By Daniel Roggen,
Stéphane Magnenat,
Markus Waibel, and
Gerhard Tröster

Designing and Sharing Activity-Recognition Systems Across Platforms

Wearable Computing

©IMAGE STATE

In robotics, activity-recognition systems can be used to label large robot-generated activity data sets. It enables activity-aware human–robot interactions (HRIs). It also opens ways to self-learning autonomous robots. The recognition of human activities from body-worn sensors is a key paradigm in wearable computing. In this field, the variability in human activities, sensor deployment characteristics, and application domains has led to the development of best practices and methods to enhance the robustness of activity-recognition systems. We argue that these methods can benefit many robotics use cases. We review the activity-recognition principles followed in the wearable computing community and the methods recently proposed to improve their robustness. These approaches aim at the seamless sharing of activity-recognition systems across platforms and application domains. Finally, we outline the current challenges in wearable activity recognition.

Recognizing, sharing, and reusing robot behaviors across multiple robot platforms with various similarities are challenging. Although descriptions for objects [e.g., computer-aided design (CAD) models and recognition models] and environments (e.g., geocoordinates, local coordinates, and feature maps) are largely interchangeable across different robot hardwares, robot task descriptions are typically hardware dependent. This has prevented the generation of generic data sets for robot behaviors. However, such data sets are important and underpin many of the algorithmic advances in object recognition [1]–[3] or in the creation of joint world models [4]–[7]. It has also hindered the progress of robot cognition and robot learning by preventing the robots to understand and learn from each other's actions.

Driven by the rapid progress in mobile sensing and computing, wearable computing has developed powerful methods for the

Digital Object Identifier 10.1109/MRA.2011.940992
Date of publication: 14 June 2011

automatic recognition, categorization, and labeling of human actions and behaviors from sensor data. Because of the stringent requirements dictated by user acceptance, these methods are typically robust to human variability and hardware-dependent factors, including variability in sensor type and placement. This makes them a potentially useful tool for the automatic recognition and labeling of robot behaviors and may lead to new opportunities for research in robotics. We detail three domains in which the methods of activity recognition can play a role in robotics.

Annotation of Large-Scale Activity Data Sets

Because of the ease of systematic data collection from robots and their potential usefulness for data mining, a future World Wide Web (WWW) for robots is likely to include data sets for a large number of behavioral strategies for different robotic platforms in different situations. Such data sets may, for example, include sensor readings for the walking behavior of a humanoid robot on different terrains or trajectory information for the grasp behavior of a pick-and-place robot for various target objects. Although individual robots are typically aware of their current behavior and may partially label such data, difficulties in creating comprehensive naming conventions and precise definitions for behaviors make such labels too vague to support comparative performance evaluation. Current methods for human activity recognition may allow to automatically supplement such labels by providing systematic and comparable categories for behaviors. In addition, they may be used to automatically identify underlying motion primitives, further increasing their precision and potential for data mining.

Human–Robot Interaction

Activity recognition supports natural HRI [8]. Here, we take the viewpoint that human activities are inferred from sensors worn by the user and broadcasted to the surrounding robots. The typical applications are in the domain of assistive robotics. Activity recognition for assistive uses is the typical aim of wearable computing.

Robot Self-Learning

Imagine that a human being teaches a robot by demonstration [9] (see Figure 1). Using activity recognition, the robot can supplement its known set of behaviors with an internal model to recognize different demonstrated activities [10], [11] and even build a repertoire of subgoals and obtain a hierarchical decomposition of its actions [12]. The inferred model can provide feedback for a self-learning process [13], e.g., using self-perception [14], reinforcement learning [15], or evolutionary techniques [16]. Such self-learning would allow the robot to develop its own realization of the motor commands in the light of the goal to reach. In addition, it may lead to increases in behavioral robustness. For instance, a damaged robot such as NASA's famous "Spirit" rover may not have been preprogrammed to cope with all modes of failures. However, activity recognition could provide it with insights into its actual behavioral performance despite the damage. Using continuous self-learning, this may allow it to develop a novel, effective motor-control program overcoming the problem.

Robot self-learning can also be applied across platforms. Rather than demonstrating all activities to all robots, activity recognition may allow robots to learn from each other. Given compatible perception systems, robots can directly share and reuse activity-recognition programs across platforms. Differences such as parameterizations of motor control programs can then be self-learnt. The direct sharing of an activity-recognition program is only possible for identical perception systems. However, efforts in activity recognition, which we review here, aim at methods that are robust across different perception systems [17]. In particular, a reduced set of sensors may nevertheless allow to recognize a common subset of activities [18]. Since many of today's robots share common sensors such as cameras or laser scanners, this potentially

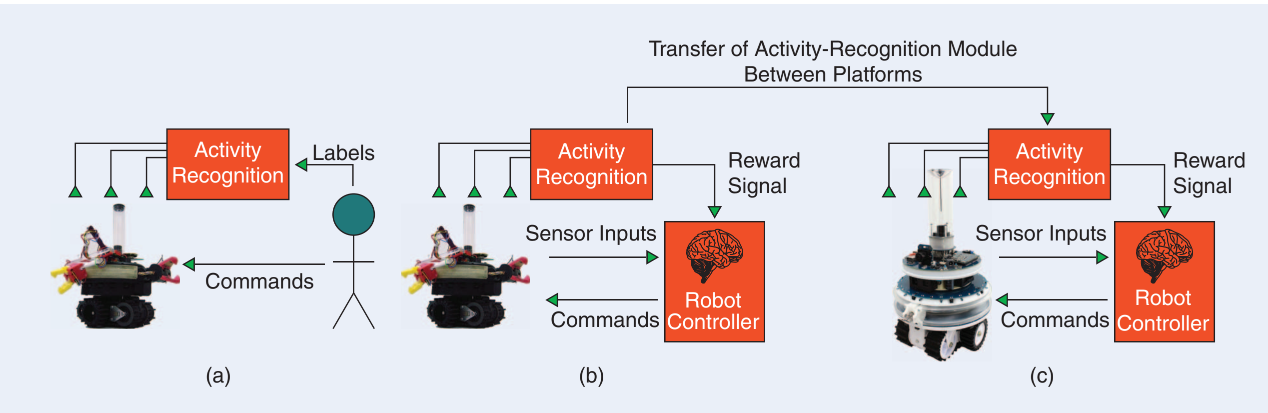

Figure 1. (a) An activity recognition system trained by user demonstration (b) guides the adaptation of the robot motor controller. (c) The recognition system can be transferred to another robot with a compatible perception subsystem and different actuators (c). This allows it to self-learn the same motor skills as the previous robot.

allows to reuse activity-recognition programs across platforms. Moreover, a robot capable of recognizing an activity using some of its sensors can learn the information content with respect to that activity in its other sensors [19]. This could be used to learn to use a new sensor. This could also lead to automatic calibration when replacing robot parts and to transparent sensor substitution in case of failures. In the context of networked robots, activity recognition may help robots identify semantic relevance of their peer's activities, enhancing collaboration in heterogeneous robot teams.

Contribution

Activity recognition is a key principle underlying wearable computing. Body-worn sensor data is interpreted to infer the user's activities and realize activity-aware applications [20], [21]. One realizes the direct parallel to robotics: the mechanical body is replaced by a human body, and wearable systems and autonomous mobile robots alike sense and interpret their environment from a first-person perspective.

Activity recognition in wearable computing is challenging because of a high variability along multiple dimensions: human action-motor strategies are highly variable, the deployment of sensors at calibrated locations is challenging, and the environments where systems are deployed are usually open ended. This has led the wearable computing community to enhance existing, and investigate new, recognition methods that cope with such variability. This article presents the principles developed for wearable activity recognition to the robotics community, emphasizing the issue of transfer and sharing of activity-recognition systems between platforms.

We introduce wearable computing in the "What Is Wearable Computing?" section. We give an overview of the approaches used for activity recognition in wearable computing along with sensors and data processing techniques in the "Wearable Activity Recognition" section. In the "Sharing Activity-Recognition Systems" section, we categorize and illustrate a few of the most relevant approaches recently proposed to share activity-recognition systems across platforms and application domains. We conclude summarizing the key insights and ongoing research challenges, indicating resources where further information about wearable activity recognition can be found in the "Conclusions and Outlook" section.

