Unsupervised Learning in Body-Area Networks

Nicola Bicocchi[1], Matteo Lasagni[1], Marco Mamei[1]
Andrea Prati[1], Rita Cucchiara[2], Franco Zambonelli[1]
[1]Dipartimento di Scienze e Metodi dell'Ingegneria
[2]Dipartimento di Ingegneria dell'Informatione
University of Modena and Reggio Emilia
ITALY
{name.surname}@unimore.it

ABSTRACT

Pattern recognition is becoming a key application in body-area networks. This paper presents a framework promoting unsupervised training for multi-modal, multi-sensor classification systems. Specifically, it enables sensors provided with patter-recognition capabilities to autonomously supervise the learning process of other sensors. The approach is discussed using a case study combining a smart camera and a body-worn accelerometer. The body-worn accelerometer sensor is trained to recognize four user activities pairing accelerometer data with labels coming from the camera. Experimental results illustrate the applicability of the approach in different conditions.

Categories and Subject Descriptors

I.5 [**Computing Methodologies**]: Pattern Recognition; I.2.3 [**Artificial Intelligence**]: Probabilistic Reasoning

General Terms

Algorithms, Design, Experimentation, Human Factors, Measurement, Performance.

Keywords

Body-Area Networks, Body-Worn Accelerometers, Smart Cameras, Activity Recognition.

1. INTRODUCTION

The automatic and unobtrusive identification of user's activities from sensor data is one of the key and most challenging goals of context-aware and wearable computing [10]. Several opportunities can arise from the availability of such an information. For example, a service would be able to flexibly adapt its behavior to different circumstances (e.g., a smart phone application could turn silent, once recognizing that the user is in a theater watching a movie). As another example, the simple description of user's activities could automatically produce entries in blogs, diaries and social network sites [2].

While advances in hardware technologies (e.g., smart phones and body-worn sensors [2, 10]) are making feasible to collect a vast amount of information about the user in an unobtrusive way, it is still difficult to organize and aggregate all the collected information in a coherent, expressive and semantically-rich representation. In other words there is a gap between low-level sensor readings and their high-level context description [15].

A possible solution to overcome these challenges is to develop multi-modal classification systems combining body-worn and environmental sensors. This kind of systems would present two main advantages. On the one hand it has been proven that multi-modal sensor fusion can greatly enhance the performance of the system by looking at the problem from different and complementary perspectives [20, 12]. Accordingly, multi-modal, multi-sensor classifiers could be able to discriminate situations otherwise difficult to be identified. On the other hand, such flexible classifiers could work with dynamically varying kind of sensors, adapting to different circumstances.

There are two mainstream approaches to build such kind of multi-modal distributed classification systems:

- *Classification fusion.* This approach to multi-modal, multi-sensor classification consists of combining the classification outputs of different sensors. In more detail, given that each sensor provides ordered or weighted class labels, the final outcome can be obtained as a combination of the class labels according to some metrics [20].

- *Distributed classification.* This approach consists of organizing sensors in a network of inference. The output of some sensors is used as input to other sensors to conduct higher-level classification tasks [14, 17].

Despite the advantages of these two approaches, in this paper we focus on a third alternative. We propose a general approach to create a self-training sensors' ecosystem. This approach (that can be used in combination with the former two) lets sensors cooperate to create a model to classify the signal. For example, a "trained sensor" (one having already a classification model) can provide ground truth information to an "untrained sensor" (one able to produce raw data only) to enable it running a learning algorithm on its

BodyNet 2010, Corfu Island, Greece

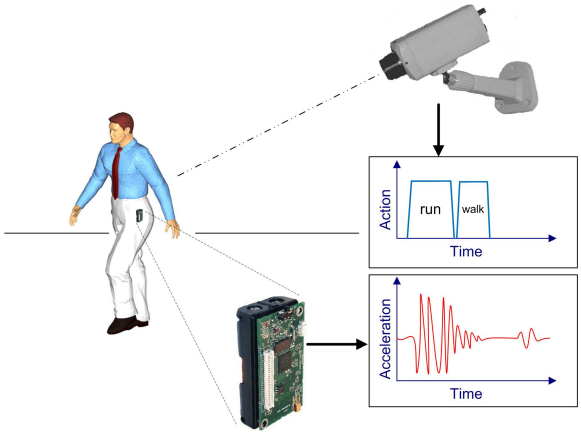

Figure 1: A multi-sensor installation. A camera sensors automatically train an acceleration sensor to detect user activities.

data and become a "trained sensor" itself. The core idea is to use labels coming from on sensor type to train another sensor type. In particular, we focus on a scenario in which a body-worn accelerometer sensor learns to recognize user activities on the basis of the labels provided by a camera sensor (see Figure 1). This combination of body-worn sensors and cameras is novel and can facilitate the deployment of large pervasive environments comprising both body-area sensor networks and sensors deployed in the environment.

