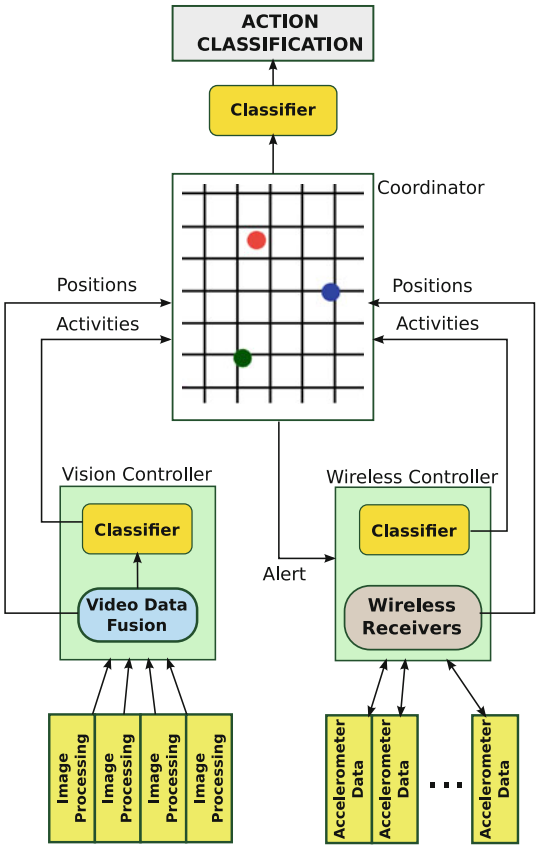

Fig. 9 Multi-sensor architecture

on a measure of the intensity of the signal coming from the node that must be localized. The main problems in doing so are:

- non-uniformity of the antenna's radiation pattern, particularly when the node is mounted on the chest of a person, as shown in Lihan et al. (2008);
- non-univocal localization due to the limited number of receiver stations available;
- instability of the received power measures over time, especially when the person is moving.

The first problem is caused by the body absorption of the transmission power and by the location of the receivers: if they are located behind the subject's back they underestimate the power. In fact, the orientation of the person significantly affects the power measured by the receivers.

A possible solution is introduced by Luo et al. (2010), where the authors create a map of the environment by measuring the power of the signals received from a limited number of locations in the room, arranged as a grid. After this preliminary step, the power of the node's signal is detected by the receivers and it is matched to this map to locate the RF source position.

Awaiting further advancements in this field, and to be able to make a first evaluation of the global performance of our system, we resorted to a preliminary implementation, in which an omni-directional antenna has been placed on the head of the monitored people and a grid localization system like the one used in Luo et al. (2010) has been employed. This solution makes the system immune to orientation and body absorption issues.

To tackle the other two main problems it is possible to rely on the localization performed by the video system. The wireless localization has to determine which person in the scene wears which accelerometer, so it must select from a very limited set of positions in the room. The simplest approach consists of selecting the best matches between the positions estimated by the wireless localization system and the ones estimated by the video system.

The remaining issue may be solved in a similar way. For instance, it is possible to exploit the tracking algorithms already present in the video system by establishing a correspondence between a person and an accelerometer when the person is still and both localization systems are reliable, and then keeping the association using the tracking capabilities of the video-based system when the person is moving.

To evaluate the raw performance of the wireless localization we performed some tests to discriminate between two people standing at different distances (2, 3, 4 and 5 m). Table 6 summarizes the results.

These tests show that the accuracy improves greatly when the distance between the subjects increases and that, at distances greater than 3 m, this information become very reliable.

5.2 Multi-sensor cooperation

Operatively, in this new architecture, the body-mounted wireless sensor transmits a beacon (presence signal) every second. This signal is received by other modules installed close to every camera in the room (as shown in Fig. 10) and connected to the server via a RS232 interface. Based on the power of the beacon received by the four boards, the system is able to estimate the location of the person who wears the accelerometer. Then, comparing these positions

Table 6 Percentage of correct localizations of two people versus distance

Distance (m)	Correct	Wrong
2	53	47
3	70	30
4	83	17
5	100	0

Fig. 10 Multi-sensor components: camera with a wireless receiver on top

with the ones calculated by the visual system, a correspondence is created for every person who wears an accelerometer. This way, the multi-sensor system can track people's actions using only the visual system, and setting all the wireless accelerometers in "idle mode". In this operation mode the accelerometer continuously senses and locally stores data in a circular buffer (up to 2 s) only when it detects any movement. For instance, if a person is completely still the sensor will only send the presence beacon. This operation allows one to extend battery life significantly with respect to continuous transmission of data.