What Is Wearable Computing?

Wearable computing, as originally presented by Mann in 1996, emphasized a shift in computing paradigm [22]. Computers would no longer be machines separate from the persons using them. Instead, they would become an unobtrusive extension of our very bodies, providing us with additional ubiquitous sensing, feedback, and computational capabilities. As implied by its name, wearable computing never considered implanting sensors or chips into the body. Rather, it emphasizes the view that clothing, which has become an extension of our natural skin, would be the substrate that technology could disappear into (Figure 2). The prevalence of mobile phones now offers an additional vector for on-body sensing and computing [23].

Mann [24] and Starner et al. [25] were among the first to show that complex contextual information can be obtained by interpreting on-body sensor data and that this would lead to novel adaptive applications. A wearable system can perceive activities, defined here to include both gestures and behaviors, from a first-person perspective. This leads to new forms of applications known as activity-based computing or interaction-based computing [20], [21]. Such applications can offer information or assistance proactively based on the user's situation as well as support explicit interaction in unobtrusive ways through natural gestures or body movements.

A few application domains include industrial assistance [26], gestural inputs for human–computer interaction (HCI) [27], behavior monitoring for personalized health care [28], and movement analysis for sports assistants [29].

The kinds of activities or gestures that are recognized are wide ranging but also depend on the available sensors. Activities span low-level actions such as modes of locomotion (walking, running, and standing), postures (sitting and lying), gestures (reaching an object and opening a door), and higher-level composite activities made of sequences of

Figure 2. A typical wearable-computing system. The system comprises a see-through head-up display (HUD) in the goggles to provide the users with a contextual information, an Internet connection, an on-body computer, and sensors to infer the user's context, such as his activities and location.

actions (e.g., preparing a breakfast consists of a statistically characteristic sequence of actions). Examples include

- the recognition of complex manipulative gestures performed by industrial workers on a car body to check its functioning [30], 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 (IMUs) (see also Figures 3 and 4)
- the recognition of seven modes of locomotion (sit, stand, walk, walk upstairs, walk downstairs, ride elevator up, and ride elevator down) from one accelerometer [31]
- the recognition of the assembly steps of a shelf or a mirror from accelerometers [32] and the recognition of nine wood-making activities (hammering, sawing, filing, drilling, sanding, grinding, screwing, using a vise, and operating a drawer) from one accelerometer and microphone [33]
- the recognition of five hand gestures (square, cross, circle, fish, and bend) for HCI from one accelerometer [27]
- the recognition of sports activities in a fitness room from inertial sensors [34].

Activity recognition in wearable computing shares a number of similarities to mobile robotics.

- Sensing is performed on the human or robot body from a first-person perspective.
- Recognition of activities is directly relevant for the application at hand.
- Continuous recognition of activities is essential to allow for the adaptation of the system's behavior.
- Activities typically have a clear semantic description (e.g., reaching and grasping).

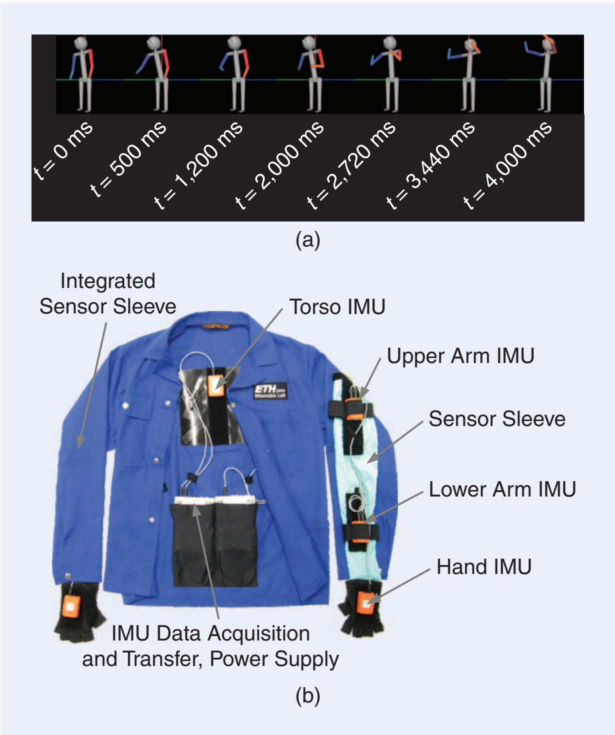

Figure 3. (a) The IMUs allow to reconstruct the user's instantaneous posture. (b) The MotionJacket allows the unobtrusive capture of the upper-body movements using seven IMUs placed on each body segments.

Wearable Activity Recognition

Activity and gesture recognition are generally tackled as a problem of learning by demonstration [33], [35]. The user is instrumented with the selected sensors and is put into a situation where he performs the activities and gestures of interest. The sensor data are acquired with ground-truth annotations describing what the user performs or experiences. The resulting data set is used to train the recognition system and test its performance. The training process consists of identifying the mapping between the user's activities or gestures and the corresponding sensor signals.

Some terminology commonly used in wearable activity recognition differs from the one used in robotics.

- *Annotation or Labeling*: This is the process by which the experimenter manually provides ground-truth information about the activities of the subject, generally, when collecting an activity data set.
- *Recognition or Spotting*: This is the actual machine identification of an activity in the data sensor stream. Activities are said to be recognized or spotted.

Sensors for Activity Recognition

Sensors are used to acquire signals related to the user's activities or gestures. User comfort is paramount. Thus, the sensors must be small, unobtrusive, and ideally invisible to the outside. The sensors are selected according to a tradeoff between wearability, computational needs, power usage, communication requirements, and information content for the activities and contexts of interest. For instance, cameras are currently seldom used in wearable computing because of the computational requirements for video analysis. Instead, sensor modalities that are computationally lighter are preferred.

Common sensor modalities are body-worn accelerometers and IMUs. Accelerometers are extremely small and have low power. The IMUs contain accelerometers, magnetometers, and gyroscopes, which allow to sense the orientation of the device with respect to a reference. The IMUs are typically placed on each body segment and allow to reconstruct a body model of the user. On-body microphones are also successfully used for activity recognition, as many human activities generate characteristic sounds (using a coffee machine and brushing teeth) [33]. Typical sensor modalities are listed in [36].

Clothing is a major platform to unobtrusively deploy on-body sensors . For instance, the IMUs can be integrated in a worker's jacket (see Figure 3). There are also ongoing efforts to develop sensorized textile fibers, which allows for truly unobtrusive garment-integrated sensing [37].

Today, the trend goes toward an increased use of multiple multimodal sensors, as this tends to increase recognition performance (see [30]). Wearable systems are also complemented by object-integrated and ambient sensors. We recently coined the term *opportunistic activity recognition* in the project OPPORTUNITY [17], funded by the seventh framework programme for research of the European Commission (EU FP7). It describes systems that make use of sensors that just happen to be available, rather than requiring specific sensor deployment. This will further address comfort issues. It also emphasizes the need for new machine-learning techniques to share activity-recognition systems across different sensor domains [17].

Activity-Recognition Chain

We refer to the activity-recognition chain (ARC) as a set of processing principles commonly followed by most researchers to infer human activities from the raw sensor data [33], [35], [38], [39] (see Figure 5).

The subsymbolic processing maps the low-level sensor data (e.g., body-limb acceleration) to semantically meaningful action primitives (e.g., grasp). Meaning is attributed to the sensor data streams by comparing them to known activity prototypes. This is realized by streaming signal processing and machine-learning techniques. The outcome of the subsymbolic processing is the event indicating the occurrence of action primitives. The ARC terminates at this stage when the activities of interest consist of simple gestures, for instance, in gestural interfaces [27].

The symbolic processing maps the sequences of action primitives (e.g., grasping and cutting) to higher-level activities (e.g., cooking). This may be realized by reasoning, expert knowledge, or statistical approaches applied to the occurrences of action primitives.

