

# The adARC pattern analysis architecture for adaptive human activity recognition systems

**Journal Article****Author(s):**

Roggen, Daniel; Förster, Kilian; Calatroni, Alberto; Tröster, Gerhard

**Publication date:**

2013-04

**Permanent link:**

<https://doi.org/10.3929/ethz-b-000043029>

**Rights / license:**

[Creative Commons Attribution-NonCommercial 2.0 Generic](#)

**Originally published in:**

Journal of Ambient Intelligence and Humanized Computing 4(2), <https://doi.org/10.1007/s12652-011-0064-0>

# The adARC pattern analysis architecture for adaptive human activity recognition systems

Daniel Roggen · Kilian Förster · Alberto Calatroni · Gerhard Tröster

Received: 6 April 2011 / Accepted: 15 June 2011 / Published online: 18 August 2011  
© The Author(s) 2011. This article is published with open access at Springerlink.com

**Abstract** Most approaches to recognize human activities rely on pattern recognition techniques that are trained once at design time, and then remain unchanged during usage. This reflects the assumption that the mapping between sensor signal patterns and activity classes is known at design-time. This cannot be guaranteed in mobile and pervasive computing, where unpredictable changes can often occur in open-ended environments. Run-time adaptation can address these issues. We introduce and formalize a data processing architecture extending current approaches that allows for a wide range of realizations of adaptive activity recognition systems. The *adaptive* activity recognition chain (*adARC*) includes self-monitoring, adaptation strategies and external feedback as components of the now closed-loop recognition system. We show an *adARC* capable of unsupervised self-adaptation to run-time changing class distributions. It improves activity recognition accuracy when sensors suffer from on-body displacement. We show an *adARC* capable of adaptation to changing sensor setups. It allows for scalability by enabling a recognition systems to autonomously exploit newly introduced sensors. We discuss other adaptive recognition systems within the *adARC* architecture. The results outline that this architecture frames a useful solution space for the real-world deployment of adaptive activity recognition systems. It allows to present and compare recognition systems in a coherent and modular manner. We discuss the challenges and new research directions resulting from this new perspective on adaptive activity recognition.

---

D. Roggen (✉) · K. Förster · A. Calatroni · G. Tröster  
Wearable Computing Laboratory, ETH Zurich, Zurich,  
Switzerland  
e-mail: droggen@gmail.com

## 1 Introduction

Recognizing human activities and gestures<sup>1</sup> (Davies et al. 2008) is important in pervasive computing (Weiser 2002), wearable computing (Mann 1998) and in human computer interaction (HCI) (Myers et al. 1996). It enables systems capable of pro-actively supporting users with just-in-time assistance, systems responding to natural interactions, or systems mining daily life patterns.

On-body sensing is emphasized in wearable computing, mobile computing and HCI as it allows to devise smart assistants or smart interfaces that do not require ambient infrastructure, and thus that work anywhere. A wide range of sensing modalities are now available, supported by technological advances that enable the large scale deployment of highly miniaturized, unobtrusive and interconnected (wireless) sensor systems (Benini et al. 2006) in our living environments, in devices we carry with us, and even in our outfits. We focus here on activity recognition from on body sensors with sporadic use of simple ambient sensors (e.g. presence, movement, contact switches).<sup>2</sup>

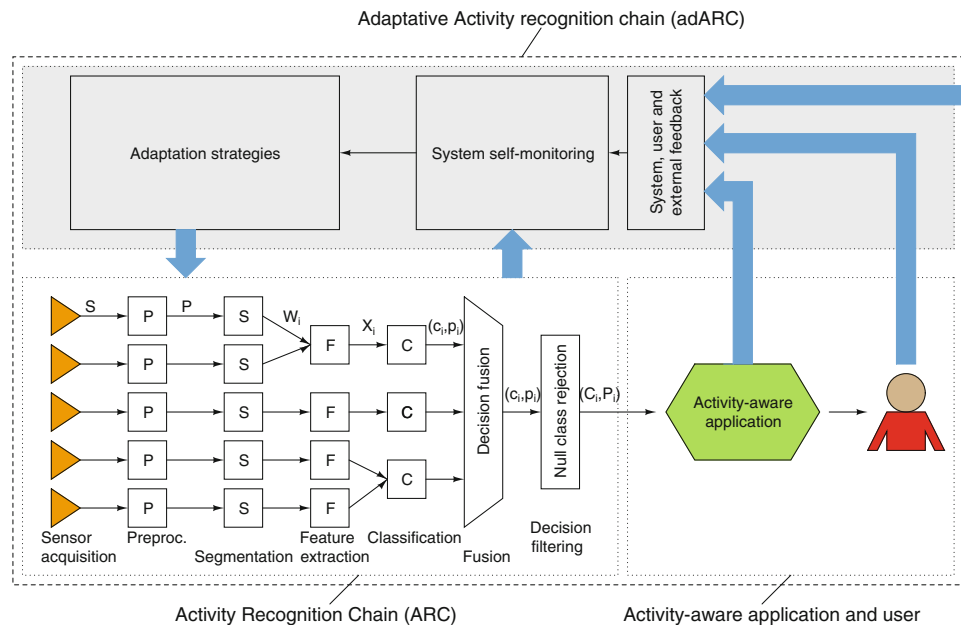
### 1.1 Problem statement

A relatively standard set of processing stages has emerged as the dominant approach for activity recognition (Bao and Intille 2004; Ward et al. 2006). We refer to this as the

---

<sup>1</sup> Sometimes also called behavior recognition, machine recognition of human behavior or activities.

<sup>2</sup> Vision-based approaches to activity recognition are also common (Mitra and Acharya 2007; Turaga et al. 2008). We do not consider these approaches here as they require specifically deployed ambient infrastructure and the methods differ significantly from activity recognition using on-body sensors.



**Fig. 1** The adaptive activity recognition chain (adARC, the overall system) builds upon the classical activity recognition chain (ARC, bottom left box). The ARC is depicted with five sensors and the typical processing stages. Data fusion is illustrated here at the feature, classifier, or decision level. The outcome of the recognition chain is the recognized activity that is used in an activity-aware application with which the user interacts (bottom right). The adaptive activity recognition chain adds to the ARC the components of self-monitoring, adaptation strategies, and exploitation of available feedback

(gray area) which interacts with the ARC, the user, and external systems (gray arrows). Self-monitoring identifies relevant changes in the activity recognition system's performance or the situation in which it operates. Accordingly, adaptation strategies control the parameters of the activity recognition chain to perform in the current situation. The adARC capitalizes on available feedback to guide its adaptation. Feedback sources include the user, the activity-aware application, and external systems. The adARC is a closed-loop dynamical system where the user is in the loop

activity recognition chain (ARC, see Fig. 1 and for details Sect. 2).

A key assumption underlying current ARCs is that there is a mapping between sensor signal patterns and activity classes that is known at design-time and remains identical at run-time. ARCs may tolerate some signal variability (e.g. hand gestures corresponding to a specific activity are always slightly different) but this must be taken into account at design time.<sup>3</sup> This assumption is unrealistic for real-world use of activity recognition, as envisioned in pervasive, mobile and wearable computing. In such open-ended environments, changes that are unpredictable can occur. Even with predictable changes, it may be unrealistic (experimental- or cost-wise) to collect design-time datasets comprising all the variability against which the recognition system must be immune at run-time.

<sup>3</sup> This is usually realized by collecting activity datasets comprising all the variability likely to be observed at runtime. This is costly for the developer of the system and the subjects providing activity datasets, as they must repeat a set of activities a large number of times in numerous sensor configurations. In some cases the number of possible variations cannot be predicted or is too large to create such datasets.

In particular, the ARC is challenged to deal with the following situations, where the mapping between the sensor signal patterns and the activity classes can vary at run-time and is usually hard to predict or even unknown. Yet, such variations are likely in a long-running activity recognition system:

- Placing on-body sensors in the exact same place and orientation day after day is not realistic. It limits the comfort and appeal of a wearable assistant. Sensor placement is rather likely to vary. The user may decide to change the location of a sensor-enabled device (e.g. in different pockets) or displace it from its nominal position (e.g. moving a bracelet on the arm). Sensor may also be displaced involuntarily (e.g. sensor slipping on the arm).
- The behavior of a user may change over time, e.g. due to aging (Winter et al. 1990) or increased proficiency at a task. Also, preferences and motor-action strategies are usually specific to an individual (Lester et al. 2006).
- The sensing infrastructure may change over time. New sensors may be introduced that are unforeseen at design-time. For instance, the user may buy a new sensor-enabled piece of clothing, or the infrastructure of a building may be upgraded with new sensing capabilities.

These new sensors provide information that may be relevant to the recognition problem and thus that should be exploited, yet current ARCs cannot take advantage of them.

- Finally, we argue that in our ever more sensorized environments, for increased user comfort, ease of system deployment, and scalability, the paradigm of activity recognition should shift from using sensors specifically deployed for an application, to using sensors that just happen to be present around the user. We refer to this as opportunistic activity recognition, capitalizing on recent advances in opportunistic sensing (Conti and Kumar 2010). This requires to develop a new kind of ARC that are able to cope with—and even take advantage of—the highly dynamic and unpredictable availability of resources at run-time (see Roggen et al. 2009; 2011a) for an insight into our approach).

## 1.2 Contribution

In this work we argue for a shift from a design-time statically defined ARC to a run-time *adaptive ARC* to address the limitations of the state of the art.

We first detail the state of the art ARC derived from representative examples of activity recognition systems in Sect. 2, and explain how state of the art approaches attempt to address its limitations.

In Sect. 3 we then introduce and formalize a pattern analysis (data processing) architecture extending current ARCs and that allows for a wide range of realizations of adaptive activity recognition systems: the *adaptive activity recognition chain* (adARC). The adARC is inspired by the principles of autonomous computing, and organizes a class of solutions to adaptive activity recognition by extending the ARC with *self-monitoring*, *adaptation strategies* and *exploitation of external feedback* as key components. The adARC is a closed-loop dynamical system whereas the ARC is an open-loop system. It adapts its future behavior (e.g. changing classifier decision boundaries) based on past classification results and activity occurrences. Thus it is well suited to adapt an activity recognition system at run-time when hard to predict changes may occur.

We demonstrate this architecture in two specific cases illustrating different aspects of adaptation. In Sect. 4 we show an adARC with unsupervised classifier self-adaptation where, upon re-occurring activity instances, class decision boundaries are adjusted through self-supervised online learning. We show that self-adaptation increases activity recognition accuracy when sensors are displaced on body segments compared to a non-adaptive approach.