When the visual system detects a possibly dangerous situation, the Coordinator sends a signal to the accelerometer worn by the subject in danger, and this starts sending data (including the ones already stored in the circular buffer) until stopped. These data are fed to the classifier shown in Sect. 4 in order to perform an accurate fall detection. If the subject appears not to be wearing an accelerometer, only the video system is used for classification.

The decision to switch on the accelerometer is taken according to the classification of the action returned by the camera-based system. Every event recognized by the camera-based system is associated to a weight that depends on the potential dangerousness of the corresponding action. If the weighted sum of the events in the last half second is greater than a threshold set experimentally, the accelerometers are switched on. Our goal is to switch on the accelerometers in all potentially dangerous situations and in situations that are possible causes of a fall, like standing up, sitting down and, obviously, falling. A useless activation is definitely preferable than missing a fall. Table 7 reports the weights assigned to the different actions.

Table 7 Weights of actions based on dangerousness

Action	Weight
Standing	0
Walking	0
Sitting	1.25
Standing up	0.75
Falling	2.5

5.3 Multi-thread architecture

During the integration of the two systems, two problems were detected, at the implementation level:

- a very high number of operations are needed, and it is impossible to run all the activities sequentially in real time;
- the interaction between the various components of the system requires a robust information exchange mechanism used to receive data from cameras or from wireless modules, switch accelerometers on and off, etc.

For these reasons, a multi-thread architecture has been created. Several optimization strategies were tested to find which architecture would produce the best results. An example of the interaction of the various components of the system may be found in Fig. 11. The architecture we implemented is composed of the following threads:

- four threads (one per camera), named *Readers*, receive images from one camera and store them in a specific buffer;
- four threads (one per camera), named *Elaborators*, process image data acquired by *Readers*. Image processing operations are executed sequentially, for each camera, even if they could have been partially parallelized, since the overhead caused by creation and maintenance of threads would have exceeded the advantages of using parallelism. The results of the processing (number, position and movement for every person seen by a camera) are stored in four other buffers;
- one thread, *Main*, reads data written by the *Elaborators*. When all buffers are not empty (all *Elaborators* completed operations on a frame), it merges data related to the same person, send them to the classifiers and awaits the answer. Eventually, it asks the *Accelerometer Control* to switch on the appropriate accelerometer;

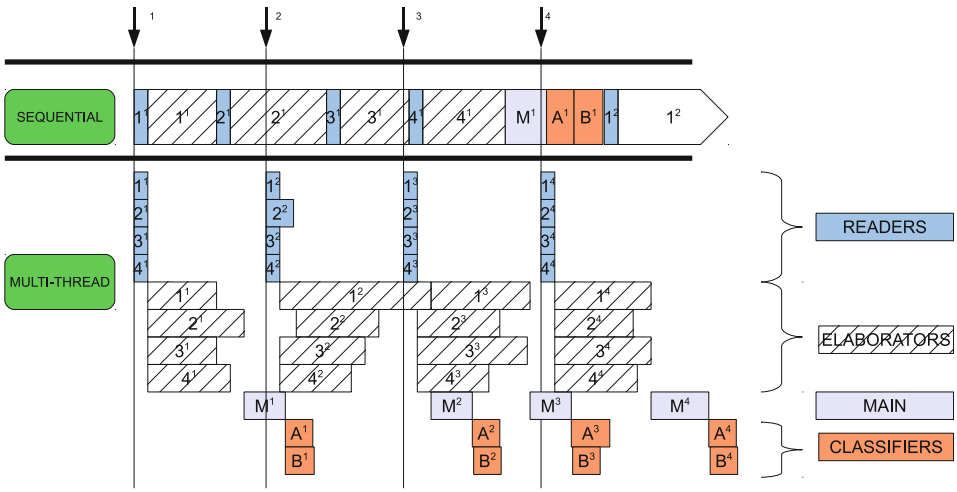

Fig. 11 Multi-thread operations. In this example, the four cameras (named 1, 2, 3, 4) are watching two people (named A, B). The arrows above correspond to the acquisition of a new frame by the cameras. The blocks on the right (indicated with *Readers*, *Elaborators*, *Main*, *Classifiers*) represent the running threads. The superscript refers to the number of the frame in which the action occurs

- each of several threads, *Video Classifiers*, handle a classifier for the optical flow data. They receive data from *Main* and send the results back to it;
- one thread, *Localization*, receives power levels from wireless receivers via RS232, estimates the positions of the accelerometers and compares them with the positions calculated by the camera-based system in order to match them;
- one thread, *Accelerometer Control*, handles the communication between the server and the body-mounted accelerometers: switching on, switching off, collecting data, classification and fall detection.