Subsymbolic processing ought to be robust to the large observed variability in sensor-signal to activity-class mapping because of changing human behaviors or sensor deployments. In wearable computing, subsymbolic processing is usually cooptimized with sensor selection to maximize comfort and recognition performance. The subsymbolic processing stages are the following [33], [35], [38] (see Figure 5):

- *Sensor-Data Acquisition:* A stream of sensor samples S is obtained.
- *Signal Preprocessing:* The sensor data stream is preprocessed. Typical transformations are calibration, denoising, or sensor-level data fusion.
- *Segmentation of the Data Stream*: The data stream is segmented into sections W that are likely to contain a gesture. Segments are identified by their start and end time in the data stream. A common type of segmentation technique is the sliding window, usually 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 segments to reduce their dimensionality, yielding a feature vector X.
- *Classification:* A classifier, trained at design time, maps the feature vector into a predefined set of output classes (activities and gestures): $X \rightarrow c, p$. Usually, a ranked likelihood p of the output classes is obtained and can be used for decision fusion.
- *Decision Fusion:* Combines multiple information sources (multiple sensors or classifiers operating on a 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 based on its likelihood. At this stage, the outcome is the detection of an action primitive A_i with likelihood p_i at time t_i.

Before operation, the classifiers used in the ARC are trained using a training set containing data instances (feature vectors) X and the corresponding activity label y. Other parameters, such as the thresholds to segment

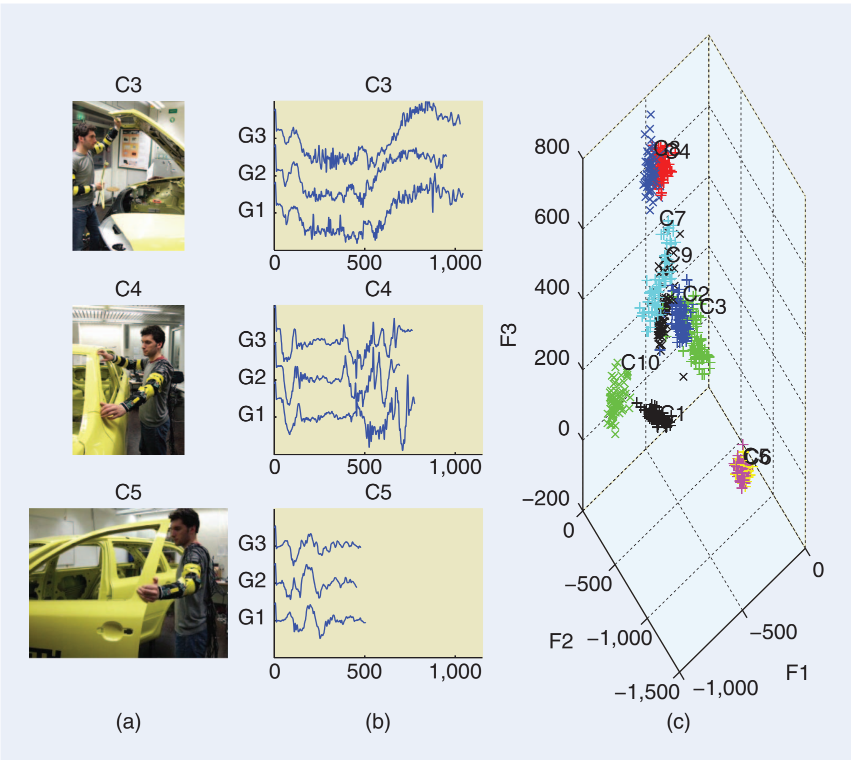

Figure 4. (a) Three activities of a car assembly scenario are shown: checking the engine hood (C3), the gap spacing between doors and car body (C4), and the opening of the front door (C5). (b) The data of an acceleration sensor placed on the right wrist are shown for three repetitions (G1–G3) of the activity. (c) After feature extraction, the sensor signals are projected into a feature space for pattern recognition. Sensor data from [18].

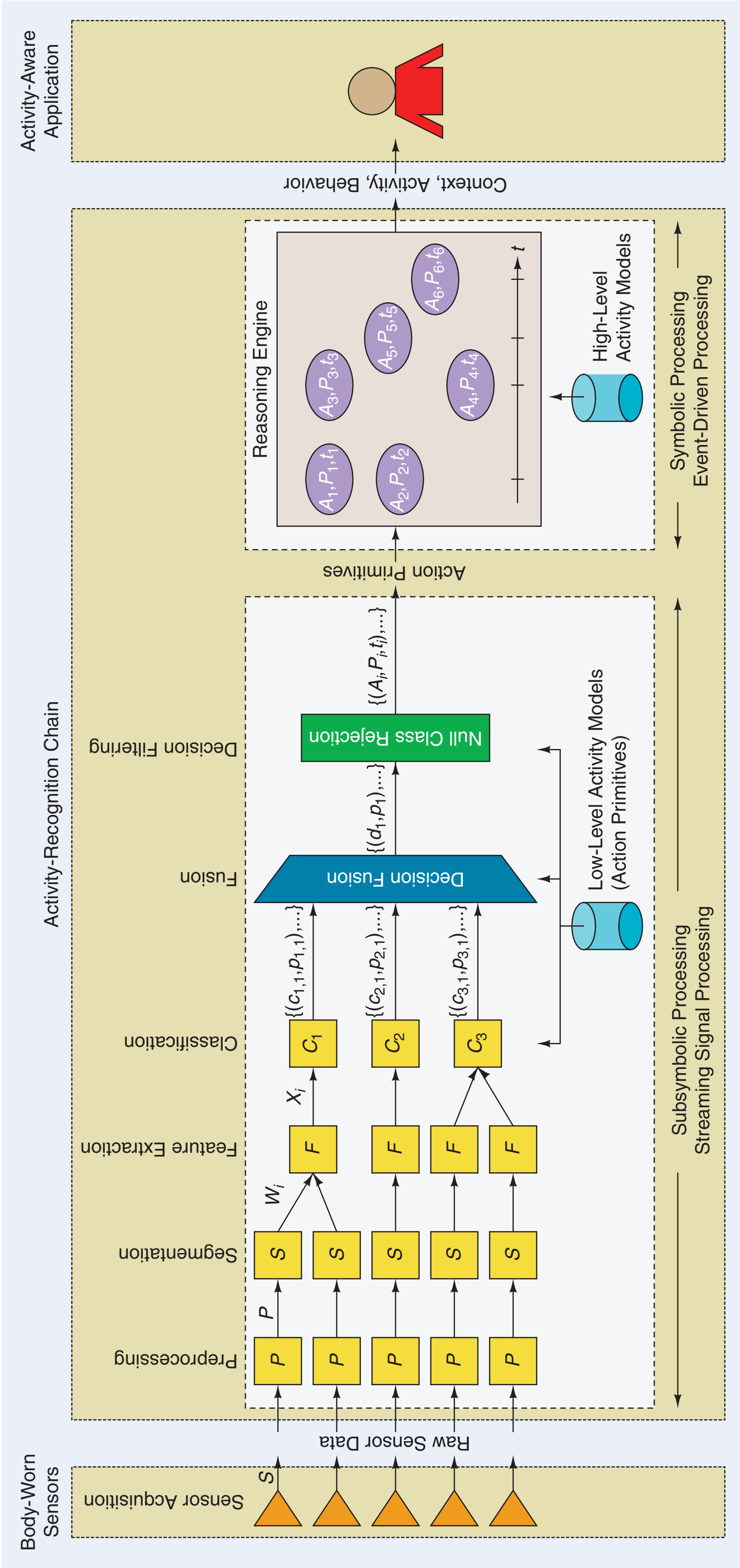

Figure 5. Processing steps used to infer activities from on-body sensors. The raw sensor data are mapped to the occurrence of action primitive (events) with signal processing and machine-learning techniques. Here, five sensors deliver data. Data fusion is illustrated at the feature, classifier, and decision levels. Symbolic processing infers higher-level activities from the occurrence of action primitives, usually with reasoning or statistical approaches.

activities or reject the null class or a set of features, are also optimized before operation.