In Sect. 5 we show an adARC with principles of autonomous evolution that allows an activity recognition

system to expand onto sensor nodes newly introduced in the system. These new sensor nodes are initially not capable of activity recognition. Through repeated interaction with the pre-existing system, they autonomously learn to recognize activities. This allows the activity recognition system to operate in the new sensor environment without the system's designer or the user's intervention.<sup>4</sup> It can confer fault-tolerance or self-repair capabilities to ambient intelligence environments,<sup>5</sup> or reduce ambient intelligence deployment effort. This supports scalable, robust and long-term operation of activity-aware systems.

We discuss the results in Sect. 6. In particular, the adARC architecture provides a descriptive frame suitable to accommodate other adaptive activity recognition systems. It allows the description and comparison of methods in a coherent and modular manner. We show that recent adaptive recognition systems by other groups can be described within the adARC architecture. We also argue that capturing the data processing structure of an adaptive activity recognition system in a generic architecture will support the development of software framework specifically dedicated to host the pattern analysis methods required for adaptive activity recognition. We discuss the benefits and challenges of the adARC and outline new research directions. Finally, we conclude in Sect. 7.

## 2 Related works

In this section we review a few representative works in human activity recognition systems. From these work, we derive a common data processing architecture which is followed by most work and which we refer to as the ARC. We evidence the limitations of the ARC in coping with activity recognition in open-ended environments or situations where the user's motor-action strategies or preferences change over time. We explain how state of the art methods attempt to address this, while retaining the ARC architecture, and we explain the ultimate limitations thus resulting.

### 2.1 Activity recognition: representative works

A large number of methods for activity recognition have been proposed by the wearable, mobile and pervasive computing communities. These methods were applied to

<sup>4</sup> That is the designer of the system is not involved when a new sensor configuration appears: the system can autonomously use the new resource after some time. Also, the user is not requested to explicitly provide training data for the system to be able to use the new sensor.

<sup>5</sup> We use ambient intelligence to describe activity- (or context-) aware systems that rely on sensors placed on-body, in object, or in the environment for activity recognition.

activities ranging from simple isolated gestures or modes of locomotion up to complex and hierarchical activities. A large number of sensors suitable for on-body usage have been proposed, including acceleration sensors, microphones, or inertial measurement units. A few representative examples of this diversity are provided here and further examples can be found in Bao and Intille (2004):

- the recognition of complex manipulative gestures performed by industrial workers on a car body to check its functioning (Stiefmeier et al. 2008), with gestures including checking the hood latch mechanism, checking the seat sliding mechanism, and checking the spacing between doors and car body, from sensors including seven inertial measurement units;
- the recognition of seven modes of locomotion (sit, stand, walk, walk upstairs, walk downstairs, ride elevator up, ride elevator down) from one accelerometer (Lester et al. 2006);
- the recognition of the assembly steps of a shelf or a mirror from accelerometers (Blanke and Schiele 2010), and the recognition of nine wood-making activities (hammering, sawing, filing, drilling, sanding, grinding, screwing, using a vise, operating a drawer) from one accelerometer and a microphone (Ward et al. 2006);
- the recognition of five hand gestures (square, cross, circle, fish, bend) for HCI from one accelerometer (Kallio et al. 2006);
- the recognition of sports activities in a fitness room from inertial sensors (Kunze and Lukowicz 2008).

The design of a recognition system starts by the selection of a set of sensors based on the activities to recognize. Many on-body sensor modalities can be used. Motion sensors (inertial measurement units or accelerometers) are among the most common sensing modalities (Kallio et al. 2006). Another very common sensor is the microphone (Ward et al. 2006), as many human activities generate characteristic sound patterns. Other modalities include, e.g. textile integrated sensors (Tognetti et al. 2006), or electromyography (Chen et al. 2007). A key design choice is to select sensors that discriminate well the activities of interest, that are comfortable for the user, and that minimize the computational complexity of the data processing, to ensure low-power and miniaturized implementation. In Roggen et al. (2010b, 2011b) we present a more exhaustive list of sensors used for activity recognition. Given these sensors, signal processing, machine learning, or reasoning techniques are used to infer the activities from the sensor data. Despite the wide variety of sensors and activities, most of the representative work cited here uses a common data processing architecture. We refer to it as the ARC. It is a roughly common processing structure that has emerged across most published work in activity recognition (Bao

and Intille 2004; Figo et al. 2010; Ward et al. 2006). We detail it in the next section.

## 2.2 The activity recognition chain

The ARC infers the activities that the user performs based on the data from the body-worn sensors. At design time, the ARC is devised based on the activities or gestures to infer, and the selected on-body sensors (see Fig. 1). Exemplary activities or gestures performed by users at design time (i.e. a training dataset) are used to define activity models and optimize the operating parameters of the ARC. At run-time, the ARC essentially “compares” the streaming sensor signals to the activity models. It identifies sensor patterns matching sufficiently the activity models to indicate that an activity has been “spotted”. The ARC operates as follows:

*Sensor data acquisition* A time series corresponding to the sensor data is obtained. Since sensors can provide multiple values (e.g. an acceleration sensor provides a 3D vectorial acceleration), or multiple sensors are jointly sampled a vectorial notation is used.

$$S = \{s_0, s_1, s_2, \dots\}$$

*Signal pre-processing.* The time series  $S$  leads to a pre-processed time series  $P$ :

$$P = \{p_0, p_1, p_2, \dots\}$$

Typical transformations are calibration or de-noising.

*Segmentation of the data stream* into sections of interest likely to contain a gesture. Segment  $i$  is delimited by its start time  $t_i^s$  and end time  $t_i^e$  within the time series, yielding a segmented time series  $W_i$ :

$$W_i = \{p_{t_i^s}, \dots, p_{t_i^e}\}$$

A common type of segmentation technique is the sliding window (for periodic movements) or energy-based or rest-position based segmentation, when the user performs isolated gestures or returns to a rest position between gestures.

*Feature extraction* Features are computed on the identified sections to reduce their dimensionality and discriminate activities of interest. The result is a feature vector  $X_i$ :

$$X_i = \Psi(W_i)$$

*Classification* of the feature vector  $X_i$  into an output class (activity)  $c_i$ :

$$X_i \rightarrow c_i, p_i$$

Usually, classification also yields an indication as to the confidence in the resulting class. This is often a probability  $p_i$  with Bayesian approaches, and many classifiers can be calibrated to provide probabilistic outputs (Cohen and Goldszmidt 2004).



*Decision fusion* combines multiple information sources (multiple sensors, or multiple classifiers operating on one sensor) into a decision about the activity that occurred.

*“Null-class” rejection* In cases where the confidence in the classification result is too low, the system may discard the classified activity  $i$  based on  $p_i$ , in the simplest case by comparison to a threshold or using statistical approaches.

Before operation, the classifiers used in the ARC are trained using a training set  $D$  containing data instances (feature vectors)  $\omega$  and the corresponding label  $\gamma$ :

$$D = \{(\omega_i, \gamma_i)\}_{i=1}^N$$

Other parameters, such as the thresholds to segment activities or reject the null class, or the set of features are also optimized prior to operation. Once specified and trained, the ARC remains unchanged throughout operation of the activity-aware system. In order to tolerate some run-time signal variability despite the static nature of the ARC, the training set  $D$  must comprise the variability likely to be seen at run-time.

A wide range of method can be used at each stage. It is outside of the scope of this paper to mention them all. A few of the most common methods for features extraction and segmentation are reviewed in Figo et al. (2010). The selection of other parameters and methods depends on the characteristic of the activities. For activities that are of periodic nature (e.g. walking, running, bicycling, rowing, hammering, tightening a screw) a sliding window segmentation is generally used. The features that are used are selected in the frequency domain to capture the repetitive nature of the activity (e.g. zero or mean crossing rate, power spectrum, dominant frequency, Figo et al. 2010). These features are selected so that the activities form separable clusters in the feature space. The typical classifiers then used to distinguish these activities include support vector machines (Qian et al. 2010), decision trees, k-nearest neighbor or naive Bayes classifiers (Randell and Muller 2000). When “activities” are static postures (e.g. stand, sit, lie, when taking specific postures in a rehabilitation scenario, or when pointing a location) a sliding window approach is also commonly used with time-domain or statistical features (e.g. limb angle or angle between several limbs, mean of acceleration). Similar classifiers are used as for periodic activities. When the activities are sporadic (i.e. they are short and occur interleaved with other activities which the system does not need to recognize) then segmentation and classification techniques that take the temporal unfolding of the sensor signal into account are used. These segmentation and classification techniques include, e.g. hidden Markov models (HMMs) (Deng and Tsui 2000; Starner et al. 1998), dynamic time warping (Ko et al. 2005; Stiefmeier et al. 2008), methods

based on feature similarities (Keogh et al. 2001), or neural networks (Yang et al. 2008).

Generally multiple sensors are improving recognition of complex real-world activities (Stiefmeier et al. 2008). Multiple sensors are often combined with ensemble classifiers (Polikar 2006). Further methods are mentioned in Figo et al. (2010), Bao and Intille (2004) and Roggen et al. (2011b).

### 2.3 Limitations of current approaches

Regardless of the specific methods used, the ARC require a mapping between the sensor signals and the activity classes that is known at design-time and remains identical at run-time. ARCs may tolerate some signal variability but it must be taken into account at design-time. This is not suitable for real-world activity recognition in open-ended environments, as envisioned in pervasive, mobile and wearable computing. There, unpredictable changes tend to occur.

The dominant approach to cope with variability in the sensor-signal to activity-class mapping, given a static ARC, is to build generic activity models. Improved tolerance to on-body sensor placement variability has been investigated by collecting training datasets from all the on-body positions of interest, and extracting features discriminative of the activities of interest on all body locations (Lester et al. 2006). Another approach is multistage classification, where first on-body sensor placement (Kunze et al. 2005) and orientation (Kunze et al. 2009) is detected in order to select an ARC appropriate for the current sensor configuration. Features that are robust to displacement can also be designed using body models (Kunze and Lukowicz 2008). Robustness to variability in motor-action strategies between users and within users is also generally tackled by collecting rich datasets covering the variability likely to occur during system operation (Lester et al. 2006). Bio-mechanical models can also be used (Parvini and Shahabi 2005). Building more generic activity models is experimentally costly as it requires acquiring data from all sensor configurations that are likely to occur at run-time and from a large number of users to cover all the motor-action variability that underlies the richness of human activities. In some cases it may even not be possible to foresee the variability likely to occur at runtime. Generic models also may limit the number of classes that can be distinguished, as they tend to lead to overlapping class distributions in the feature space. An initial calibration phase may be used to adjust the system to new operating conditions. This has been investigated in speech recognition (Tang et al. 2008), EEG-based brain-computer interfaces (del R Millán 2004), writing recognition (Huang et al. 2009), and recently in activity recognition in wearable computing (Ohmura et al. 2009). The calibration requires user supervision and

therefore such approaches are not well suited for an unobtrusive system.