Comparing this architecture to a corresponding sequential one, the system improves its performances by almost three times, from an average running time of 0.153 s for processing 1 frame (6.53 fps) sequentially, to an average time of 0.056 s (theoretically 17.86 fps) when the bottleneck becomes the image acquisition frequency of the cameras (15 fps).

5.4 Experimental results

The experimental setup and dataset are the same used to evaluate the two single-sensor systems. The performances of the multi-sensor system can be analyzed according to different criteria:

- correct detection of dangerous situations and accelerometer activation;
- speed of activation of the accelerometers, in order for the system to be able to detect the falls timely;

Table 8 shows the statistics of accelerometer activation related with the different events.

In Table 9 we show the results of fall detection. Obviously, a fall can be detected only if the accelerometer sends all data related to such an event; therefore the fall must occur within two seconds before the activation of the accelerometer or shortly after. As can be seen, all falls were detected and correctly classified.

Table 8 Multi-sensor system

Event/switch on	No	Yes
Stand	92	8
Walk	63	37
Sit down	4	96
Stand up	4	96
Fall	0	100

Percentage of accelerometer activations by event

Table 9 Multi-sensor architecture

Present/detected	Yes	No
Yes	100	0
No	0	100

Correct fall detection percentage on test set

6 Conclusions

We presented a multi-sensor architecture for detecting and classifying human activities and a preliminary application aimed at detecting falls. The system is made up of two subsystems: one uses four cameras and the other one uses wearable accelerometers as sensors. The camera-based system reaches good results in event classification, but produces a higher number of false positives in detecting falls. Moreover, performances degrade when the subject cannot be observed by all cameras, as happens when the subject is out of the field of view of a camera. The accelerometer-based system obtains very good results in

detecting falls but the physical limits of the accelerometers (bandwidth and battery life) make this system unsuitable for monitoring several people for a long time. Furthermore, event classification performances are worse than those obtained by the vision-based system.

Therefore, the multi-sensor architecture based on the joint use of the two subsystems overcomes these disadvantages while being able to detect falls with very good accuracy and in a timely manner, thanks to its multi-threaded operations. Both the single-sensor and the multi-sensor systems classify different human actions exploiting a hybrid classifier that integrates Hierarchical Temporal Memories and SVMs.

The proposed hybrid classification system shows good results in classifying events that develop over time like human actions and demonstrates good generalization performances. Moreover, its modular architecture allows the system to scale with the complexity of the environment.

Future developments of this work will focus mostly on improving recognition accuracy and the resolution of data analysis, specifically of the accelerometer-based system. Moreover, we will test the results of the multi-sensor system when dealing with multiple people at one time. We aim at distinguishing not only between different kinds of events, but also between different instances of the same event, e.g., to evaluate the correctness of a subject's gait. As well, we will try to improve localization accuracy of the joint wireless- and vision-based system.

As a possible future application, we foresee our system as a tool used in a physical rehabilitation center to monitor patients and to assess their improvements over time.

Acknowledgments We would like to thank Gabriele Camellini for the great effort spent in the hardware setup of the environment and for the precious work on the wireless module-based localization. Moreover, we would like to thank Henesis s.r.l. for providing us with the wireless modules and the fellow VisLab laboratory for the cameras.

References

Burges CJ (1998) A tutorial on Support Vector Machines for pattern recognition. Data Min Knowl Discov 2:121–167

Csapo A, Baranyi P, Tikk D (2007) Object categorization using VFA-generated nodemaps and Hierarchical Temporal Memories. In: IEEE International Conference on Computational Cybernetics, ICCC 2007, pp 257–262

Cucchiara R, Grana C, Piccardi M, Prati A, Sirotti S (2001) Improving shadow suppression in moving object detection with HSV color information. In: Proceedings of IEEE Intelligent Transportation Systems, 2001, pp 334–339

Farahmand N, Dezfoulian M, GhiasiRad H, Mokhtari A, Nouri A (2009) Online temporal pattern learning. In: Proceedings of International Joint Conference on Neural Networks, IJCNN 2009, pp 797–802

George D, Hawkins J (2009) Towards a mathematical theory of cortical micro-circuits. PLoS Comput Biol 5(10). doi:10.1371/journal.pcbi.1000532

George D, Jaros B (2007) The HTM learning algorithms. Technical report, Numenta inc. http://www.numenta.com/htm-overview/education/Numenta_HTM_Learning_Algos.pdf. Accessed 16th Aug 2011

Grossmann R, Blumenthal J, Golatowski F, Timmermann D (2007) Localization in Zigbee-based sensor networks. In: Proceedings of 1st European ZigBee Developers Conference, EuZDC 2007