Classifiers commonly used for activity recognition have been reviewed in [38] together with the typical features derived from acceleration signals. If the features corresponding to activities form clusters in the feature space (see Figure 4), then the classifiers that are typically used include support vector machines [40], decision trees, *k*-nearest neighbor, or Naive Bayes classifiers [41]. This is usually the case with isolated gestures and when static postures are recognized with features such as limb angles. It is also the case with periodic activities when frequency-domain features are used (e.g., walking leads to energy in specific frequency bands). When the temporal unfolding of the gesture is analyzed, such as sporadic gestures, approaches such as dynamic time warping [42] or hidden Markov models (HMMs) [25] are used. Other methods include neural networks [43] or fuzzy systems [44].

In Figure 4, we illustrate a set of activities and its corresponding sensor signals. We can note the variability in the gesture execution length and signal shape. With simple statistical features, the sensor signals can be projected in a feature space where the activities form clusters suitable for classification. During the training of the recognition chain, the selection of preprocessing steps and features aims at increasing the separation between the activity classes. Some activities are well separated, leading to accurate classification (e.g., C10, C1), while others overlap as they are more similar (e.g., C5, C6, bottom right).

Symbolic-level processing is usually event driven, with the events corresponding to activity occurrences. Higher-level activity models are thus built on event occurrences instead of raw sensor data. Approaches typically used for symbolic processing include ontological and statistical reasoning: probabilistic and temporal logic, Bayesian networks, fuzzy logic, Dempster–Shafer, and hybrid approaches [45]–[47]. Modeling and reasoning methods used for human context inference are further reviewed in [39].

High-level models are also usually derived from data recordings. Alternative approaches include the use of expert knowledge. For instance, in [30], we relied on a documented step-by-step guide for the industry workers to detect a high-level task—the assembly of a car lamp—from a sequence of action primitives.

Few works have attempted to use expert knowledge to detect complex gestures from raw sensor data, such as accelerometer readings [48]. The main challenge faced is the large inter- and intrauser variability, which is better captured by learning by demonstration approaches.

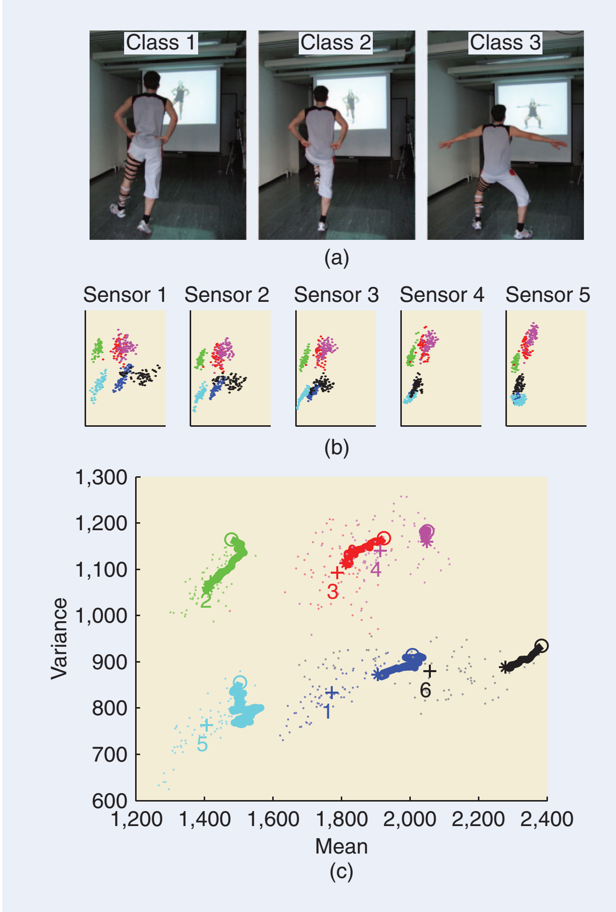

Figure 6. (a) Three of the six fitness activities performed by the subject to assess unsupervised classifier self-calibration (flick kicks, knee lifts, and jumping jacks are depicted). Ten on-body sensors at regular intervals are visible on the subject's left leg. (b) Distribution of the six fitness activity classes in the feature space. (c) Adaptation dynamics of the nearest class center classifier trained on one sensor position and deployed on another position.

Sharing Activity-Recognition Systems

Human activity recognition in wearable computing is challenging because of a large variability in the mapping of sensor signals to activity classes. This variability has multiple origins, which is shown below.

- Semantically identical action primitives (e.g., drinking from a glass) can be executed in a large number of ways (e.g., grasp with the left or right hand, while sitting, standing, or walking at various speeds). This is referred to as intrauser variability. These variations arise from personal preferences. Moreover, aging, injuries, or increased proficiency at a task also lead to variability. Figure 4 illustrates intrauser variability.
- Although different persons may be considered as robots of identical make, in practice, there is an even higher variability in action-motor strategies between users (interuser variability) than for a single user. Personal preferences, differences in expertise, body proportions, or fitness level explain this variability.
- The placement of the sensors on body cannot be done with a high precision, especially when the users deploy the sensors themselves. For comfort reasons, the user must be able to detach sensors when not needed (e.g., during sleep) and reattach them when needed or to displace them when uncomfortable. The placement of sensors in loose-fitting clothing is affected by the deformation of the garment depending on the user's activities and posture [49]. Figure 6 illustrates the effect of sensor placement on the projection of sensor data into the feature space.
- Abstracting the specific environment in which the system can recognize activities is important to ensure cost-effective deployment on a large scale. Thus, activity-recognition methods should work for a generic class of problems (e.g., in any smart home) rather than a specific instance of the problem class (e.g., a specific smart home).
- To further increase unobtrusiveness, we argue in the project OPPORTUNITY (http://www.opportunity-project.org, a Future and Emerging Technologies project funded by the European Commission in the seventh framework programme for research) to use opportunistically discovered sensors for activity recognition [17], [50]. Thus, the available sensor configuration depends on the sensorized objects users take with themselves, on the smart clothing they wear, and on the environment in which they are located. For each sensor kind and placement, there is a different sensor-signal to activity-class mapping that an opportunistic activity-recognition system should be able to abstract.

The wearable computing community has developed best practices and novel methods to deal with some forms of variability. In the following subsections, we present a selection of methods developed by various groups and ours. To share an ARC, there must be a common representation at some stage in the recognition chain. We organize the methods along the level at which the methods assume a common representation. We describe methods operating at the sensor, feature, classifier, and reasoning levels (see Figure 7).

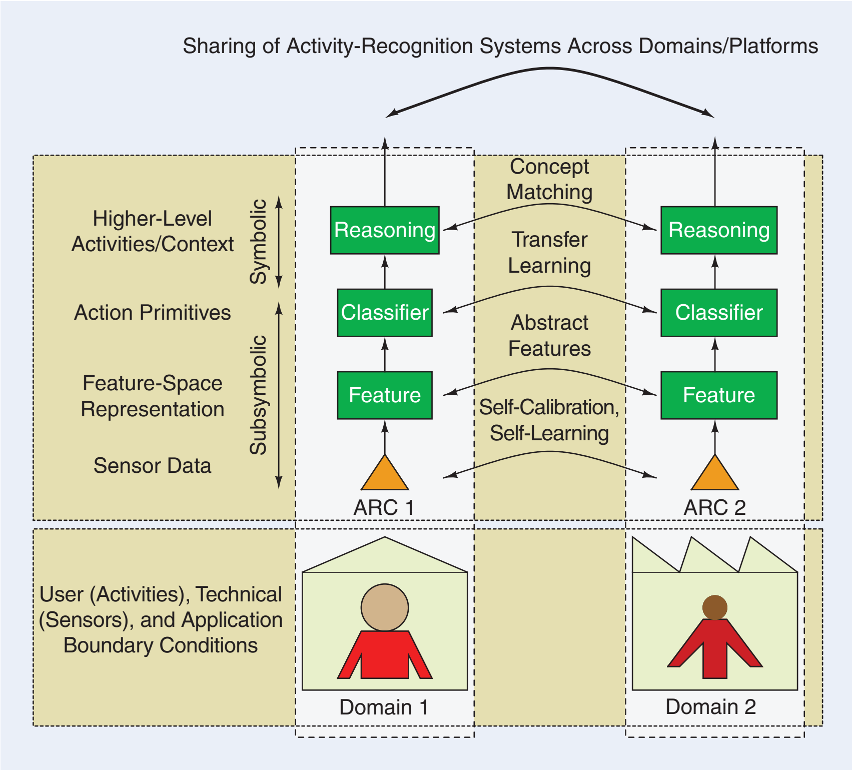

Figure 7. Representation of the level at which a common representation is assumed to share a recognition system between users (platforms) or domains.