The rest of this paper will be structured as follows: Section 2 will present related works in the area. Section 3 presents the system architecture. Section 4 presents an implemented use case illustrating our approach and highlights the statistical approach used to run learning and classification algorithms. Section 5 shows experiments and results conducted on our test-bed. Finally Section 6 concludes and outlines future work in the area.

2. RELATED WORKS

The work described in [20] presents a multi-modal, multi-sensor classification system using body-worn microphones and accelerometers. Sensors' outputs are independently classified and their values are merged together using different kinds of fusion techniques. Besides different sensors have been used, our work is different in that we combine sensors also in the learning phase. Furthermore, our system also allows individual sensors to work in isolation.

Another interesting work in the same area is presented in [12]. This work combines accelerometer-based classification with data coming from a RFID-glove reporting information about the RFID-enabled objects touched by the user.

A system to perform multi-sensor motion recognition is described in [13]. In this work, data collected by 5 body-worn accelerometers are used to classify user activities. Feature extraction and classification techniques are similar to our approach. However, it is based on supervised learning while our approach is unsupervised.

The analysis of human movements has been a very active research area in the computer vision community. Gavrila in [8] surveyed the existing approaches to whole-body or hand motion analysis. Most of the reported approaches focus on specific human motions and are heavily based on a model used for the system training. For instance, Yam *et al.* [21] proposed a system for people identification based on the analysis of their gait. The system extracts leg motion by temporal template matching in which the periodic motion of leg is used as a model. Fourier analysis is employed to analyze the periodicity of the leg motion. However, this approach can only work on pure lateral views and is basically capable to distinguish only between walking and running. Cutler and Davis [5] made a similar assumption and aimed at detecting periodic motion by using Short Term Fourier Transform (STFT) for time-frequency analysis. Their system is able to work also with moving cameras, but is limited to periodic human motion patterns.

Urtasun and Fua [19] exploited a more general model and used temporal motion models based on PCA to formulate the human body tracking problem as one of minimizing differentiable objective functions. Moreover, a multi-activity database is accessed to compare extracted features with a theoretically infinite set of human motion models.

Finally, recent advances in statistical pattern recognition and computer vision techniques contribute to a new era of research on human motion understanding, as demonstrated by the quite recent special issue on Computer Vision and Image Understanding journal [11]. As an example among the many existing proposals, Robertson and Reid [18] model human behaviors as a stochastic sequence of actions and evaluate the likelihood by using HMM. The observations come from position, velocity and motion descriptors, and matching is performed against a labeled database of actions.

3. DESCRIPTION OF THE SYSTEM

The proposed framework comprises three classes of components: trained sensors, untrained sensors and tuple spaces. They are illustrated in Figure 2 and described below.

Trained sensors are able to sample environmental data and produce labels based on the classification of their inputs. Labels are pushed towards a shared tuple space (described below). Trained sensors can be divided in two categories: *(i)* sensors that successfully completed the training stage of their classification algorithm or *(ii)* sensors with hard-coded classification algorithms that do not require training.

Untrained sensors did not successfully completed the training stage of their classification algorith. They sample environmental data and store them in a circular buffer. Each entry of this buffer is composed by two elements. The first one is an environmental sample (i.e., a feature vector), while the second one is a classification label. The training process ends when a sufficient number of entries are properly filled with a feature vector and the corresponding label. At this stage, the sensor can start classifying and publishing labels autonomously (i.e., becoming a *trained sensor*).

Trained and untrained sensor share the same internal architecture and are composed by three functional modules: *(i)* a data acquisition module dedicated to environmental data sampling, *(ii)* a learning module that pairs collected data with external labels to build a classification model, and *(iii)* a classification module that uses the classification model to classify new data (see Figure 2a).