Hawkins J, Blakeslee S (2004) On intelligence. In: Times books, New York

Hsu CW, Chang CC, Lin CJ (2003) A practical guide to support vector classification. Technical report. Department of Computer Science, National Taiwan University, Taipei

Ji X, Liu H (2010) Advances in view-invariant human motion analysis: a review. IEEE Trans Syst Man Cybern Part C 40:13–24

Kellokumpu V, Pietikinen M, Heikkil J (2005) Human activity recognition using sequences of postures. In: Proceedings of IAPR Conference on Machine Vision Applications. MVA 2005, pp 570–573

Kern N, Schiele B, Schmidt A (2003) Multi-sensor activity context detection for wearable computing. In: Proceedings of EUSAI, pp 220–232

Khan A, Lee YK, Lee S, Kim TS (2010) A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. IEEE Trans Inf Technol Biomed 14(5):1166–1172

Kim Y, Ling H (2009) Human activity classification based on micro-Doppler signatures using a support vector machine. IEEE Trans Geosci Remote Sens 47(5):1328–1337

Leone A, Diraco G, Distante C, Siciliano P, Malfatti M, Gonzo L, Grassi M, Lombardi A, Rescio G, Malcovati P, Libal V, Huang J, Potamianos G (2008) A multi-sensor approach for people fall detection in home environment. In: Cavallaro A, Aghajan H (eds) Workshop on multi-camera and multi-modal sensor fusion algorithms and applications, M2SFA2 2008, Marseille France

Lester J, Choudhury T, Kern N, Borriello G, Hannaford B (2005) A hybrid discriminative/generative approach for modeling human activities. In: Proceedings of International Joint Conference on Artificial Intelligence, IJCAI, pp 766–772

Lihan M, Tsuchiya T, Koyanagi K (2008) Orientation-aware indoor localization path loss prediction model for wireless sensor networks. In: Network-based information systems, Springer, pp 169–178

Luo R, Chen O, Lin CW (2010) Indoor human monitoring system using wireless and pyroelectric sensory fusion system. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2010, pp 1507–1512

Monekosso D, Remagnino P, Kuno Y (eds) (2009) Intelligent environments—methods, algorithms and applications. Springer, Berlin

Mountcastle V (1978) An organizing principle for cerebral function: the unit model and the distributed system. In: Edelman G, Mountcastle V (eds) The mindful brain. MIT Press, MA

Niebles JC, Fei-Fei L (2007) A hierarchical model of shape and appearance for human action classification. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Los Alamitos, CA, pp 1–8

Rashidi P, Cook D, Holder L, Schmitter-Edgecombe M (2011) Discovering activities to recognize and track in a smart environment. IEEE Trans Knowl Data Eng 23(4):527–539

Rutishauser U, Joller J, Douglas R (2005) Control and learning of ambience by an intelligent building. IEEE Trans Syst Man Cybern Part A Syst Humans 35(1):121–132

Sassi F, Ascari L, Cagnoni S (2009) Classifying human body acceleration patterns using a hierarchical temporal memory. In: Proceedings of the XIth International Conference of the Italian

Association for Artificial Intelligence: emergent perspectives in artificial intelligence. Springer, AI*IA '09, pp 496–505

Schuldt C, Laptev I, Caputo B (2004) Recognizing human actions: a local SVM approach. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, vol 3, pp 32–36

Shi J, Tomasi C (1994) Good Features to Track. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR94), pp 593–600

Turaga P, Chellappa R, Subrahmanian VS, Udrea O (2008) Machine recognition of human activities: a survey. IEEE Trans Circuits Syst Video Technol 18(11):1473–1488

van Doremalen J, Lou B (2008) Spoken digit recognition using a hierarchical temporal memory. In: Proceedings of Interspeech 2008. Brisbane, Australia, pp 2566–2569

Ward JA, Lukowicz P, Troster G, Starner TE (2006) Activity recognition of assembly tasks using body-worn microphones and accelerometers. IEEE Trans Pattern Anal Mach Intell 28:1553–1567

Wu X, Ou Y, Qian H, Xu Y (2005) A detection system for human abnormal behavior. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2005, pp 1204–1208

Zhang S, Ang M, Xiao W, Tham C (2008) Detection of activities for daily life surveillance: eating and drinking. In: 10th International Conference on e-health Networking, Applications and Services, HealthCom 2008, pp 171–176

Zhu C, Sheng W (2009) Multi-sensor fusion for human daily activity recognition in robot-assisted living. In: Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction, ACM, New York, NY, USA, HRI '09, pp 303–304