Sensor-Level Sharing

This level focuses on training an ARC on the first platform and reusing it on the second platform. This assumes that the sensor-signal to activity-class mappings are statistically identical on two platforms. This is usually not the case in practice because of the slight variations in sensor placement and human action-motor strategies. Training an ARC on one system is referred to as a user-specific system, and it is known to show degraded performance when deployed to another user [33]. Training user-specific ARCs is costly and thus not adequate for the deployment of wearable system on a large scale.

The best practice to realize an ARC that generalizes to new situations consists in training it on a data set containing the variability to be seen when the system is deployed. By collecting a data set from multiple users, the ARC can be trained to be user independent [33]. By collecting a data set comprising multiple on-body sensor positions, the ARC can be trained to be independent of sensor placement [31].

A similar approach in learning by demonstration in robotics could lead to platform-independent activity-recognition models by demonstrating a task to multiple platforms.

The previous approach requires to foresee all the variations likely to be encountered at run time. Thus, we proposed an unsupervised self-calibration approach that removes this requirement [51]. The self-calibration approach operates as follows:

- the ARC continuously operates and recognizes the occurrence of activities/gestures
- upon detection of an activity/gesture, the corresponding sensor data is stored as training data
- the classifiers are retrained, including this new training data, using an incremental learning algorithm.

Thus, the activity models are optimized upon each activity instance to better model that activity. We demonstrated the benefits of this approach on the recognition of six fitness activities (see Figure 6) when the position of sensors on the body are displaced between the training and testing phases. The figure illustrates the sensor placement and the mapping of the activity classes in the feature space. During adaptation, the method tracks the displacement of the activity clusters in the feature space. The assumptions underlying the approach are that activities form distinct clusters in the feature space and that the speed of adaptation is matched to the speed at which the clusters shift. In [51], we argue that this approach may also be applied to cope with the slight changes in action-motor strategies because of aging or change of user. A related approach is proposed in [52].

A translation to robotics of these principles may allow the activity models to adapt when sensors or actuators deteriorate.

Feature-Level Sharing

At this level, the ARC devised for the first platform is translated to the second platform from the feature level onwards. Thus, the ARC must abstract from the specific sensors. The use case for sharing ARCs at this level include systems where the sensor modalities on the two platforms do not coincide or show large on-body displacement for which a placement-independent ARC cannot be envisioned.

Kunze et al. [34], [53] have explored approaches to elevate the processing of the ARC to abstract features. They show that features that are robust to on-body displacement can be designed using body models and by fusing multiple sensors such as an accelerometer and a gyroscope [34]. They also show that a specific sensor modality (magnetic field sensor) can be replaced by another specific modality (gyroscope) [53].

In [17], we argue that other such transformations may be feasible, such as performing activity recognition on a three-dimensional (3-D) body model [see Figure 3(a)], which can be obtained from modalities such as IMUs, fitting a body model in video sequences, or inferring limb angles using clothing-integrated elongation sensors [54].

A hybrid approach between sensor-level and feature-level sharing was further proposed by Kunze et al., who demonstrated that sensors can autonomously self-characterize their on-body placement [55] and orientation [56] using machine-learning techniques. They propose to use on-body sensor placement self-characterization as a way to select, among a number of preprogrammed ARCs, the one most suited for the detected sensor placement.

Similarly in robotics, data from different sensors can be converted into identical abstract representations. For instance, 3-D point clouds can be measured by stereovision or a laser-range finder.

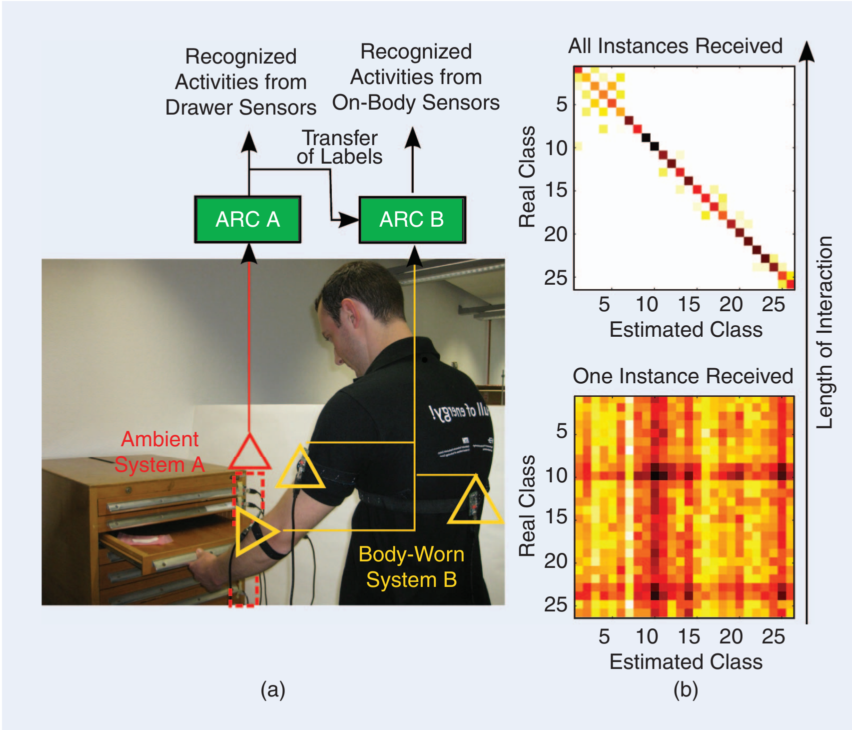

Figure 8. (a) An ambient system *A* consists of 13 drawers equipped with acceleration sensors and an ARC capable of recognizing which drawer is being opened or closed. A wearable system *B* consists of three on-body acceleration sensors. Its ARC is initially untrained. (b) The activity-recognition confusion matrix of system *B* before and after transfer learning. It indicates for each user activity (rows, here opening or closing a drawer) how the ARC classifies the activity. The closer the diagonal distribution the higher the recognition accuracy.

Classifier-Level Sharing

Transfer learning allows to translate a classification problem from one feature space to another [57] and was used to transfer perceptual categories across modalities in biological and artificial systems [58]. Conceptually, transfer learning may thus be used to translate the capability to recognize activities from one platform to another without enforcing a similar input space (i.e., sensors and features). Thus, the transfer does not affect higher-level reasoning.

Practical principles allowing a system A to confer activity-recognition capabilities to another system B are outlined in [19]. Each system *A* and *B* is composed of a set of sensors S_A, S_B, ARCs ARC_A, ARC_B, and a unified communication protocol. The process of transfer learning works as follows (see Figure 8).

- The user employs an activity-aware system *A* with ARC_A and sensor set S_A. For instance, a set of instrumented drawers is capable of reporting which one is being opened or closed in a storage-management scenario.
- A new system is deployed in the user's personal area network comprising a set of unknown new sensors S_B (on body and/or in the user's surroundings) and an untrained ARC_B. For instance, the user wears a new sensorized wristband with an integrated acceleration sensor.
- As the user performs activities, the ARC_A recognizes them and broadcasts this information.
- The new system *B* receives the class labels of the recognized activities. The ARC_B incrementally learns the mapping between the signals of the sensor set S_B and the activity classes.
- Eventually, the system *A* can be removed. The activity-recognition capability is now entirely provided by the system *B*.

The underlying assumptions are the two systems that coexist for a longer time to operate transfer learning. In Figure 8, we show that, as the user interacts with a set of drawers, the body-worn system incrementally learns to recognize opening and closing gestures.