Tuple spaces [9] receive labels from *trained sensors* and

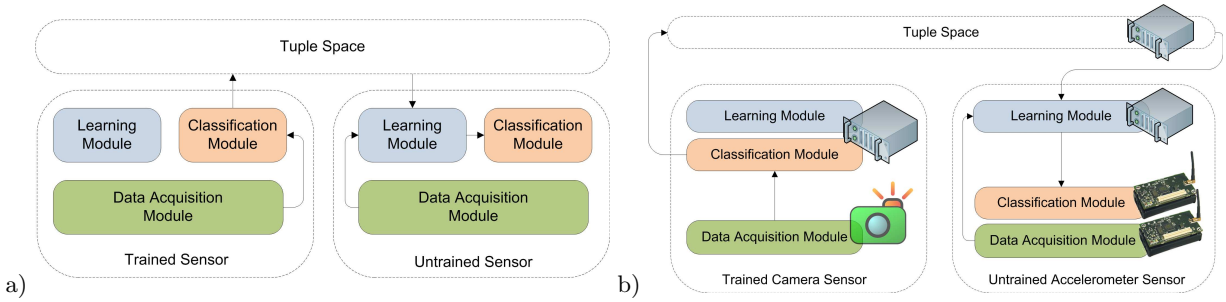

Figure 2: Representation of the functional components of the system (a), and their distribution on different computational devices (b).

forward them to *untrained sensor*. To avoid broadcast communication while keeping the system simple, tuple space access is arbitrated by a publish-subscribe mechanism [7]. Trained sensors publish their labels. Untrained sensors subscribe a template to specify which type of labels they need to receive. This allows engineers to realize complex pervasive environments populated by a large number of sensors monitoring different aspects. For example, this approach allows multiple *trained sensors* to jointly train multiple *untrained sensors*.

4. IMPLEMENTED CASE STUDY

To assess the feasibility of the approach, we implemented a case study. We trained a body-worn accelerometer sensor to recognize user activities using labels provided by a camera sensor. Specifically, we have used an analog camera, a couple of Crossbow MicaZ mote equipped with MTS310 sensor boards (*http://www.xbow.com*), and a dedicated server. Considering the resource constraints of the body-worn sensors we mapped the functional blocks described in Figure 2a to several devices according to Figure 2b. In particular: the camera is considered as a simple data acquisition module, while the two coupled motes run both the data acquisition and classification modules. All the remaining components run on the dedicated server.

We started by considering the following four activities: "walking", "running", "standing still" and "falling down". These activities can be recognized either by a camera sensor running a video analysis algorithm [3] or by an accelerometer sensor analyzing the acceleration patterns [6]. We installed the camera sensor in our department and configured it to upload arising labels to the tuple space. Then, we configured the untrained accelerometer sensor to receive from the tuble space labels produced by the camera.

The accelerometer sensor started receiving the labels from the camera and pairing them with acceleration samples. Once a sufficient number of acceleration samples and labels was collected, the accelerometer sensor started to classify the four user activities without any external support.

4.1 Accelerometer Sensor

In our implementation the accelerometer sensor is built by two MicaZ motes with MTS310 sensor board each sampling data at 100Hz. The MTS310 board is provided with a two-axis accelerometer with a sensing range of +/- 2g and a sensibility of +/- 2mg. We used two motes to acquire three-axis acceleration data (see Figure 3).

Learning Module

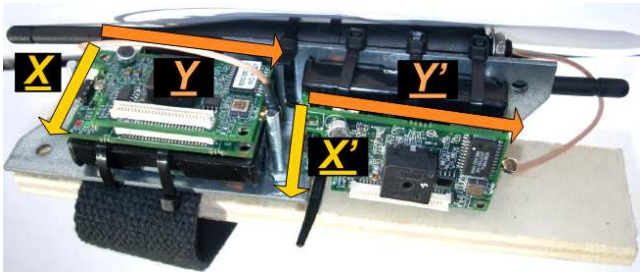

Figure 3: Picture of two MicaZ motes (used in our experiments) with their coordinate reference system.

Due to the fact that accelerations were recorder by two distinct motes, they were not perfectly temporally aligned. Thus we aligned the four time series (X, Y), (X', Y') (see Figure 3) in a way to that two axes (Y, Y') overlap. To achieve this type of synchronization on events recorded by different sensors simultaneously we used a particular event, a jump, brightly visible in both the time series, similarly to what has been proposed in [1]. Accelerometer signal is also aligned with the camera one. Once an event is detected by the camera, it is notified to all the subscribing sensors with neglibile communication delays.

After the alignment, the raw input signal $\mathbf{T} = \{\overrightarrow{a_x}, \overrightarrow{a_y}, \overrightarrow{a_z}\}$ provided by the accelerometer consists of the separate accelerations along the three axes x, y and z. To simplify the representation and reduce the dimensionality, the magnitude of the sum vector is obtained:

$$A = \|\overrightarrow{a_x} + \overrightarrow{a_y} + \overrightarrow{a_z}\| \quad (1)$$

In addition, processing the non-oriented scalar magnitude would produce results independent on the actual way the accelerometer is worn and, thus, its orientation.