So far there has been little work in devising systems capable of exploiting new sensors without any training data, and adapting to changing resources. Theoretical insights from transfer learning have the potential to fill this gap (Pan and Yang 2008). A statistical “concept matching” was proposed to infer the meaning of a new sensor compared to pre-existing ones as a way to transfer activity recognition systems across different, but related, smart homes (van Kasteren et al. 2010). Semi-supervised learning approaches were proposed to collect sparse labels in daily life to annotate and train recognition systems (Stikic et al. 2009). Such approaches have been proposed for a more convenient design-time training of recognition systems. They do not address the autonomous run-time exploitation of newly discovered resources. Other fields have considered the issue of handling changing resources, but not for activity recognition. Nevertheless it is worth mentioning frameworks in pervasive computing pursuing self-organized data ecologies to ensure availability of service (Bicocchi et al. 2009). In modular robotics, self-assembly of new modules is pursued to allow robots to self-reconfiguration and self-repair (Gross and Dorigo 2008). In bio-inspired electronics, new resources provide self-repair and self-replication capabilities (Stauffer et al. 2001). In artificial life and bio-inspired systems, the growth of multi-cellular systems onto new resources is a key component towards scalability and robustness (Roggen et al. 2007; Streichert et al. 2003).

### 3 Adaptive activity recognition chain

Autonomous adaptation has been previously proposed to allow for systems to operate in complex, hard to predict or changing situations in image processing (Vanzella et al. 2004), evolvable hardware (Miller 2003; Roggen et al. 2007) or evolutionary robotics (Floreano and Keller 2010). It belongs to the broader class of optimization methods in uncertain environments (Jin and Branke 2005) or in dynamic environments (González et al. 2010). These methods are a foundation for autonomous operation (Kephart and Chess 2003). A common underlying characteristics of these approaches are usually closed-loop dynamical systems, running continuously, and constantly adapting their future behavior based a monitoring of their own internal states, behaviors, external inputs, past decisions, and if available a “reward” signal.

Inspired by the principles of autonomous computing, we present here a novel pattern analysis architecture for the problem of adaptive activity recognition where the

sensor-signal to activity-class mapping is subject to adaptation during operation. Thus, upon detection of activities, the ARC may adapt its behavior (e.g. change its class decision boundaries) or structure (e.g. include additional sensors in the processing chain). The ARC thus becomes a closed-loop dynamical system. We refer to it as *Adaptive Activity Recognition Chain* or *adARC*. We illustrate the structure of the adARC in Fig. 1. As is the case with the ARC, the adARC does not define a specific set of methods, but rather defines processing principles. The adARC builds on top of a classic ARC. It includes in addition system self-monitoring, adaptation strategies, and exploitation of user or external feedback, in a closed-loop dynamical system. We describe below the general function of each of these elements. In the next sections we exemplify two specific instances of adARCs.

#### 3.1 System self-monitoring

Self-monitoring estimates the suitability of the system at recognizing activities in the environment where it currently operates. This can be used to guide system adaptation, rate the confidence of the system’s decisions, or prompt the user for action. Self-monitoring assumes there is no external *ground truth* that can be used to compare the effective system behavior against the desired behavior. Thus, the system must observe its own dynamics. Heuristics, change detection and statistical approaches may be used for self-monitoring (see Sect. 6 for a discussion of self-monitoring methods). In essence, self-monitoring provides a signal guiding adaptation.

#### 3.2 Adaptation strategies

The adaptation strategies adjust the parameters of the adARC in order to perform under the current operating conditions. Relevant methods include, e.g. adaptive filtering, evolutionary computation approaches, reinforcement learning, or adaptive classifiers. Each stage of the adARC may be subject to adaptation. For instance, the set of sensors participating to activity recognition may be updated to replace faulty sensors. Classifier decision boundaries may be adjusted through incremental learning, and decision fusion may update the weights assigned to individual classifiers.

Formally, the adaptation rule by which the model  $\xi_{c_i}$  of class  $c_i$  is updated at the  $i$ th activity instance can be expressed as:

$$\xi_{c_i}(i+1) = f(L, \xi_{c_i}(i), \mathbf{X}_i, c_i), \quad (1)$$

$L$  is the learning rate and controls the trade-off between adaptation rate and stability of the models.

### 3.3 User and external feedback

In pervasive and wearable computing the user is an actor in the context-aware system and he has the possibility to provide input to the system, for instance by a mobile user interface. Such feedback can be used to guide the system adaptation according to the user's preferences. Current systems rarely attempt to exploit user feedback as this is generally obtrusive.<sup>6</sup> However, this is a matter of devising adaptation strategies that can take advantages of minimalistic forms of feedback, provided only sporadically by the user. In some cases, implicit feedback can be captured as the user's reaction to the behavior of the system may indirectly indicate how the system performs (e.g. a user being frustrated by a gestural interface may tend to perform more nervous gestures). In Sect. 6 we give example of forms of feedback using explicit user interaction, and implicit feedback that is provided without conscious user intervention via a brain-computer interface.

This component of the adARC has for goal to collect occasional sources of feedback from the user or other systems that are suitable to guide the system adaptation. Thus it comprises interface design for the acquisition of explicit feedback, or the implicit inference of feedback from suitable sensor modalities. A typical feedback may be for the user to signal moments when the system did not behave as expected, and optionally to indicate the desired behavior. When an activity recognition system is part of a larger infrastructure, complementary sources of information may provide this feedback. For instance a calendar or meeting minutes provide information about the presence of a person in a meeting that can be used as ground truth feedback for an activity recognition system (see e.g. Lovett et al. 2010) for the use of a calendar as ground truth information).

## 4 adARC for unsupervised self-adaptation

We present an adARC with online unsupervised classifier self-adaptation as adaptation strategy (see Fig. 2). Upon re-occurring context occurrences, the class decision boundaries are adjusted to better reflect the class statistics, effectively adapting to class drift in the feature space, akin to an expectation maximization principle.

We show this adaptive strategy in the recognition of activities despite variability in sensor placement (e.g.

<sup>6</sup> Exceptions are the use of "experience sampling" and semi-supervised learning approaches, where users sporadically label their activities as they are executing them (Stikic et al. 2009). This is however a form of input used to train the system at design time, rather than an online use of feedback to adapt the system's behavior at run-time.

sensor slipping). This slipping typically leads to class displacement in the feature space which affect the recognition accuracy if the activity models are not adapted.

### 4.1 adARC characteristics

The ARC underlying the adARC is a nearest class center (NCC) classifier capable of incremental learning. NCC is commonly used in wearable computing due to its low complexity and its suitability for low-power embedded devices (Roggen et al. 2006).

Self-monitoring controls the operation of the adARC. Under normal operation the adARC behaves as a non-adaptive ARC (trained to operate with a pre-defined sensor position). We simulate here self-monitoring by automatically enabling adaptation whenever sensor displacement occurs. This is comparable to a user noticing a degrading system performance and triggering the self-adaptation. Alternatively self-monitoring could enable adaptation when the sensors are first worn (displacements compared to nominal position are expected each time a sensor is worn). The latter two alternatives map to the user or external feedback envisioned in the adARC architecture.

In adaptive operation the system continuously classifies the feature vectors  $X_i$  yielding classification results  $c_i$  and adapts the activity model using this "self-labeled" sample  $(X_i, c_i)$  with supervised online learning. For the NCC classifier the online learning function is:

$$C_{c_i}(i+1) = (1-L) \cdot C_{c_i} + L \cdot X_i \quad (2)$$

with  $C_c$  the center of class  $c$  and  $L$  the learning rate. In the following  $L$  is constant:  $L = 0.3$ .

### 4.2 Validation on a fitness activity dataset

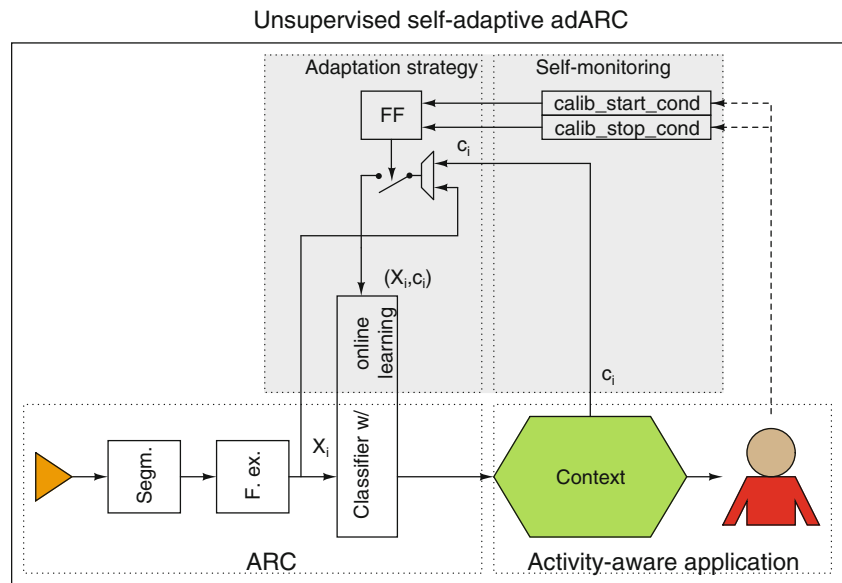
We characterize this adARC on the recognition of physical activities in a fitness scenario, with the NCC classifier and parameters indicated above. The occurrence of fast and repetitive movements may easily lead to sensor displacements.

We simulate the sensor displacement. We recorded the acceleration of the left leg for six typical aerobic movements (Fig. 4) from ten wireless acceleration sensors<sup>7</sup> at the subjects leg (Fig. 3). We placed the sensors at equal intervals and with the same orientation. An experienced subject copied the movements of a teacher shown in a video. The video, containing six activity classes of equal duration, lasted 4:22 min. The subject repeated the session five times.

<sup>7</sup> Triaxial accelerometers, Analog Devices ADXL330, sampled at 64 Hz. The sensors are open hardware and described in Roggen et al. (2010a).