In robotics, this sharing approach may be used to allow the robots with different sensory inputs to learn to recognize semantically identical activities or to learn how to use a new sensor when the robot parts are upgraded, thus easing programming.

Symbolic-Level Sharing

The reasoning program to infer higher level activities from spotted action primitives is shared between platforms. As the environment in which the two platforms operate may lead to the detection of semantically different action primitives, a direct transfer of the reasoning is not always possible. Carrying out a prior concept matching can address this.

For instance, to reason about the activity of a user, one needs first to know in which room he is located. One environment may have a sensor allowing to detect the action primitive "room door activated." Another environment may have a proximity infrared sensor allowing to detect "movement in the room." The interpretation of the sensor data requires different features and classifiers in each case. However, although the classifiers deliver semantically different action primitives, they may be both found to indicate the presence of a user in a room. Thus, higher-level reasoning may remain identical if these two different concepts are first matched.

Van Kasteren et al. extended transfer-learning methods to operate on time series resulting from the activation of simple binary sensors [59]. They applied this method to transfer behavior-recognition capabilities from one smart home kind to another with different and a priori unknown number and placement of sensors. The system first automatically finds how sensor activations in different environments relate to identical higher-level concepts using statistical approaches. A recognition system can also learn internal hierarchical representation of activities or concepts [60], upon which reasoning is performed. Hu et al. further report on using Web mining to match concepts [61]. Advances in merging concepts in ontologies [62], [63] support the transfer of activity-recognition reasoning across different conceptual spaces.

In robotics, these principles may allow the robots to exchange the knowledge they have individually gained about the world. This may be especially relevant when principles of autonomous mental development are used, as robots can develop distinct world representations according to their capabilities.

Other Approaches

Some approaches do not fit in the taxonomy above. Blanke and Schiele proposed a form of transfer learning to reuse action primitives across different but related application domains. Action-primitive spotting (hammering, screwing, and cutting) was trained on the data set of a shelf-assembly task. These primitives were reused as is to detect higher-level steps of a mirror-assembly task, thus considerably reducing the amount of training data needed for the new task [32].

Semisupervised learning allows to combine a limited number of labeled data with a large amount of unlabeled data to train classifiers.

Most of the approaches previously described attempt to reduce or eliminate the need for training data for activity recognition on a new platform. Beigl and coworkers proposed to "crowd-source" the acquisition of training data. They addressed the issues related to shared data labeling by developing a framework suitable for end users operating on a mobile phone [64]. Semisupervised learning allows to combine a limited number of labeled data with a large amount of unlabeled data to train classifiers. It was successfully used to train activity-recognition systems using only sparse activity labels [65]. Recent trends seek further reduction in the data-collection efforts by automatically generating activity-recognition models from online sources by data mining [66].

Calatroni et al. argue that many existing sensors can be repurposed for activity recognition, although they were initially deployed for other uses [67]. They show, for instance, how reed switches placed in windows for security purposes can be used to infer standing or walking by means of assumptions about human behavior when interacting with the instrumented object. They indicate several other sensors and behavioral assumptions that allow to obtain sporadic labels about the modes of locomotion of the user or his or her gestures. They suggest to incrementally train the body-worn recognition system whenever such labels are obtained, with the transfer-learning method described earlier. Eventually, the wearable system becomes capable of activity recognition even when the user does not interact with the source of labels. Since this process can be continuous, the system can perform activity recognition with many unforeseen combinations of on-body sensors as long as they provide discriminative signals.

Finally, in wearable computing, the user and the technical system are tightly interacting. Thus, in some cases, the user may provide information to the wearable system, such as whether it correctly identified the last activity. This can be used in an online learning paradigm to refine activity models according to the user's expectations. We showed in [68], how a binary true/false feedback can be used to guide adaptation. In the example, the feedback was provided by an electroencephalography (EEG)-based brain–computer interface, but a simple push-button feedback is also possible. In robot teams, this may be akin to robot sporadically judging the performance of their peers when performing a collective task.

Conclusions and Outlook

Activity recognition enables a WWW for robots by providing a tool to label large robot-generated activity data sets, by enabling activity-aware HRI, and by opening the way to self-learning autonomous robots capable of monitoring their own proficiency at a task. The large data sets of collective human behaviors collected in wearable computing may also be used to bootstrap humanoid robot behaviors and thus achieve more humanlike interactions in human–robot societies.

Human activity recognition has been a major object of research in wearable computing since the mid-1990s. We summarized the methods developed in the community along the ARC, which is a set of processing principles followed in most activity-recognition research. Since human activities are highly variable, we reported some of the recent advances to enhance the robustness of activity-recognition systems when they are shared among different users or deployed in different application domains. Human activity recognition from on-body sensors is far from a solved problem. Some of the continuing challenges include:

- finding more efficient sensor modalities for activity recognition. They should satisfy multiple requirements:

minimize obtrusiveness, be highly discriminative of the activities of interest, and minimize subsequent computational complexity
- spotting rare events and short activities in a large stream of data, which is still a challenging segmentation and null-class rejection problem
- despite recent advances surveyed in this article, coping with human motion variability remains an open area of research
- deploying activity recognition to new problem domains without an expensive training phase is still elusive
- shared reference activity-recognition data sets are important for benchmarking purposes. We reviewed a few activity-recognition data sets and proposed a new benchmark data set in [36]
- building and updating the state of a world model according to the user's actions. For instance, when a user displaces a cup, this changes the meaning of a "grasp" gesture performed at the prior location of the cup. Most current approaches assume stateless world models.

Other challenges relate to the use of activity-recognition systems in robotics. The annotation of large-scale data sets or the recognition of human activities for HRI must take into account that the machine recognition of activities is not perfectly accurate. Thus, probability distributions on the recognized activity classes need to be taken into account for further processing, for instance, in a Bayesian framework.

Using activity recognition in a robotic self-learning paradigm builds on the assumption that it is preferable to translate an activity-recognition system between robots rather than a motor program. Translation between robots of identical make is relatively straightforward and may allow the robots to learn new motor strategies when the actuators are damaged. Translation across heterogeneous platforms assumes a greater invariance in the activity-recognition system than in the motor program. The coming years will see whether self-learning in heterogeneous platforms driven by a common activity-recognition system can be reliably achieved.

Nevertheless, there are some important differences between human and robot activity recognition. One can design the robot hardware to place sensors at good positions. Moreover, one can adapt the robot behavior to optimize the quality of the sensor data. For instance, the speed of the robot can be reduced if the sensors are slow. The mission of the robot may be temporarily put on hold while it refines its world models or tests a hypothesis. This is a clear advantage for robotics. It is exploited in autonomous mental development and allows for a joint coevolution of the capability to use sensors and actuators and to discover knowledge about the world.

Instead, with human activity recognition, the sensors are moved by the user, and the methods must make best use of the sensor data as they come in, with little possibility for the wearable system to influence the user's behavior. Initiatives toward networked wearable systems attempt to address this by capitalizing on a large deployment of similar systems [64]. However, deploying sensors on people is not free of difficulties. User acceptance must be taken into account, and this limits the kind of sensors that can be integrated into clothing. Finally, the morphology of the platform plays a key role. Humans have roughly similar morphologies. The larger variability in robot morphologies makes the challenge of activity recognition harder. One way to address this in a WWW for robots is to ensure that robots are capable of similar "abstract capabilities." Thus, rather than constraining morphology and sets of sensors, the designer could provide a means to transform the effective robot capabilities to a common set of abstract capabilities. This is akin to the use of abstract features described here.

The activity-recognition methods must be integrated in a software architecture to fully realize a WWW for robots. We refer to [50] for a discussion of a software framework that can be used to create flexible activity-aware systems that can be deployed on heterogeneous platforms, using the methods presented here. We illustrate this on the example of activity-aware applications that are downloaded from Internet application repositories and executed on opportunistically discovered sensing and actuating resources surrounding the user.

We invite the interested readers to look for further information on activity recognition in wearable and pervasive computing in the following conference proceedings: International Symposium on Wearable Computers, International Conference on Pervasive Computing, and International Conference on Ubiquitous Computing. The following journals also cover the topic: *IEEE Pervasive Computing Magazine*, *Personal and Ubiquitous Computing* (Springer), and *Pervasive and Mobile Computing* (Elsevier).