As stated in [6], the power spectrum of the magnitude A can be a good feature to use. First, the time series $\mathbf{AS} = \{A\}$ provided by the accelerometer is analyzed in the frequency domain through the FFT (Fast Fourier Transform) transform. To reduce the computational load, a small 64 element window with 32 samples of overlap is used. From these 64 elements, one 32 element power spectrum (every 0.32 seconds) is produced. Ultimately, the DC component is canceled since it affects numerical stability (due to its relatively large value). Summarizing, this procedure converts a time series **AS** defined in the spatial domain on $\mathbb{R}$ to a time series $\mathbf{X}$ defined in the frequency domain on $\mathbb{R}^{31}$.

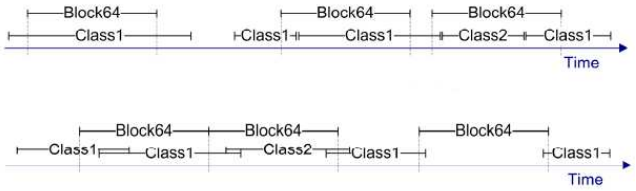

Figure 4: The labelling process of an untrained sensor. During the time slice covered by a feature vector, multiple and conflicting labels might be received. Majority voting is applied to solve the conflicts.

Our cooperative learning approach assigns automatically a label/class to untrained sensor data using the posterior-based classification provided by the trained sensors. Thus, data coming from both types of sensors are paired to build the training set $\mathbf{TS} = \{S_1, S_2, \cdots, S_{N_{TR}}\}$, where N_{TR} is the total number of samples in the training set and $S_i = \langle X_i, C_i \rangle$ represents the i-th sample and is composed by the 31-dimensional accelerometer observation X and the corresponding label C provided by the camera with a majority-voting process described in Figure 4.

Following a generative model, we first solve the inference problem of determining the class-conditional densities $p(\mathbf{X}|C_i)$ for each class C_i individually. Thus, let $\mathbf{X}^{C_i}$ be the set of accelerometer observations in the training set associated in the labelled data to the class C_i. This likelihood can be modelled with a 31-variate mixture of Gaussians (MoG):

$$p\left(\mathbf{X}^{C_i}|C_i\right) = p\left(\mathbf{X}^{C_i}|\mathbf{A}^{C_i}\right) = \sum_{k=1}^{K} \pi_k^{C_i} \mathcal{N}\left(\mathbf{X}^{C_i}|\boldsymbol{\mu}_k^{C_i}, \boldsymbol{\Sigma}_k^{C_i}\right) \tag{2}$$

where $\mathbf{A}^{C_i} = \left\{\boldsymbol{\mu}^{C_i}, \boldsymbol{\Sigma}^{C_i}, \boldsymbol{\pi}^{C_i}\right\}$ is the set of parameters for for the class C_i, including the mean vector $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K\}$ for each of the K components, the covariance matrix $\boldsymbol{\Sigma}$ and the weight vector $\boldsymbol{\pi}$. The single 31-variate Gaussian can be written as:

$$\mathcal{N}(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{31/2}} \frac{1}{(\det(\Sigma))^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{X}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{X}-\boldsymbol{\mu})\right\} \tag{3}$$

In order to estimate the MoG's parameters, we employ the well-known EM algorithm and chose to keep the number of MoG components K fixed to two. This choice has been validated by a thorough experimentation which demonstrates that increasing K brings no benefit to the performance.

One important optimization we performed is to force the MoG 31×31 covariance matrices $\boldsymbol{\Sigma}_k^{C_i}$ to be diagonal, thus assuming that the 31 Fourier components are statistically independent. This notably reduces the size of the parameter set $\mathbf{A}$ used to represent the model, making $\mathbf{A}$ storage on the tiny memory of the body-worn sensor possible.

Classification Module

The set $\mathbf{A}$ of estimated parameters is stored in the mote to be used for motion pattern classification during the on-line testing phase. Since the accelerometer is now trained the on-line classification of a new sample X_{new} is simply performed by applying Bayes rule and MAP (Maximum A Posteriori) framework:

$$C_{new} = C_s \Leftrightarrow s = \arg\max_{\forall r} p(C_r|X_{new}, \mathbf{A}) \tag{4}$$

where, assuming uniform sample's priori, the posterior class probability can be written as:

$$p(C_r|X_{new}, \mathbf{A}) \propto p(X_{new}|\mathbf{A}, C_r)\, p(C_r|\mathbf{A}) \tag{5}$$

The first term in the right-side of equation 5 corresponds to the MoG for a given class C_r with parameters $\mathbf{A}^{C_r}$, while the second term can be simplified as $p(C_r)$ since there is no dependency of the class on the MoG's parameters. The term $p(C_r)$ represents the prior of class C_r and can be computed as the normalized occurrence of that class in the training set.