**Fig. 2** Principle of the adARC with unsupervised classifier self-adaptation (gray the extension compared to a traditional ARC). Upon recognition of an activity instance  $c_i$ , the adaptation strategy consists in re-training the ARC classifier using online learning with the self-labeled data sample  $(X_i, c_i)$ . This is a form of expectation maximization. Self-monitoring controls the start and stop of the adaptation process. Optionally, user feedback may enable adaptation

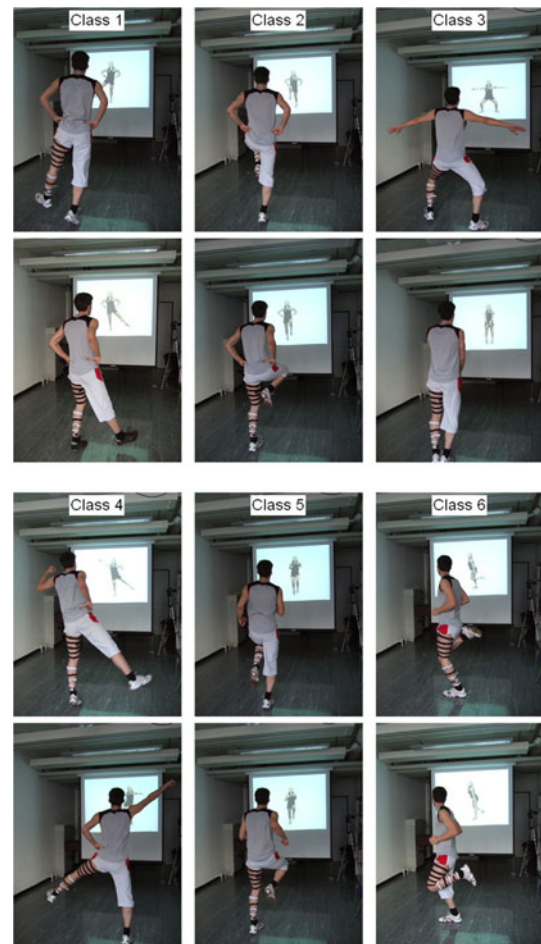


**Fig. 3** Placement of ten wireless acceleration sensors in the fitness scenario: five at the thigh and five at the lower leg



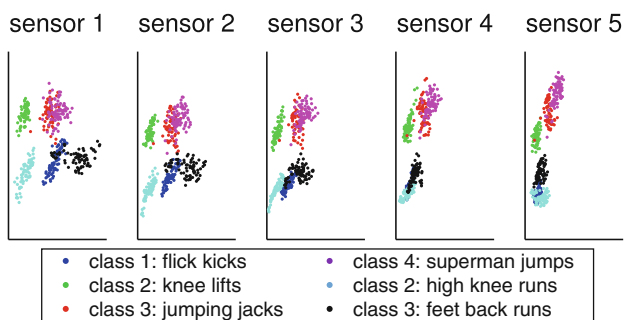
For the data of each sensor we calculated the acceleration magnitude and extracted mean and variance features based on a sliding window of 8 s with two-thirds of overlap.<sup>8</sup> The class distribution in the feature space is depicted in Fig. 5. The class distributions are more similar for adjacent sensor positions than for sensor positions further apart.

We simulate the adARC with classifier self-adaptation by training classifiers on the data from sensor position  $s$  and by using data obtained from sensor position  $t$  for adaptation (using Eq. 2) and testing (we consider the 5 sensors on the lower leg, similar results are obtained for the upper leg). We apply a threefold cross validation, using two folds to adapt the classifier on the new sensor position and one fold to test the adapted classifier model. The data samples used for the adaptation are picked randomly from all classes and are not presented in any specific order. The



**Fig. 4** The fitness scenario includes six classes: (1) flick kicks, (2) knee lifts, (3) jumping jacks, (4) superman jumps, (5) high knee runs, (6) feet back runs. For each class, the extent of the body movements is shown on two rows

<sup>8</sup> We selected on purpose a two-dimensional feature vector for the lesser computational costs incurred in wearable activity recognition systems, and at the same time it simplifies visualization of the feature space.

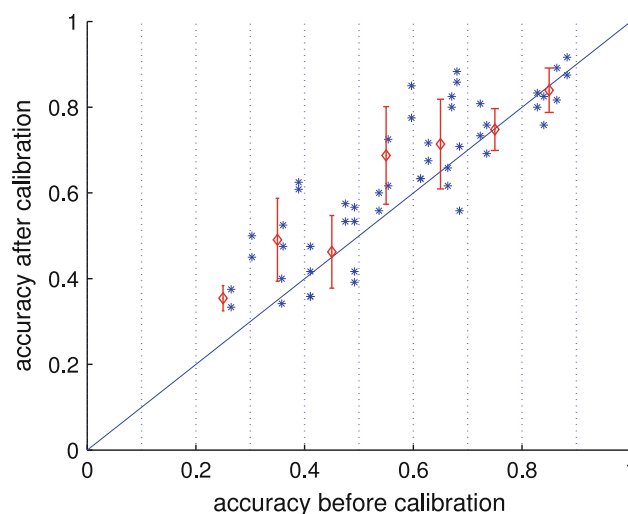


**Fig. 5** Class distributions in the feature space with visible differences from sensor to sensor. Each point represents one activity instance of the dataset

accuracies obtained when training a classifier on one body location and testing it on the same location ( $t = s$ ) is in average 83.0%. If we test on the direct neighboring sensors  $|t - s| = 1$  the average accuracy drops to 65.7%. If we test on sensor positions which are even further apart ( $|t - s| > 1$ ) the accuracy of the classifiers trained on  $s$  decreases to 42.0%.

The adaptation results are illustrated in Table 1 and Fig. 6. In the figure, all the points above the diagonal represent configurations where adaptation was beneficial. The adaptation of classifiers operated on displaced sensor positions on the lower leg is beneficial in most of the cases (average of 13.4% relative performance improvement on displacements to the immediate neighboring position, 20.5% on further away positions). Good classifiers on displaced sensor position (in this case above 70%) are less likely to benefit from adaptation.

The conditions under which self-adaptation is beneficial is function of the separability between the classes and the amount of class displacement in the feature space with respect to the nominal distribution (Förster et al. 2009). Well separated activity classes tend to benefit more from adaptation than when class distributions overlap. On the other hand, the improvement potential brought about by self-adaptation is lesser with well separated classes as classifiers tend to be more robust and have a higher initial



**Fig. 6** Accuracies after adaptation versus the accuracies before adaptation for all sensor displacement combinations for the full fitness dataset

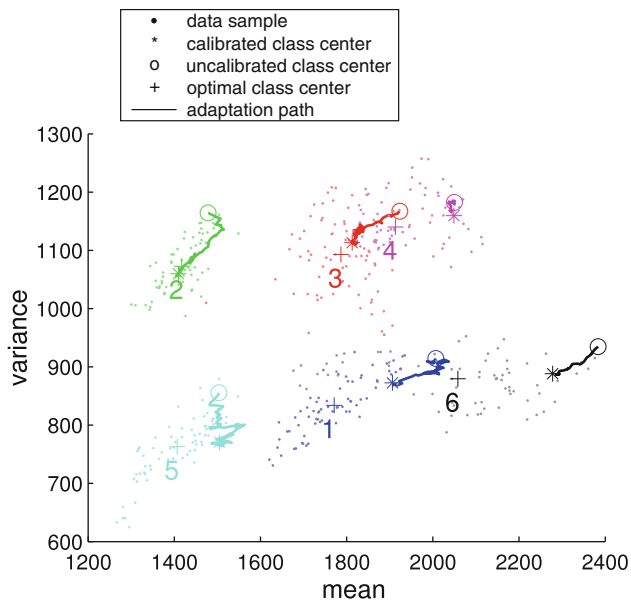
accuracy before adaptation. This is illustrated in Table 1 where we repeat this analysis on a reduced four-class dataset where the confusing classes 1 and 3 are removed. In this case sensor displacement has less influence on the accuracies of classifiers working on displaced sensors compared to the full dataset, due to the better separability of the classes.

In Fig. 7 we show an example of the adaptation dynamics for the full datasets. For the confused class 4 the calibrated class centers do not end up close to the optimal class center. This is a typical case where less class separation (here between classes 3 and 4) confuses the self-adaptation. Here only one class center benefits, (class 3). Some class centers (e.g. for class 6) end up quite far from the optimal class centers even though their paths seem to lead directly to the optimum. This indicates that an insufficient number of activity instances were used for the self-adaptation so that the optimal class centers could be reached (i.e. this means that sufficient operating time is required before the benefit of adaptation is fully realized).

**Table 1** Accuracies without adaptation and with adaptation with the relative improvement brought about by the adaptation

Accuracy	$t = s$		$ t - s  = 1$		$ t - s  > 1$	
	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$
Fitness dataset (6 activities)						
w/o adapt. (%)	83.0	5.7	65.7	4.1	42.0	9.1
w/ adapt. (%)	82.8	5.9	74.4	9.9	49.5	9.4
Rel. imp. (%)	-0.2	1.8	13.4	14.8	20.5	23.1
Reduced fitness dataset (4 activities)						
w/o adapt. (%)	95.1	3.4	89.4	4.8	67.3	9.5
w/ adapt. (%)	95.4	3.6	95.8	3.6	69.8	10.8
Rel. imp. (%)	0.4	4.0	7.2	5.1	4.1	12.6

The improvement is given relative to the accuracy before adaptation



**Fig. 7** Adaptation paths of the class centers during adaptation shown for a NCC classifier trained on sensor position 1 and calibrated on sensor position 2 for the full fitness dataset

In Förster et al. (2009) we characterized this approach in a HCI gesture recognition system with similar benefits, and we modeled the approach on a synthetic dataset.

## 5 adARC for autonomous exploitation of changing resources

Here we envision a system able to expand the activity recognition capabilities onto new resources discovered in the user's surrounding, akin to organic growth or self-replication investigated in simulated organisms and bio-inspired hardware (Stauffer et al. 2001; Tempesti 2007).<sup>9</sup> It may also allow for fault-tolerance or self-repair by having new resources replicating and replacing the behavior of pre-existing ones.

The adARC introduced here exemplifies a recognition system capable of coping to changes in sensor infrastructure

<sup>9</sup> A number of application scenarios benefit from this. For example, a user performs a short stay in a rehabilitation center with activity-aware assistance provided by sensors placed in the environment. He eventually expects the same level of assistance when he is at home or on the move. His own wearable system can automatically learn how to recognize and react to the relevant activities during the user's stay in the instrumented environment. This happens without user or expert intervention, using the resources that the patient happens to have in his body-area network. Another case is when labeling the user's activities must be done without the presence of an experimenter or of video cameras for privacy reasons, such as when developing a wearable assistant for activities of daily living, or to avoid conditioning the behavior of the user by the presence of the experimenter.