Acknowledgments

We acknowledge the financial support of EU FP7 under the project OPPORTUNITY with grant number 225938 and RoboEarth with grant number 248942.

References

[1] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[2] J. Geusebroek, G. Burghouts, and A. Smeulders, "The Amsterdam library of object images," *Int. J. Comput. Vis.*, vol. 61, no. 1, pp. 103–112, 2005.

[3] A. Torralba, R. Fergus, and W. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 30, no. 11, pp. 1958–1970, 2008.

[4] J. Fenwick, P. Newman, and J. Leonard, "Cooperative concurrent mapping and localization," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2002, vol. 2, pp. 1810–1817.

[5] R. D'Andrea and P. Wurman, "Future challenges of coordinating hundreds of autonomous vehicles in distribution facilities," in *Proc. IEEE Int. Conf. Technologies for Practical Robot Applications*, 2008, pp. 80–83.

[6] M. Roth, D. Vail, and M. Veloso, "A real-time world model for multi-robot teams with high-latency communication," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2003, vol. 3, pp. 2494–2499.
[7] R. Madhavan, K. Fregene, and L. E. Parker, "Distributed cooperative outdoor multirobot localization and mapping," *Autonom. Robots*, vol. 17, no. 1, pp. 23–39, 2004.
[8] C. Martin, F.-F. Steege, and H.-M. Gross, "Estimation of pointing poses for visually instructing mobile robots under real world conditions," *Robot. Autonom. Syst.*, vol. 58, no. 2, pp. 174–185, 2010.
[9] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and reproduction of gestures by imitation: An approach based on hidden Markov model and Gaussian mixture regression," *IEEE Robot. Automat. Mag.*, vol. 17, no. 2, pp. 44–54, 2010.
[10] F. Bertsch and V. Hafner, "Gesture recognition as a prerequisite of imitation learning in human-humanoid experiments," in *Proc. 9th Int. Conf. Epigenetic Robotics*, 2009, pp. 197–198.
[11] V. Hafner and F. Bachmann, "Human-humanoid walking gait recognition," in *Proc. 8th IEEE-RAS Int. Conf. Humanoid Robots*, 2008, pp. 598–602.
[12] T. Duong, D. Phung, H. Bui, and S. Venkatesh, "Efficient duration and hierarchical modeling for human activity recognition," *Artif. Intell.*, vol. 173, no. 7–8, pp. 830–856, 2009.
[13] J. Tani, R. Nishimoto, J. Namikawa, and M. Ito, "Codevelopmental learning between human and humanoid robot using a dynamic neural-network model," *IEEE Trans. Syst., Man, Cybern. B*, vol. 38, no. 1, pp. 43–59, 2008.
[14] J. Sturm, C. Plagemann, and W. Burgard, "Unsupervised body scheme learning through self-perception," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2008, pp. 3328–3333.
[15] P. Varshavskaya, L. P. Kaelbling, and D. Rus, "Automated design of adaptive controllers for modular robots using reinforcement learning," *Int. J. Robot. Res.*, vol. 27, no. 3–4, pp. 505–526, 2008.
[16] I. Harvey, E. Di Paolo, R. Wood, M. Quinn, and E. Tuci, "Evolutionary robotics: A new scientific tool for studying cognition," *Artif. Life*, vol. 11, no. 1–2, pp. 79–98, 2005.
[17] D. Roggen, K. Förster, A. Calatroni, T. Holleczek, Y. Fang, G. Tröster, P. Lukowicz, G. Pirkl, D. Bannach, K. Kunze, A. Ferscha, C. Holzmann, A. Riener, R. Chavarriaga, J. del, and R. Millán, "Opportunity: Towards opportunistic activity and context recognition systems," in *Proc. 3rd IEEE WoWMoM Workshop Autononomic and Opportunistic Communications*, 2009.
[18] P. Zappi, C. Lombriser, E. Farella, D. Roggen, L. Benini, and G. Tröster, "Activity recognition from on-body sensors: Accuracy-power trade-off by dynamic sensor selection," in *Proc. 5th European Conf. Wireless Sensor Networks (EWSN 2008)*, R. Verdone, Ed. Berlin: Springer, 2008, pp. 17–33.
[19] A. Calatroni, C. Villalonga, D. Roggen, and G. Tröster, "Context cells: Towards lifelong learning in activity recognition systems," in *Proc. 4th European Conf. Smart Sensing and Context*. Berlin: Springer, 2009, pp. 121–134.
[20] N. Davies, D. P. Siewiorek, and R. Sukthankar, "Special issue: Activity-based computing," *IEEE Pervasive Comput.*, vol. 7, no. 2, pp. 20–21, 2008.
[21] P. Lukowicz, O. Amft, D. Roggen, and J. Cheng, "On-body sensing: From gesture-based input to activity-driven interaction," *IEEE Comput.*, vol. 43, no. 10, pp. 92–96, 2010.
[22] S. Mann, "Smart clothing: The shift to wearable computing," *Commun. ACM*, vol. 39, no. 8, pp. 23–34, 1996.
[23] R. Want, "When cell phones become computers," *IEEE Pervasive Comput.*, vol. 8, no. 2, pp. 2–5, 2009.
[24] S. Mann, "Humanistic computing: Wearcom as a new framework and application for intelligent signal processing," in *Proc. IEEE*, vol. 86, no. 11, pp. 2123–2151, 1998.
[25] T. Starner, J. Weaver, and A. Pentland, "Real-time American sign language recognition using desk and wearable computer based video," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 12, pp. 1371–1375, 1998.
[26] V. Stanford, "Wearable computing goes live in industry," *IEEE Pervasive Comput. Mag.*, vol. 1, no. 4, pp. 14–19, 2002.
[27] S. Kallio, J. Kela, P. Korpipää, and J. Mäntyjärvi, "User independent gesture interaction for small handheld devices," *Int. J. Pattern Recogn. Artif. Intell.*, vol. 20, no. 4, pp. 505–524, 2006.
[28] A. Pentland, "Healthwear: Medical technology becomes wearable," *Computer*, vol. 37, no. 5, pp. 42–49, 2004.
[29] E. H. Chi, J. Song, and G. Corbin, "Killer app of wearable computing: Wireless force sensing body protectors for martial arts," in *Proc. 17th Annual ACM Conf. User Interface Software and Technology*, 2004, pp. 277–285.
[30] T. Stiefmeier, D. Roggen, G. Ogris, P. Lukowicz, and G. Tröster, "Wearable activity tracking in car manufacturing," *IEEE Pervasive Comput. Mag.*, vol. 7, no. 2, pp. 42–50, 2008.
[31] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," in *Proc. Pervasive Computing*, 2006, pp. 1–16.
[32] U. Blanke and B. Schiele, "Remember and transfer what you have learned-Recognizing composite activities based on activity spotting," in *Proc. 14th IEEE Int. Symp. Wearable Computers*, 2010.
[33] J. Ward, P. Lukowicz, G. Tröster, and T. Starner, "Activity recognition of assembly tasks using body-worn microphones and accelerometers," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 28, no. 10, pp. 1553–1567, 2006.
[34] K. Kunze and P. Lukowicz, "Dealing with sensor displacement in motion-based onbody activity recognition systems," in *Proc. 10th Int. Conf. Ubiquitous Computing*, 2008, pp. 20–29.
[35] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Proc. 2nd Int. Conf. Pervasive Computing*, 2004, pp. 1–17.
[36] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, M. Creatura, and J. del R. Millán, "Collecting complex activity data sets in highly rich networked sensor environments," in *Proc. 7th Int. Conf. Networked Sensing Systems*, New York: IEEE Press, 2010, pp. 233–240.
[37] A. Tognetti, N. Carbonaro, G. Zupone, and D. De Rossi, "Characterization of a novel data glove based on textile integrated sensors," in *Proc. 28th IEEE Int. Conf. Engineering in Medicine and Biology Society*, 2006, pp. 2510–2513.
[38] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. P. Cardoso, "Preprocessing techniques for context recognition from accelerometer data," *Pervasive and Mobile Comput.*, vol. 14, no. 7, pp. 645–662, 2010.
[39] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of context modelling and reasoning techniques," *Pervasive Mobile Comput.*, vol. 6, no. 2, pp. 161–180, 2010.
[40] H. Qian, Y. Mao, W. Xiang, and Z. Wang, "Recognition of human activities using SVM multi-class classifier," *Pattern Recognit. Lett.*, vol. 31, no. 2, pp. 100–111, 2010.