4.2 Camera Sensor

The other modality of our system is provided by a standard fixed color camera and based on the video data, such as those shown in Figure 5 which includes both lateral and longitudinal views. The classical computer vision flow considers first to segment moving objects (*segmentation*), then to track them on the field of view of the camera (*tracking*), and ultimately to analyze the objects to classify them and to infer their behavior (*object analysis*).

Background suppression-based algorithms are the most common techniques adopted in video surveillance for detecting moving objects. The background model used should be adaptive with both high responsiveness, avoiding the detection of transient spurious objects, such as cast shadows, static non-relevant objects or noise, and high selectivity in order to not include in the background the objects of interest. A robust and efficient approach is to detect visual objects by means of background suppression with pixel-wise temporal median statistics, that is, a statistical approach to decide whether the current pixel value is representative of the background information is employed. In this way, a background model for non object pixels is constructed from the set:

$$S = \left\{I^t, I^{t-\Delta t}, \cdots, T^{t-(n-1)\Delta t}\right\} \cup \left\{\underbrace{B^t, \cdots, B^t}_{w_b \text{times}}\right\} \tag{6}$$

The set contains n frames sub-sampled from the original sequence of frames I taking one frame every Δt, and an adaptive factor that is obtained by including the previous background model B^t w_b times. The new value for the background model is computed by taking the median of the set S, for each pixel p:

$$B_{stat}^{t+\Delta t}(p) = \arg\min_{x \in S(p)} \sum_{y \in S(p)} \max_{c=R,G,B} (|x_c - y_c|) \tag{7}$$

The distance between two pixels is computed as the maximum absolute distance between the three channels R, G, B, which has experimentally proved to be more sensitive to small differences and quite consistent in its detection ability. Although the median function has been experimentally tested as a good compromise between efficiency and reactivity to luminance and background changes, the background model should not include the interesting moving objects if

Figure 5: An example of appearance-based tracking showing: the input image (a), the appearance mask model AMM (b), and the probability mask PM (c).

their motion is low or zero for a short period. Therefore, precise segmentation should exploit the knowledge of the type of the object associated with each pixel to selectively update the reference background model. In order to define a general-purpose approach, we have defined a framework where visual objects are classified into three classes: actual visual objects, shadows, or "ghosts", i.e. apparent moving objects typically associated with the "aura" that an object that begins moving leaves behind itself. Further details can be found in [3].

Once that moving objects/people have been segmented, they need to be tracked along time in order to analyze the scene (counting the number of people present in there) and analyze their behavior (analyzing the paths or the human motion patterns). In people tracking, in order to cope with non-rigid body motion, frequent shape changes and self-occlusions, probabilistic and appearance-based tracking techniques are commonly proposed [4].

In this paper, we describe an approach of appearance-based probabilistic tracking, specifically conceived for handling all occlusions. The algorithm uses a classical predict-update approach. It takes into account not only the status vector containing position and speed, but also the *Appearance Memory Model AMM* and the *Probabilistic Mask PM* of the shape. AMM represents the estimated aspect (in RGB space) of the object's points: each value $AMM(p)$ represents the "memory" of the point's appearance, as it has been seen up to now. In the probability mask $PM(p)$ each value defines the probability that the point p belongs to the object. An example for "walking", is reported in Figure 5.

These two features are used both to perform the matching between the existing tracks and the objects detected in the current frames, and to detect the presence of occlusions. Occlusions are detected by analyzing a measure of confidence and likelihood computed on the probability mask PM. Further details can be found in [4].

Classification Module

Given the observation I^t, the algorithm bases its classification on the analysis of the tracked person. It is worth noting that the computer vision algorithm is capable to track multiple targets but needs to associate that (or those) target(s) wearing the accelerometers with the corresponding target ID(s). This association can be obtained with the same synchronization procedure described in Section 4.1, based on a specific event such as a jump. Thanks to the sophisticated tracking algorithm above mentioned, the distinction between "standing still" (class S), "walking" (class W) and "running" (class R) is almost straightforward. Given that the frame rate is constant, the movement speed can be easily estimated, even though perspective distortion can strongly affect the estimation.