(Fig. 8). The adARC is distributed on several networked sensor nodes called *ContextCells*. A ContextCell contains sensor(s) and a corresponding adaptive recognition chain. When it detects an activity instance it exchanges the class label with its neighbors. Upon reception of labels from its peers, a ContextCell incrementally learns the mapping between the signal measured on its sensor and the received class label.

We demonstrate this adARC in a scenario where a wearable system, unable to recognize activities, learns autonomously to do so when the user interacts with instrumented furniture.

### 5.1 adARC characteristics

The ARC underlying the adARC is an NCC classifier (see Sect. 4). Self-monitoring encompasses networking aspects for the coordinated emergence of a sensing network so that ContextCells autonomously form a networked ensemble and can exchange information with each other. We assume that this is addressed using existing technical solutions.<sup>10</sup>

Self-monitoring also controls the learning rate to achieve a specific stability-plasticity trade-off. Here self-monitoring ensures that newly introduced ContextCells learn until a given number of activity instances of each classes are observed. Afterwards the activity models do not adapt.

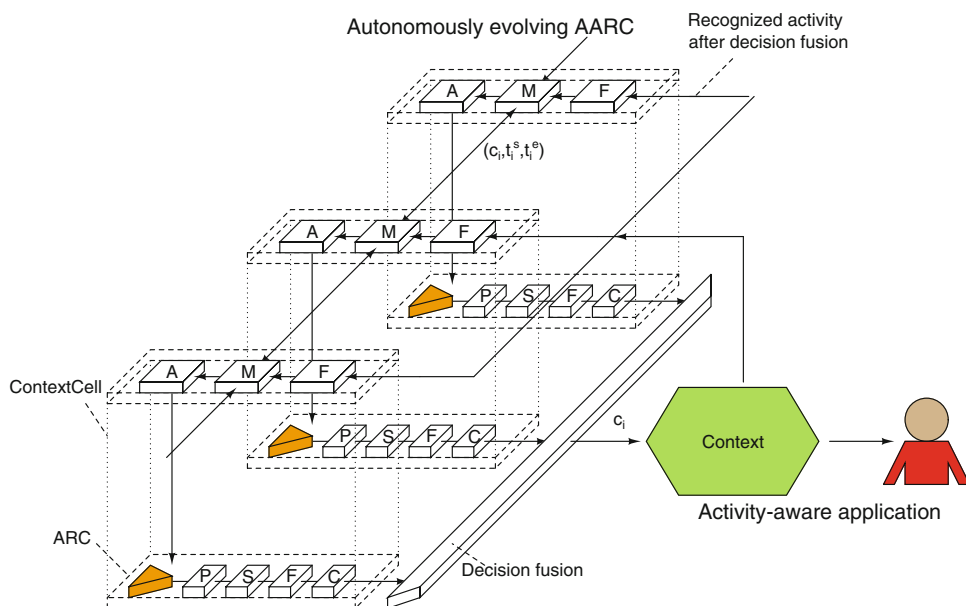
The ContextCell recognizing an activity instance  $i$  broadcasts the start and end time of the activity  $t_i^s$  and  $t_i^e$  and the label  $c_i$  associated with it. The ContextCells receiving this information compute the sensor signal feature  $X_i$  on the segment between  $t_i^s$  and  $t_i^e$  and updates the model of class  $c_i$ . Upon first reception of a label of class  $c$ , a new activity class model is created. Upon reception of a label corresponding to an existing activity model, this model is updated following Eq. 2. Here,  $L = \frac{1}{n_c + 1} \cdot n_c$  is the number of received instances for class  $c$ .

In principle, all ContextCells continuously perform activity recognition and adaptation and the ensemble of ContextCells is a dynamical system. When all ContextCells are able to recognize the same set of activities, the behavior of this adARC bears similarities to the unsupervised self-adaptive adARC but in a distributed manner.

### 5.2 Validation: expanding activity recognition to new resources

We consider a simple storage management scenario, in which a user opens and closes drawers in order to store

<sup>10</sup> For example Zigbee wireless protocol provides some self-organization capabilities, and this is the object of active research (Beal and Bachrach 2006; Dressler 2007).



**Fig. 8** The autonomously evolving adARC is distributed over several sensor nodes or ContextCells. ContextCells contain a set of sensors and a dedicated activity recognition chain (*lower layer*, the ARC processing stages are indicated by *P, F, S, C*). Self-Monitoring (*M*) allows ContextCells to form a sensing network, and exchange information (*upper layer*). The user activity may be detected by a single ContextCell, or after decision fusion (as depicted here). Upon recognition of an activity instance (Feedback *F*), the ContextCell

notifies its peers of the time of occurrence and the label of the activity (*M*). Upon reception of a notification, a ContextCell incrementally updates the sensor signal to class mapping (Adaptation *A*). In this article we analyze a case where there are two ContextCells one training the other one. Thus the learning ContextCell receives directly the notification of activity occurrence from the trainer ContextCell without going through the decision fusion block

goods in them. The activity classes are instances of “opening drawer” and “closing drawer” for 13 closely spaced (5–15 cm) drawers in a drawers set. The recognition goal is to classify these 26 tasks. Each drawers is instrumented with a trained Ambient ContextCell (a.k.a. *trainer cell*, initially capable of activity recognition) containing an accelerometer. The subject wears three wearable ContextCells (a.k.a. *learner cells*, initially not capable of activity recognition) with accelerometers on the mid-back, shoulder and mid-arm (see Fig. 9).

As a simulation dataset, we collected a minimum of ten instances of opening and closing each drawer. All sensors are synchronously recorded at 100 Hz. The signals are manually labeled and segmented. Each instance consists of a rest position, followed by the gesture (opening or closing of the drawer) and ends with the same rest position.

### 5.2.1 Ambient ContextCells

The ambient ContextCells placed on each drawer locally classify based on the acceleration data whether the drawer to which they are attached is being opened, closed or left untouched. An NCC classifier in the ContextCells is trained offline to detect the opening/closing of the drawer or no action (3-class problem).

Three feature sets are used (FS1, FS2, FS3) for comparison purposes.<sup>11</sup>

Table 2 shows the average accuracy obtained by the ContextCells mounted on the drawers for each feature set. Classification accuracy varies between individual drawers. Mechanical coupling between the drawers makes the classification challenging as the interaction with one drawer generates strong vibrations throughout the drawer set.

### 5.2.2 adARC simulation

We simulate the behavior of the adARC when the user interacts with the drawers. To account for the many ways in which drawers can be activated, the instances in the dataset are randomly shuffled in a run to simulate a casual sequence of opening and closing of the drawers. The data is partitioned into a training and test set with a 4–1 size ratio between them. We perform 2,000 simulation runs and average the results.

<sup>11</sup> The features outline characteristics of the signal linked to the action performed on the drawers. They are: (1) variance of the magnitude of the acceleration vector in five signal windows; (2) variance of the x, y and z components of the acceleration vector in the whole signal; (3) mean of the magnitude of the acceleration vector in five signal windows.





**Fig. 9** Setup: 13 drawers equipped with ambient ContextCells (trainer) and 3 wearable ContextCells (learner) on body

The sensor data corresponding to each instance of the training set are successively presented to the ContextCells on the drawers. In Fig. 10 we show the activities that are recognized by each ambient ContextCell on the drawers in one run. Each time an ambient ContextCell detects an opening or closing, it transmits the action and drawer number. The ContextCells may make conflicting classifications. When labels conflict, one label is chosen at random for the training of the wearable ContextCell. Some drawer actions may also be undetected.

We analyze the behavior of the mid-back wearable ContextCell. The tilt of the mid-back sensor relates to the bending of the subject when he reaches the various drawers. We selected as feature vector the average tilt of the acceleration sensor with respect to the vertical axis in five signal windows, thereby capturing the temporal sequence of movement of the user.

In Fig. 11 we show the temporal evolution of the NCC classifier on the wearable ContextCells. New class centers appear and their position is adjusted as more activity instances are recognized by the ambient ContextCells. In the initial steps many new centroids appear and there is major displacement of existing ones. Then the centroids tend to reach stable positions.

The performance of the wearable ContextCells is evaluated at each time step on the test set. In Fig. 12 we show the evolution of the average accuracy with the three feature sets used on the ambient ContextCells, and the upper bound on the performance with accurate ground truth labels. The performance of the wearable ContextCells relates to the capacity of the ambient ContextCells at providing accurate activity labels. However, the longer the interaction with the environment and the better the classification accuracy becomes. This outlines that it is important that the ContextCells operate on a long period of time. This is the

situation envisioned for this adARC, as changes in sensor environments tend to occur on long time scales.

Overall, the activity recognition capabilities can be expanded from the ambient to the on-body ContextCells by repeated interactions between them. Eventually, as the wearable ContextCell is capable of recognizing the relevant activities of the user, the same context-aware assistance can be provided to the user also outside of instrumented environments. Further improvements may be obtained by considering labels as noisy labels and taking their label confidence into account in the learning rate (Angluin and Laird 1988). We present in Calatroni et al. (2009) further details on the technical realization of the ContextCells. In Calatroni et al. (2011) we show how this adARC can be used to transfer the capability to recognize modes of locomotion from existing to newly deployed and untrained sensors on the body, thus reflecting the situation that a user faces when he buys a new and untrained sensorized gadget or garment, yet wants to keep the recognition capabilities pre-existing in his current body-worn system.

## 6 Discussion

We discuss hereafter the two exemplary adARCs presented in this paper. We then show in Sect. 6.3 that recently proposed adaptive systems introduced by other groups fit within the adARC architecture as well. We finally discuss new research directions.

### 6.1 adARC for unsupervised self-adaptation

The unsupervised self-adaptive adARC is worth considering when a generic model cannot be obtained either due to hard to predict or hard to model run-time variability, or when a generic model leads to class confusions. In such cases, it may also reduce design-time data collection and modeling effort.