[41] C. Randell and H. Muller, "Context awareness by analysing accelerometer data," in *Proc. 4th IEEE Int. Symp. Wearable Computers*, 2000, pp. 175–176.
[42] M. Ko, G. West, S. Venkatesh, and M. Kumar, "Online context recognition in multisensor systems using dynamic time warping," in *Proc. Int. Conf. Intelligent Sensors, Sensor Networks and Information Processing*, 2005, pp. 283–288.
[43] J.-Y. Yang, J.-S. Wang, and Y.-P. Chen, "Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers," *Pattern Recognit. Lett.*, vol. 29, no. 16, pp. 2213–2220, 2008.
[44] G. Bailador and G. Triviño, "Pattern recognition using temporal fuzzy automata," *Fuzzy Sets Syst.*, vol. 161, no. 1, pp. 37–55, 2010.
[45] D. Riboni and C. Bettini, "Context-aware activity recognition through a combination of ontological and statistical reasoning," in *Ubiquitous Intelligence and Computing* (Lecture Notes in Computer Science, vol. 5585), D. Zhang, M. Portmann, A.-H. Tan, and J. Indulska, Eds. Berlin: Springer-Verlag, 2009, pp. 39–53.
[46] A. Ranganathan, J. Al-Muhtadi, and R. H. Campbell, "Reasoning about uncertain contexts in pervasive computing environments," *IEEE Pervasive Comput. Mag.*, vol. 3, no. 2, pp. 62–70, 2004.
[47] S. McKeever, J. Ye, L. Coyle, and S. Dobson, "Using dempster-shafer theory of evidence for situation inference," in *Proc. 4th European Conf. Smart Sensing and Context*, (ser. EuroSSC'09). Berlin: Springer-Verlag, 2009, pp. 149–162.
[48] A. Benbasat and J. Paradiso, "An inertial measurement framework for gesture recognition and applications," in *Gesture and Sign Language in Human-Computer Interaction*, I. Wachsmuth and T. Sowa, Eds. Berlin: Springer-Verlag, 2002, pp. 77–90.
[49] H. Harms, O. Amft, and G. Tröster, "Modeling and simulation of sensor orientation errors in garments," in *Proc. 4th Int. Conf. Body Area Networks*, New York: ICST, 2009.
[50] D. Roggen, C. Lombriser, M. Rossi, and G. Tröster, "Titan: An enabling framework for activity-aware "Pervasive Apps" in opportunistic personal area networks," *EURASIP J. Wireless Commun. Networking*, vol. 2011, 2011.
[51] K. Förster, D. Roggen, and G. Tröster, "Unsupervised classifier self-calibration through repeated context occurences: Is there robustness against sensor displacement to gain?" in *Proc. 13th IEEE Int. Symp. Wearable Computers*, New York: IEEE Press, 2009, pp. 77–84.
[52] H. Bayati, J. d. R. Millán, and R. Chavarriaga, "Unsupervised adaptation to on-body sensor displacement in acceleration-based activity recognition," in *Proc. 15th Int. Symp. Wearable Computers*, New York: IEEE Press, to be published.
[53] K. Kunze, G. Bahle, P. Lukowicz, and P. Partridge, "Can magnetic field sensors replace gyroscopes in wearable sensing applications?" in *Proc. 14th Int. Symp. Wearable Computers*. New York: IEEE Press, 2010.
[54] C. Mattmann, F. Clemens, and G. Tröster, "Sensor for measuring strain in textile," *Sensors*, vol. 8, no. 6, pp. 3719–3732, 2008.
[55] K. Kunze, P. Lukowicz, H. Junker, and G. Troester, "Where am I: Recognizing on-body positions of wearable sensors," in *Proc. 1st Int. Workshop on Location- and Context-Awareness*, T. Strang and C. Linnhoff-Popien, Eds. Berlin: Springer-Verlag, 2005, pp. 257–268.
[56] K. Kunze, P. Lukowicz, K. Partridge, and B. Begole, "Which way am I facing: Inferring horizontal device orientation from an accelerometer signal," in *Proc. 13th Int. Symp. Wearable Computers*, New York: IEEE Press, 2009, pp. 149–150.
[57] W. Dai, Y. Chen, G. Xue, Q. Yang, and Y. Yu, "Translated learning: Transfer learning across different feature spaces," in *Proc. Neural Information Processing Systems (NIPS)*, 2008, pp. 353–360.
[58] F. Orabona, B. Caputo, A. Fillbrandt, and F. W.Ohl, "A theoretical framework for transfer of knowledge across modalities in artificial and biological systems," in *Proc. 8th Int. Conf. Development and Learning*, 2009, pp. 1–7.
[59] T. van Kasteren, G. Englebienne, and B. Kröse, "Transferring knowledge of activity recognition across sensor networks," in *Proc. 8th Int. Conf. Pervasive Computing*, 2010, pp. 283–300.
[60] T. Huynh, M. Fritz, and B. Schiele, "Discovery of activity patterns using topic models," in *Proc. 10th Int. Conf. Ubiquitous Computing*, New York: ACM, 2008, pp. 10–19.
[61] D. H. Hu, V. W. Zheng, and Q. Yiang, "Cross-domain activity recognition via transfer learning," *Pervasive Mobile Comput.*, to be published.
[62] D. Corbett, "Comparing and merging ontologies: A concept type hierarchy approach," in *Proc. 14th Int. Symp. Methodologies for Intelligent Systems*, 2003, pp. 75–82.
[63] N. F. Noy, "Semantic integration: A survey of ontology-based approaches," *SIGMOD Rec.*, vol. 33, no. 4, pp. 65–70, 2004.
[64] M. Berchtold, M. Budde, D. Gordon, H. Schmidtke, and M. Beigl, "Actiserv: Activity recognition service for mobile phones," in *Proc. 14th IEEE Int. Symp. Wearable Computers*. New York: IEEE Press 2010.
[65] M. Stikic, D. Larlus, and B. Schiele, "Multi-graph based semi-supervised learning for activity recognition," in *Proc. 13th IEEE Int. Symp. Wearable Computers*, 2009, pp. 85–92.
[66] M. Perkowitz, M. Philipose, K. Fishkin, and D. J. Patterson, "Mining models of human activities from the web," in *Proc. 13th Int. Conf. World Wide Web* (Ser. WWW'04), S. Feldman, M. Uretsky, M. Najork, and C. Wills, Eds. New York: ACM, 2004, pp. 573–582.
[67] A. Calatroni, D. Roggen, and G. Tröster, "A methodology to use unknown new sensors for activity recognition by leveraging sporadic interactions with primitive sensors and behavioral assumptions," in *Proc. Opportunistic Ubiquitous Systems Workshop 12th ACM Int. Conf. Ubiquitous Computing*, 2010.
[68] K. Förster, A. Biasiucci, R. Chavarriaga, J. del R. Millàn, D. Roggen, and G. Tröster, "On the use of brain decoded signals for online user adaptive gesture recognition systems," in *Proc. 8th Int. Conf. Pervasive Computing*, 2010, pp. 427–444.

Daniel Roggen, Wearable Computing Laboratory, Swiss Federal Institute of Technology, Zürich, Switzerland. E-mail: droggen@gmail.com.

Stéphane Magnenat, Autonomous Systems Laboratory, Swiss Federal Institute of Technology, Zürich, Switzerland.

Markus Waibel, Institute for Dynamic Systems and Control, Swiss Federal Institute of Technology, Zürich, Switzerland.

Gerhard Tröster, Wearable Computing Laboratory, Swiss Federal Institute of Technology, Zürich, Switzerland.