Let H^t and V^t be the height of the bounding box and the velocity of the tracked person in I^t, respectively. The velocity is estimated through the distance (in pixels) covered by the person's centroid between I^{t-1} and I^t. By using a discriminative approach we can infer the posterior class C_i probabilities as follows:

$$p\left(C_i = S|I^t\right) = \frac{1}{\sqrt{2\pi}\sigma_1} exp\left\{-\frac{\left(V^t\right)^2}{2\sigma_1^2}\right\} \quad (8)$$

$$p\left(C_i = W|I^t\right) = \frac{1}{\sqrt{2\pi}\sigma_2} exp\left\{-\frac{\left(V^t - Th_1\right)^2}{2\sigma_2^2}\right\} \quad (9)$$

$$p\left(C_i = R|I^t\right) = \frac{1}{1 + exp\left\{-\left(V^t - Th_2\right)\right\}} \quad (10)$$

$$p\left(C_i = F|I^t\right) = \frac{1}{1 + exp\left\{-\left(\Delta H - Th_3\right)\right\}} \quad (11)$$

In other words, a person is classified as standing still if his/her velocity is almost zero, modeled with a zero-mean Gaussian (see equation 8) with a small standard deviation σ_1 (set to 1 in our tests). The same applies for classifying walking people (equation 9), but with a Gaussian centred on a positive value Th_1 (set to 5) and with a large standard deviation σ_2 (set to 3). Moreover, the person is classified as running if the velocity is greater than a threshold Th_2 (set to 10 pixels). Thresholding is performed with a logistic sigmoid function centred on the value Th_2 (see equation 10). Logistic sigmoid function provides a smoother transition than the classical Heaviside function used when thresholding. Finally, the classification of falling people is performed by looking at the variation $\Delta H = H^{t-1} - H^t$ of the height of the bounding box: if this variation is greater than a threshold Th_3 (set to 5) this means that the height is changing fast and this is a cue for a fall.

Given the equations 8-11, a MAP approach provides the class maximizing the posterior probabilities:

$$C_j \quad | \quad j = \arg\max_{\forall k} \; p\left(C_k|I^t\right) \quad (12)$$

5. EXPERIMENTAL RESULTS

To evaluate the performances of our system we did the following experiment. We recorded with the camera a person moving inside a corridor in our department, acting in several motion states. At the same time, the same person wore the accelerometer sensors on his belt. In this way we collected data representing two observations of the same phenomenon. While the scene was recorded, we manually collected with a digital chronometer a list of labels to be used as ground truth. The camera sampled at 30Hz, while the accelerometers at 100Hz. We upsampled the camera's

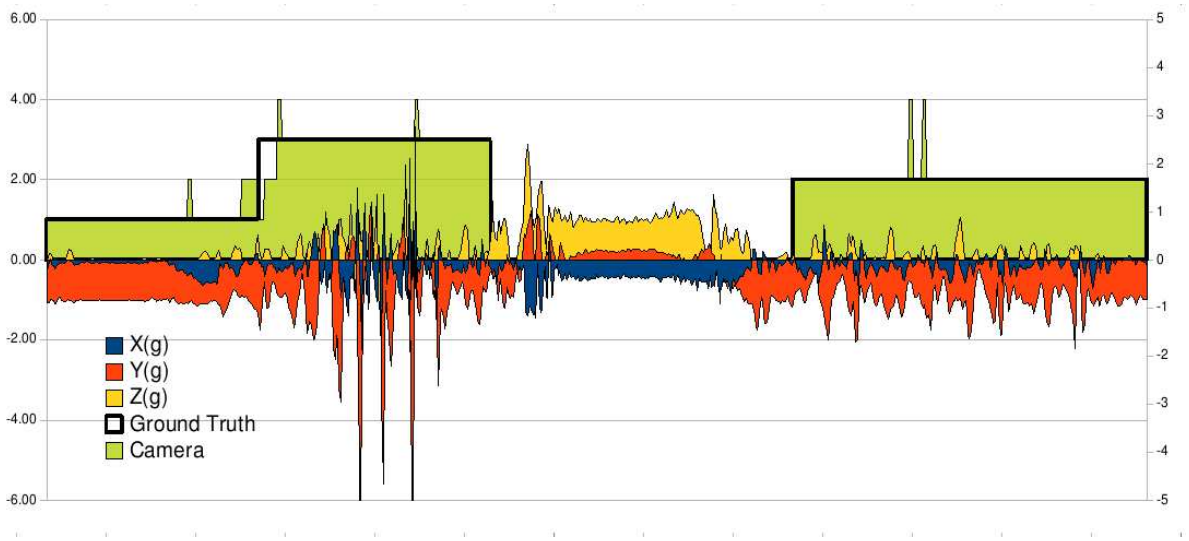

Figure 6: A snapshot of the multi-sensor training set we used. Camera's classification labels and ground truth are superimposed over acceleration data.