We expect this adARC to be advantageous under these assumptions: the run-time variability cause the existing classifier to under-perform on the new class distributions, the classes remain separable in the new distribution, the adaptation rate is matched to the speed at which the class distributions change. We expect these assumptions to be verified in a set of real-world problems. Besides adaptation to changing sensor position, this adARC may be applicable to: gradual changes in sensor characteristics (e.g. sensitivity of a textile-integrated strain sensor as it degrades over time due to washing and stress), gradual change in user behavior (e.g. due to motor learning, aging, recovery from injuries), adaptation to different users performing the same activity but with some differences in motor-actions. This method is not applicable to large (with respect to the distance between



**Table 2** Classification accuracy of the ambient ContextCells for feature sets FS1–FS3

	Ambient ContextCell (trainer) accuracy			Wearable ContextCell (learner) accuracy	
	Individual best (%)	Individual worst (%)	Combined (%)	Average (%)	$\sigma$ (%)
GT	100	100	100	79	8.1
FS1	88	33	79	69	8.2
FS2	100	40	72	67	8.6
FS3	87	45	34	35	7.8

“Individual best and worst” indicate the performance of the best and worst performing Ambient ContextCell for the 3-class individual drawer action classification problem. “Combined” indicates the resulting performance of all 13 ambient ContextCells at detecting which of the 13 drawers is being opened or closed (single 26-class problem). The performance of the Wearable ContextCell (26-class problem) located on the mid-back alone is indicated after the system autonomously expanded to it. “GT” is the upper bound performance of the wearable ContextCell assuming perfect ambient ContextCell recognition

the classes in the feature space and the size of the activity clusters), non-gradual changes.<sup>12</sup> We showed this in the case of the sensor displacement. Small displacements (up to 10 cm) are tolerated, whereas adaptation to displacement on a longer distance, or across limb segments is not possible, due to the significant change of the class mapping in the feature space. As with other unsupervised approaches, the adaptation rate needs to be adjusted to ensure the system remains stable. This adARC bears some similarity to growing neural gas or online k-means but it remains a classification method, rather than a distribution representation or clustering method. This adARC can build upon any classifier capable of incremental learning. We showed this with incremental versions of the NCC, kNN and SVM classifiers in Calatroni et al. (2011).

## 6.2 adARC for autonomous exploitation of changing resources

This adARC may be used to train a wearable system without the presence of experimenters (e.g. for privacy reasons). It may also confer fault-tolerance and self-repair capabilities to ambient intelligence environment. Deployment effort can be reduced, as the activity recognition can autonomously expand onto new resources, as well as expand to new activity classes, without re-programming of the sensor nodes in the system. In larger-scale ambient intelligence environments these characteristics are beneficial to support scalability and robustness, and thus support long-term operation of activity-aware systems in open-ended environments.

Currently the set of features that is used is defined at design-time based on expert knowledge of the expected type of activities and sensor kinds. Future work should

<sup>12</sup> In that case, user-feedback could be exploited to update the class centers and then come back to the unsupervised mode. We explored implicit user feedback with a brain-computer interface system in Förster et al. (2010a) and an explicit user feedback by means of a push-button in Förster et al. (2010b).

consider how the set of features can itself evolve autonomously at run-time. Evolutionary computation approaches for feature extraction have been proposed (Zhang and Rockett 2009). The objective function might be defined as the degree of agreement between the classification results of multiple nodes. Other approaches of interest include, e.g. Learn++ (Polikar et al. 2001) to adapt both features and classifiers, incremental PCA to adapt the feature space (Zhao et al. 2006). This adARC also bears some similarities to transfer learning or inductive learning (Taylor and Stone 2009).

## 6.3 Other instances of adARCs

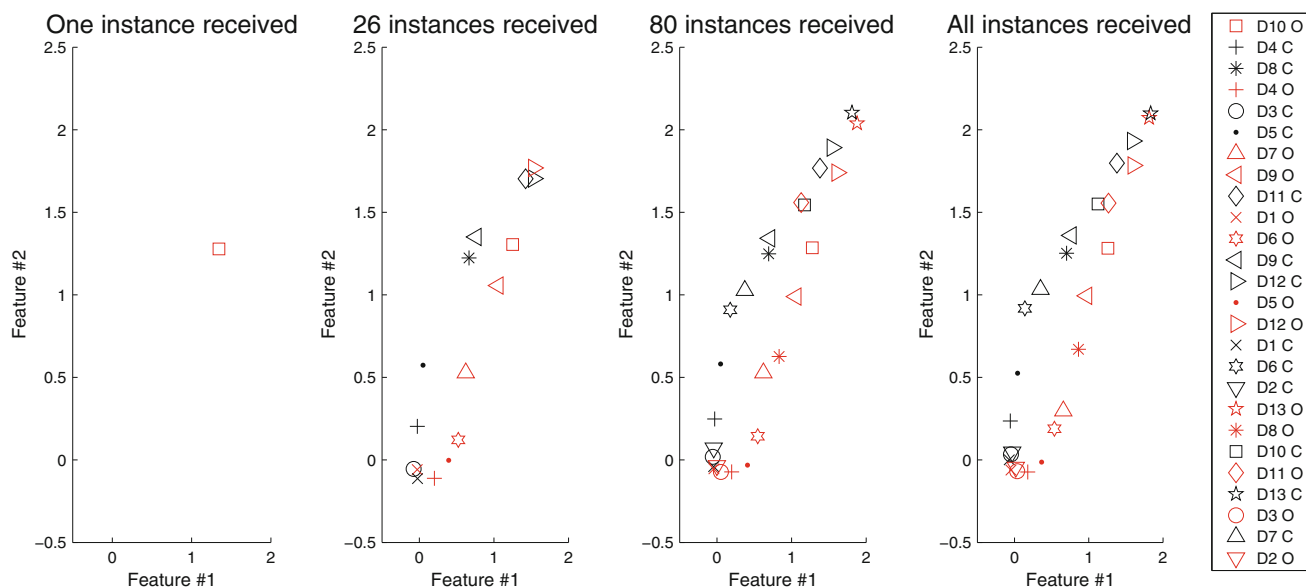
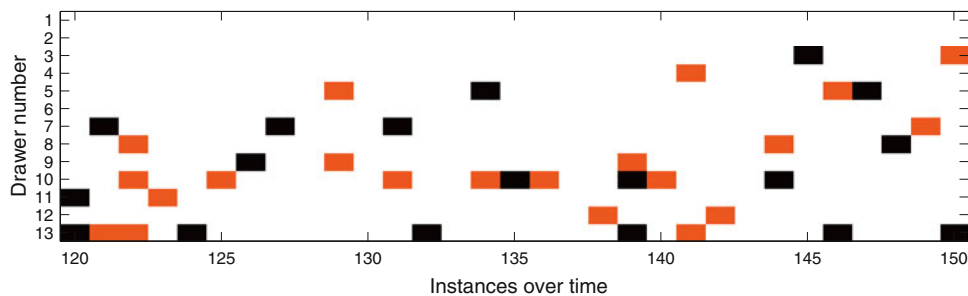
The adARC architecture can be used to organize the solution to other kinds of adaptive activity recognition systems.

In Zappi et al. (2008) we presented an adARC distributed over a dynamic set of sensor nodes (as in Sect. 5). It makes a trade-off between the recognition performance of an activity recognition system and the operation time of the system (energy use). Self-monitoring assesses whether the current set of sensors allows to reach the desired classification accuracy, and uses an adaptation heuristics to recruit an adequate set of sensors to reach the desired power-performance trade-off at run-time. Finally user or application feedback can control the desired system performance or operation time at run-time.

Recently other groups have proposed related systems although they do not explicitly formalize them as an instance of a broader pattern analysis architecture geared at providing adaptivity to activity recognition systems. Bayati et al. (2011) present another approach to cope with sensor displacement. As the adARC presented in Sect. 4 it relies on self-monitoring of the class distribution in the feature space, and the adaptation strategy consists of expectation maximization.

Rossi et al. (2010) presented a pervasive computing system for unsupervised speaker identification with

**Fig. 10** Recognition of the drawer activation in one simulation run by the ContextCells on the drawers (*light opening, black closing*). Feature set 1 is used. Also visible are conflicting classification results (e.g. at time 120) and mis-detection of drawer activations (e.g. at time 133)



**Fig. 11** 2D projection of the class centroids in the wearable ContextCells at four instants of the simulation showing the evolution of the learning process. The legend shows the order in which the

centroids appear. Displacement of centroids is visible, e.g. for the opening of drawer 7 (D7 O, *triangle*) between the first apparition of the class and the end of the simulation

autonomous incremental learning of new speakers in a collaborating set of microphones. Their approach follows the adARC presented in Sect. 5 with each microphone implementing a functionality akin to the ContextCell. Their approach addresses autonomous adaptation to new classes, rather than to new resources.

## 6.4 New research directions

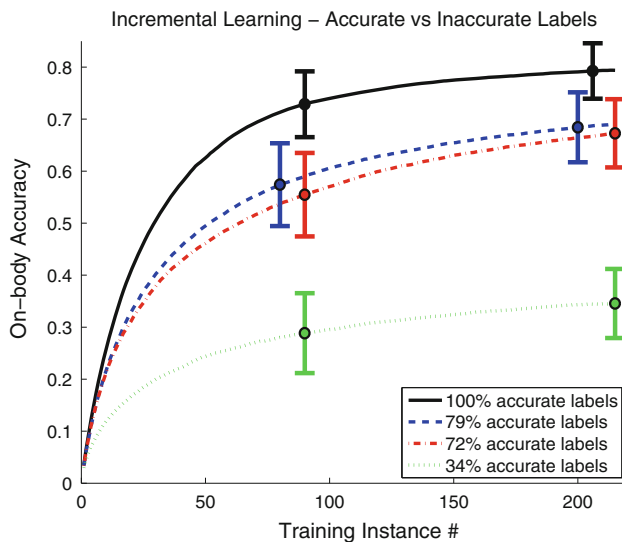
### 6.4.1 Self-monitoring

Self-monitoring approaches that detect relevant run-time changes in system operation are required to control adaptation. It is especially important for the adARCs presented here to detect trends in the activity class distributions in the feature space. To our knowledge there are few methods specific to the problem domain of activity recognition. In Betta and Pietrosanto (2000) the authors differentiate between methods based on physical redundancy and analytical redundancy. Translating this to activity recognition, the first approach may correspond to measuring the degree

of agreement in ARCs individually applied to different sensors. The second approach may correspond to modeling the typical distribution of the activity classes and detecting a significant trends towards a deviation from this model. A number of approaches exist to detect unexpected changes, anomalies or deviations from expected behavior (Chandola et al. 2009). Sagha et al. (2011) in particular presented an approach suitable for activity recognition in sensor networks.

### 6.4.2 Adaptation strategies

The adARC relies on a strategy to adapt activity models at run-time. Thus an important research direction is the design of classifiers that have some of these properties: incremental learning, possibility to guide adaptation by an external signal, robustness to hidden context and concept drift (Widmer and Kubat 1996), and low complexity. As alternatives to the NCC classifier used here several other classifiers may be considered: incremental SVM (Cauwenberghs and Poggio 2000), incremental ensembles (Freund and Schapire 1997),



**Fig. 12** Evolution of mid-back wearable ContextCell classification accuracy with the number of interaction with the drawers. Results presented for the three feature sets FS1–FS3 and for perfect ambient ContextCell activity recognition (100% accurate labels) as an upper bound. Average of 2,000 runs

online hidden Markov models (Stiller and Radons 1999), or neural networks (Polikar et al. 2001), etc. In wearable or pervasive computing the computational costs and memory requirements must be minimized for implementation on miniature sensor nodes (Roggen et al. 2006). New adaptation strategies may also be pursued towards: adaptation to the motor patterns of a specific user, adaptation to changing user preferences, as well as adaptation to variable run-time goals (e.g. changing performance target as in Zappi et al. 2008).

#### 6.4.3 User feedback

Exploiting user feedback requires further investigation of the kind of feedback and the mobile input modalities suited for pervasive and mobile computing. It also requires research on the methods suitable to exploit this feedback. In particular, the information gained from the user feedback should be maximized, but the feedback should be minimally obtrusive, thus simple, infrequent, and minimizing cognitive load.

Active learning can be used to prompt for labels when the system benefits most from the user input (Settles 2009). We presented how to use a minimally obtrusive explicit feedback in Förster et al. (2010a). The feedback consists in the user sporadically tagging the system’s behavior as “correct” or “incorrect”. In order to exploit this form of feedback we devised a novel classifier capable of adaptation with a true/false sporadic feedback (Förster et al. 2010a).

Besides this explicit feedback we also considered implicit feedback, where the adaptation of the system is guided by the user’s unconscious brain signals (Förster et al. 2010a). The system detected error-related potentials (ErrP) from an electroencephalography cap. These signals arise when the user observes an incorrect behavior of a system he interacts with. Thus, the brain signal replaces the explicit button press. This form of implicit feedback may also be considered a form of self-monitoring that takes advantage of the presence of the user in the system.

#### 6.4.4 Performance metrics

Classical ARCs are characterized offline on pre-recorded datasets. adARCs are dynamical systems. They are influenced by feedback from the user or interactions with other context aware systems. This feedback is usually not predictable and thus the adARC must be characterized online. This poses new simulation and experimental challenges. For instance, a user-adaptive system cannot be characterized on a pre-recorded dataset to optimize the system’s parameters, as the behavior of the user would likely be different with each new set of parameters. However, the degree of satisfaction of the user with respect to the system’s behavior can be compared for various parameter sets in an online evaluation. When dealing with changing sensing environment, methods ought to be compared on the same variations, thus calling for new simulation approaches, or experimental testing on a larger number of instances of variations. Traditional machine learning performance metrics (Ward et al. 2006) must be expanded to include the aspects of dynamical systems such as stability or adaptability, and the convergence conditions.

## 7 Conclusion

Motivated by the limitations of current activity recognition approaches in dealing with a number of variations that can be expected in the long-term use of pervasive, mobile and wearable activity-aware systems, we presented a new pattern analysis architecture that allows for adaptation mechanisms: the *adaptive* activity recognition chain (adARC). It attempts to relax the need for generic design-time activity models in favor of an autonomous adaptation of the system to runtime conditions. The adARC extends the classical activity recognition approaches with self-monitoring, adaptation strategies, and the inclusion of user or external feedback. Self-monitoring detects relevant changes in the sensor signal to activity mapping or in the sensor environment. Adaptation strategies adjust accordingly the recognition system to operate in the new conditions. Finally, the user is part of the activity-aware system

in most scenarios. The system ought to exploit user feedback to guide adaptation, as well as the feedback from other context aware systems. The adARC forms a closed-loop dynamical system. The adARC defines an *architecture* suitable to host a variety of processing methods. Its main point is to organize a class of solutions to the problem of adaptive activity recognition in changing situations. It allows to describe adaptive activity recognition systems in a coherent manner. It allows to modularize the investigation of new methods, and it helps identify new research directions within the adARC elements.

We presented two instances of adARCs. The first is an unsupervised self-adaptive adARC. This approach can increase the accuracy of an activity recognition system in a scenario where sensors are unpredictably displaced on the body, compared to a non-adaptive system. This adARC may be applicable to other problems such as adaptation to degrading sensors, to changing user behavior, or to different users.

The second adARC provides autonomous adaptation capabilities in changing sensor configurations. It is distributed over several sensor nodes and allows to extend the capability of a system to recognize activities to new nodes introduced into the environment. We showed that this adARC can be used to train a wearable system without manual intervention while the user performs in a pre-existing ambient intelligence environment. Eventually, the same activity-aware assistance can be provided outside of the instrumented environment by the wearable system. This adARC may also confer fault-tolerance and self-repair capabilities to ambient intelligence environment, or reduce deployment efforts. These characteristics are important to support long-term operation of activity-aware systems in open-ended environments. These two adARCs also play a key role in the development of activity recognition systems operating in opportunistic sensor configurations as envisioned in Roggen et al. (2009, 2011a), thus using efficiently resources that just happen to be available, rather than requiring specific sensor deployments.

We discussed other works that follow the adARC structure. These results show that the adARC allows to frame a set of solutions to the problem of real-world deployment of activity recognition systems. The adARC supports the investigation of further adaptive activity recognition systems by modularizing research along methods for self-monitoring, adaptation strategies and exploitation of feedback. This may lead to a pool of building block methods that can be combined to form adARCs.

Finally, current software frameworks dedicated to activity recognition are mostly targeting a static ARC. The formalization of the adARC captures a wide range of adaptive activity recognition systems, yet in a well-defined data processing architecture. This supports the development

of generic frameworks specifically designed to host adaptive activity recognition algorithms.

**Acknowledgments** The project OPPORTUNITY acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 225938.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

## References

- Angluin D, Laird PD (1988) Learning from noisy examples. *Mach Learn* 2(4):343–370
- Bao L, Intille SS (2004) Activity recognition from user-annotated acceleration data. In: *Proceedings of the 2nd international conference on pervasive computing*, pp 1–17
- Bayati H, del R Millán J, Chavarriga R (2011) Unsupervised adaptation to on-body sensor displacement in acceleration-based activity recognition. In: *IEEE international symposium on wearable computers (ISWC)*
- Beal J, Bachrach J (2006) Infrastructure for engineered emergence on sensor/actuator networks. *IEEE Intell Syst* 21(2):10–19
- Benini L, Farella E, Guiducci C (2006) Wireless sensor networks: enabling technology for ambient intelligence. *Microelectron J* 37(12):1639–1649. doi:10.1016/j.mejo.2006.04.021
- Betta G, Pietrosanto A (2000) Instrument fault detection and isolation: state of the art and new research trends. In: *IEEE Trans Instrum Meas* 49(1):100–107
- Bicocchi N, Baumgarten M, Bruglja N, Kusber R, Mamei M, Mulvenna M, Zambonelli F (2010) Towards self-organized data ecologies for pervasive services: the knowledge networks approach. *IEEE Trans Syst Man Cybern Part A Syst Humans* 40(4):789–802
- Blanke U, Schiele B (2010) Remember and transfer what you have learned—recognizing composite activities based on activity spotting. In: *Proceedings of the 14th IEEE international symposium on wearable computers (ISWC '10)*. Seoul, Korea
- Calatroni A, Villalonga C, Roggen D, Tröster G (2009) Context cells: towards lifelong learning in activity recognition systems. In: *Proceedings of the 4th European conference on smart sensing and context (EuroSSC)*. Springer, Berlin
- Calatroni A, Roggen D, Tröster G (2011) Automatic transfer of activity recognition capabilities between body-worn motion sensors: training newcomers to recognize locomotion. In: *8th international conference on networked sensing systems*. IEEE Press
- Cauwenberghs G, Poggio T (2000) Incremental and decremental support vector machine learning. In: *Proceedings of the NIPS*, pp 409–415
- Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. *ACM Comput Surv* 41(3):15:1–15:58
- Chen X, Zhang X, Zhao ZY, Yang JH, Lantz V, Wang KQ (2007) Hand gesture recognition research based on surface EMG sensors and 2d-accelerometers. In: *11th IEEE international symposium on wearable computers*, pp 11–14, 11–13 October 2007
- Cohen I, Goldszmidt M (2004) Properties and benefits of calibrated classifiers. In: *Proceedings of the knowledge discovery in databases (PKDD)*