	Fall	Stand	Walk	Run
Fall	**84.27%**	8.70%	4.76%	2.28%
Stand	0.00%	**86.43%**	13.00%	0.57%
Walk	2.71%	0.48%	**94.81%**	2.00%
Run	0.43%	9.12%	1.72%	**88.73%**

(a)

	Fall	Stand	Walk	Run
Fall	**15.00%**	21.25%	63.75%	0.00%
Stand	0.00%	**77.63%**	22.37%	0.00%
Walk	0.00%	1.96%	**97.20%**	0.84%
Run	0.79%	0.79%	10.24%	**88.19%**

(b)

Table 1: Confusion matrix of camera sensor classification (a), and accelerometer sensor classification (b).

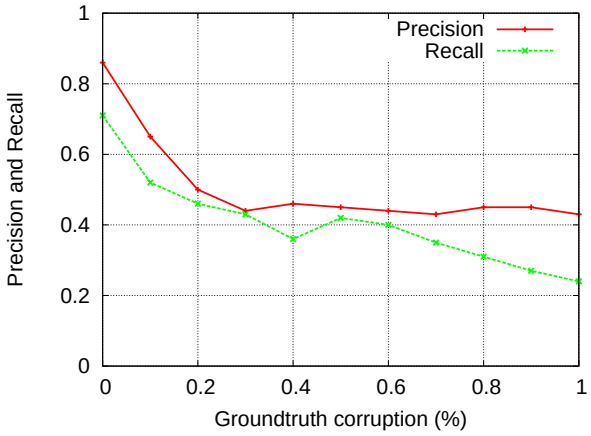

Figure 7: Precision and recall of the accelerometer sensor classification varying the quality of ground truth labels provided by the camera.

data to 100Hz and aligned the time series with the ground truth. To ground the discussion we reported in Figure 6 a snapshot of the data we collected in this experiment. This plot shows the three accelerations (on X, Y and Z direction) with superimposed both the ground-truthed and the camera-based classifications, in order to show the correspondence of accelerations' changes with mition variations.

We used these data in two ways. First, we compared the labels produced by the camera with the ones manually collected to measure the computer vision algorithm performances. Table 1 presents the confusion matrix of this classification. Looking at the reported confusion matrix of the camera sensor classification, it is possible to see that the vision algorithm works pretty well, having a precision of 88.59% and a recall of 88.56%.

As second experiment, we measured the classification performance of the accelerometer, once it has been trained with the labels coming from the camera. We compared the labels produced by the accelerometer with the ground truth to obtain the confusion matrix reported in Table 1. Overall the classification has a precision of 85.25% and a recall of 69.50%. The poor performance on detecting falls (that contributes also to decrease the overall recall of the system) is due to the fact that the falling action is composed of a transient phase in which the accelerometer values can vary significantly. A more correct modeling of this action would be to take the sequence of observations into account, for instance with a HMM.

Although accelerometer classification performance is quite good, it is possible to see a reduction of performance with respect to the vision classification. This is due to two simple facts: first and most importantly, a belt-worn accelerometer produces a signal that is less descriptive than a full-fledge video sequence. Accordingly, the features extracted from the acceleration signal are less capable of discriminating among high-level behaviors (e.g., falling and walking) than the video sensor. Second, the accelerometer is trained with labels coming from the camera sensor and thus it is affected also by the errors coming from the camera.

To better understand the contribution of this latter source

of errors, we conducted another experiment aimed at evaluating the robustness of the approach with respect to the camera classification errors.

We run a number of learning phases of the accelerometer data, passing to the accelerometer learning module increasingly corrupted class labels: we corrupted the ground truth information with errors sampled accordingly to the camera sensor confusion matrix reported in Table 1. Then, we test the accelerometer classification performance. The graph in Figure 7 shows on the x-axis the percentage of errors introduced, varying from 0 to 100 percent. Precision and recall are reported on the y-axis.

To summarize the discussion about the performances of the system it is possible to see that: *(i)* the algorithms proposed to track moving persons and classify their motion state are suitable for this application scenario; *(ii)* the algorithm we choose to classify acceleration data is working properly; *(iii)* our idea of a self-training sensor ecosystem is feasible and could help to reduce resources needed to deploy near-coming pervasive infrastructures.

6. CONCLUSION AND FUTURE WORK

In this paper we presented a framework promoting unsupervised training for multi-sensor classification systems deployed in body-area networks. We also grounded the idea using a case study in which a body-worn accelerometer sensor is automatically trained to recognize four human activities by a camera sensor. The presented approach might facilitate the deployment of large distributed sensor system in that no human intervention is required.

We also have identified three interesting direction to improve this work. First, we will include additional sensors into the system (e.g., accelerometers in different parts of the body, acoustic sensors and, body-worn cameras). Second, we will introduce more accurate classification models to detect a wider spectrum of human activities. Finally, we will combine sensor data with labels coming form the Internet to further reduce the need of human interventions [16].

7. REFERENCES

[1] D. Bannach, P. Lukowicz, and O. Amft. Rapid prototyping of activity recognition applications. *IEEE Pervasive Computing*, 7(2):22 – 31, 2008.

[2] A. Campbell, S. Eisenman, N. Lane, E. Miluzzo, R. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G. Ahn. The rise of people-centric sensing. *IEEE Internet Computing*, 12(4):12–21, 2008.

[3] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani. Probabilistic posture classification for human behaviour analysis. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 35(1):42–54, Jan. 2005.

[4] R. Cucchiara, C. Grana, G. Tardini, and R. Vezzani. Probabilistic people tracking for occlusion handling. In *Proceedings of IAPR International Conference on Pattern Recognition (ICPR 2004)*, volume 1, pages 132–135, Aug. 2004.

[5] R. Cutler and L. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):781–796, 2000.

[6] R. W. DeVaul and S. Pentland. The mithril real-time context engine and activity classification. Technical Report, MIT Media Lab, 2003.

[7] P. Eugster, P. Felber, R. Guerraoui, and A. Kermarrec. The many faces of publish/subscribe. *ACM Computing Surveys*, 35(2):114 – 131, 2003.

[8] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, Jan 1999.

[9] D. Gelernter and N. Carriero. Coordination languages and their significance. *Communication of the ACM*, 35(2):96 – 107, 1992.

[10] M. Hanson, H. Powell, A. Barth, K. Ringgenberg, B. Calhoun, J. Aylor, and J. Lach. Body area sensor networks: Challenges and opportunities. *IEEE Computer*, 42(1):58–65, 2009.

[11] A. Hilton, P. Fua, and R. Ronfard. Foreword: Special issue on modeling people: Vision-based understanding of a personŠs shape, appearance, movement, and behaviour. *Computer Vision and Image Understanding*, 104(2-3):87–89, 2006.

[12] I. Kim, S. Im, E. Hong, S. Ahn, and H. Kim. Adl classification using triaxial accelerometers and rfid. In *International Conference on Ubiquitous Computing Convergence Technology*. Beijing, China, 2007.

[13] S. I. L. Bao. Activity recongnition form user-annotated accleration data. In *International Conference on Pervasive Computing*. Linz/Vienna, Austria, 2004.

[14] N. Oliver and E. Horvitz. A comparison of hmms and dynamic bayesian networks for recognizing office activities. In *International Conference on User Modeling*. Edinburgh, UK, 2005.

[15] D. Patterson, L. Liao, D. Fox, and H. Kautz. Inferring high-level behavior from low-level sensors. In *International Conference on Ubiquitous Computing*. ACM Press, Seattle (WA), USA, 2003.

[16] M. Philipose, K. Fishkin, M. Perkowitz, D. Patterson, D. Fox, H. Kautz, and D. Hahnel. Inferring activities from interactions with objects. *IEEE Pervasive Computing*, 3(4):50 – 57, 2004.

[17] J. Predd, S. Kulkarni, and H. Poor. Distributed learning in wireless sensor networks. *IEEE Signal Processing Magazine*, 23(4):56 – 69, July 2006.

[18] N. Robertson and I. Reid. A general method for human activity recognition in video. *Computer Vision and Image Understanding*, 104(2-3):232–248, 2006.

[19] R. Urtasun and P. Fua. 3d human body tracking using deterministic temporal motion models. In *ECCV (3)*, pages 92–106, 2004.

[20] J. Ward, P. Lukowicz, G. Troster, and T. Starner. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1553 – 1567, 2006.

[21] C.-Y. Yam, M. Nixon, and J. Carter. On the relationship of human walking and running: Automatic person identification by gait. In *ICPR (1)*, pages 287–290, 2002.