- Conti M, Kumar M (2010) Opportunities in opportunistic computing. *Computer* 43(1):42–50
- Davies N, Siewiorek DP, Sukthankar R (2008) Special issue: activity-based computing. *IEEE Pervasive Comput* 7(2):20–21
- del Millán RJ (2004) On the need for on-line learning in brain-computer interfaces. In: *IEEE international joint conference on neural networks*, vol 4, pp 2877–2882
- Deng J, Tsui H (2000) An HMM-based approach for gesture segmentation and recognition. In: *15th international conference on pattern recognition*, vol 3. IEEE Computer Society Press, pp 679–682
- Dressler F (2007) *Self-organization in sensor and actor networks*. Wiley, New York
- Figo D, Diniz PC, Ferreira DR, Cardoso JMP (2010) Preprocessing techniques for context recognition from accelerometer data. *Pervasive Mobile Comput* 14(7):645–662
- Floreano D, Keller L (2010) Evolution of adaptive behaviour in robots by means of darwinian selection. *PLoS Biol* 8(1):e1000292 doi:10.1371/journal.pbio.1000292
- Förster K, Roggen D, Tröster G (2009) Unsupervised classifier self-calibration through repeated context occurrences: is there robustness against sensor displacement to gain? In: *Proceedings of the 13th IEEE international symposium on wearable computers (ISWC)*, pp 77–84
- Förster K, Biasiucci A, Chavarriga R, del R Millán J, Roggen D, Tröster G (2010a) On the use of brain decoded signals for online user adaptive gesture recognition systems. In: *Proceedings of the 8th international conference on pervasive computing*. Springer, Berlin, pp 427–444
- Förster K, Monteleone S, Calatroni A, Roggen D, Tröster G (2010b) Incremental kNN classifier exploiting correct-error teacher for activity recognition. In: *Proceedings of the ninth international conference on machine learning and applications (ICMLA)*, pp 445–450
- Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci* 55(1):119–139
- González JR, Masegosa AD, García IJ (2010) A cooperative strategy for solving dynamic optimization problems. *Memet Comput J* 3(1):3–14
- Gross R, Dorigo M (2008) Self-assembly at the macroscopic scale. *Proc IEEE* 96(9):1490–1508
- Huang Z, Ding K, Jin L, Gao X (2009) Writer adaptive online handwriting recognition using incremental linear discriminant analysis. In: *International conference on document analysis and recognition*, pp 91–95. doi:10.1109/ICDAR.2009.28
- Jin Y, Branke J (2005) Evolutionary optimization in uncertain environment—a survey. *IEEE Trans Evol Comput* 9(3):303–317
- Kallio S, Kela J, Korpipää P, Mäntyjärvi J (2006) User independent gesture interaction for small handheld devices. *Int J Pattern Recognit Artif Intell* 20(4):505–524
- Keogh E, Chu S, Hart D, Pazzani M (2001) An online algorithm for segmenting time series. In: *Proceedings of the IEEE international conference on data mining*, pp 289–296
- Kephart JO, Chess DM (2003) The vision of autonomic computing. *Comput Mag* 36(1):41–50
- Ko M, West G, Venkatesh S, Kumar M (2005) Online context recognition in multisensor systems using dynamic time warping. In: *Proceedings of the conference on intelligent sensors, sensor networks and information processing*, pp 283–288
- Kunze K, Lukowicz P (2008) Dealing with sensor displacement in motion-based onbody activity recognition systems. In: *Proceedings of the 10th international conference on ubiquitous computing*
- Kunze K, Lukowicz P, Junker H, Troester G (2005) Where am I: recognizing on-body positions of wearable sensors. In: *Strang T, Linnhoff-Popien C (eds) Proceedings of the 1st international workshop on location and context-awareness*. Springer-Verlag, Berlin, pp 257–268
- Kunze K, Lukowicz P, Partridge K, Begole B (2009) Which way am I facing: Inferring horizontal device orientation from an accelerometer signal. In: *Proceedings of the 13th international symposium on wearable computers (ISWC)*. IEEE Press, New York, pp 149–150
- Lester J, Choudhury T, Borriello G (2006) A practical approach to recognizing physical activities. In: *Fishkin K, Schiele B, Nixon P, Quigley A (eds) Proceedings of the Pervasive Computing*, vol 3968. Springer, Berlin, pp 1–16. doi: 10.1007/11748625\_1
- Lovett T, O’Neill E, Irwin J, Pollington D (2010) The calendar as a sensor: analysis and improvement using data fusion with social networks and location. In: *Proceedings of the 2010 ACM conference on ubiquitous computing*, pp 3–12
- Mann S (1998) Humanistic computing: “wearcom” as a new framework and application for intelligent signal processing. *Proc IEEE* 86(11):2123–2151
- Miller J (2003) Evolving developmental programs for adaptation, morphogenesis, and self-repair. In: *Banzhaf W, Christaller T, Dittrich P, Kim JT, Ziegler J (eds) Proceedings of the 7th European conference on artificial life*. Springer, Heidelberg, pp 256–265
- Mitra S, Acharya T (2007) Gesture recognition: a survey. *IEEE Trans Syst Man Cybern Part C Appl Rev* 37(3):311–324. doi: 10.1109/tsmcc.2007.893280
- Myers B, Hollan J, Cruz I, Bryson S, Bulterman D, Catarci T, Citrin W, Glinert E, Grudin J, Ioannidis Y (1996) Strategic directions in human-computer interaction. *ACM Comput Surv* 28(4):794–809
- Ohmura R, Hashida N, Imai M (2009) Preliminary evaluation of personal adaptation techniques in accelerometer-based activity recognition. In: *Proceedings of the 13th IEEE international symposium on wearable computers: late breaking results*
- Pan SJ, Yang Q (2010) A survey on transfer learning. *IEEE Trans Knowl Data Eng* 22(10):1345–1359
- Parvini F, Shahabi C (2005) Utilizing bio-mechanical characteristics for user-independent gesture recognition. In: *Proceedings of the 21st international conference on data engineering workshops*, p 1170. doi:10.1109/ICDE.2005.302
- Polikar R (2006) Ensemble based systems in decision making. *IEEE Circuits Syst Mag* 6(3):21–45
- Polikar R, Upda L, Upda S, Honavar V (2001) Learn++: an incremental learning algorithm for supervised neural networks. *IEEE Trans Syst Man Cybern Part C Appl Rev* 31(4):497–508. doi:10.1109/5326.983933
- Qian H, Mao Y, Xiang W, Wang Z (2010) Recognition of human activities using SVM multi-class classifier. *Pattern Recognit Lett* 31(2):100–111
- Randell C, Muller H (2000) Context awareness by analysing accelerometer data. In: *ISWC 2000: proceedings of the 4th international symposium on wearable computers*, pp 175–176
- Roggen D, Bharatula NB, Stäger M, Lukowicz P, Tröster G (2006) From sensors to miniature networked sensorbuttons. In: *Proceedings of the 3rd international conference on networked sensing systems (INSS06)*. Transducer Research Foundation, San Diego, CA, pp 119–122
- Roggen D, Federici D, Floreano D (2007) Evolutionary morphogenesis for multi-cellular systems. *Genetic Program Evolvable Mach* 8(1):61–96
- Roggen D, Förster K, Calatroni A, Holleczeck T, Fang Y, Tröster G, Lukowicz P, Pirkl G, Bannach D, Kunze K, Ferscha A, Holzmann C, Riener A, Chavarriga R, del R Millán J (2009) Opportunity: towards opportunistic activity and context recognition systems. In: *Proceedings of the 3rd IEEE WoWMoM workshop on autonomic and opportunistic communications*



- Roggen D, Bächlin M, Schumm J, Holleczeck T, Lombriser C, Tröster G, Widmer L, Majoe D, Gutknecht J (2010a) An educational and research kit for activity and context recognition from on-body sensors. In: Proceedings of the IEEE international conference on body sensor networks (BSN), pp 277–282
- Roggen D, Calatroni A, Rossi M, Holleczeck T, Förster K, Tröster G, Lukowicz P, Bannach D, Pirkl G, Ferscha A, Doppler J, Holzmann C, Kurz M, Holl G, Chavarriaga R, Creatura M, del R Millán J (2010b) Collecting complex activity data sets in highly rich networked sensor environments. In: 7th international conference on networked sensing systems. IEEE Press, pp 233–240
- Roggen D, Lombriser C, Rossi M, Tröster G (2011a) Titan: an enabling framework for activity-aware “PervasiveApps” in opportunistic personal area networks. EURASIP J Wirel Commun Networking 2011:22. doi:10.1155/2011/172831
- Roggen D, Magnenat S, Waibel M, Tröster G (2011b) Designing and sharing activity recognition systems across platforms: methods from wearable computing. IEEE Robot Autom Mag 18(2):83–95
- Rossi M, Kusserow M, Amft O, Tröster G (2010) Collaborative real-time speaker identification for wearable systems. In: 8th international conference on pervasive computing and communications, pp 180–189
- Sagha H, del R Millán J, Chavarriaga R (2011) Detecting and rectifying anomalies in opportunistic sensor networks. In: Proceedings of the International Conference on Body Sensor Networks
- Settles B (2009) Active learning literature survey. Computer sciences technical report 1648, University of Wisconsin-Madison
- Starmer T, Weaver J, Pentland A (1998) Real-time American sign language recognition using desk and wearable computer based video. IEEE Trans Pattern Anal Mach Intell 20(12):1371–1375
- Stauffer A, Mange D, Tempesti G, Teuscher C (2001) A self-repairing and self-healing electronic watch: the biowatch. In: Liu Y et al (eds) Proceedings of the 4th international conf on evolvable systems (ICES 2001). Heidelberg, pp 112–127
- Stiefmeier T, Roggen D, Ogris G, Lukowicz P, Tröster G (2008) Wearable activity tracking in car manufacturing. IEEE Pervasive Comput 7(2):42–50
- Stikic M, Larlus D, Schiele B (2009) Multi-graph based semi-supervised learning for activity recognition. In: Proceedings of the 13th IEEE international symposium on wearable computers (ISWC '09). Linz, Austria
- Stiller JC, Radons G (1999) Online estimation of hidden Markov models. IEEE Signal Process Lett 6(8):213–215. doi:10.1109/97.774870
- Streichert F, Spieth C, Ulmer H, Zell A (2003) Evolving the ability of limited growth and self-repair for artificial embryos. In: Banzhaf W, Ziegler J, Christaller T et al (eds) Proceedings of the 7th European conference on artificial life. Springer, Heidelberg, pp 289–298
- Tang Y, Rose R (2008) Rapid speaker adaptation using clustered maximum-likelihood linear basis with sparse training data. IEEE Trans Audio Speech Lang Process 16(3):607–616
- Taylor ME, Stone P (2009) Transfer learning for reinforcement learning domains: a survey. J Mach Learn Res 10:1633–1685
- Tempesti G (2007) Biological inspiration in the design of computing systems. Proc IEEE 95(3):463–464
- Tognetti A, Carbonaro N, Zupone G, De Rossi D (2006) Characterization of a novel data glove based on textile integrated sensors. In: 28th annual international conference of the IEEE on engineering in medicine and biology society (EMBS '06), pp 2510–2513, 30 August–3 September 2006
- 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. doi:10.1109/tcsvt.2008.2005594
- van Kasteren T, Englebienne G, Kröse B (2010) Transferring knowledge of activity recognition across sensor networks. In: Proceedings of the 8th international conference on pervasive computing, pp 283–300
- Vanzella W, Pellegrino FA, Torre V (2004) Self-adaptive regularization. IEEE Trans Pattern Anal Mach Intell 26(6):804–809
- Ward J, Lukowicz P, Tröster G, Starmer T (2006) Activity recognition of assembly tasks using body-worn microphones and accelerometers. IEEE Trans Pattern Anal Mach Intell 28(10):1553–1567
- Weiser M (2002) The computer for the 21st century (reprint). Pervasive Comput 1(1):19–25
- Widmer G, Kubat M (1996) Learning in the presence of concept drift and hidden contexts. Mach Learn 23(1):69–101
- Winter DA, Patla AE, Frank JS, Walt SE (1990) Biomechanical walking pattern changes in the fit and healthy elderly. Phys Ther (United States) 70(6):340–347
- Yang JY, Wang JS, Chen YP (2008) Using acceleration measurements for activity recognition: an effective learning algorithm for constructing neural classifiers. Pattern Recognit Lett 29(16):2213–2220
- Zappi P, Lombriser C, Farella E, Roggen D, Benini L, Tröster G (2008) Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection. In: Verdone R (ed) 5th European conference on wireless sensor networks (EWSN 2008). Springer, Berlin, pp 17–33
- Zhang Y, Rockett PI (2009) A generic multi-dimensional feature extraction method using multiobjective genetic programming. Evol Comput 17(1):89–115
- Zhao H, Yuen PC, Kwok JT (2006) A novel incremental principal component analysis and its application for face recognition. IEEE Trans Syst Man Cybern Part B 36:873–886