Real-Time Transfer and Evaluation of Activity Recognition Capabilities in an Opportunistic System

Marc Kurz, Gerold Hölzl, Alois Ferscha
Johannes Kepler University Linz
Institute for Pervasive Computing
Linz, Austria
{*kurz, hoelzl, ferscha*}@*pervasive.jku.at*

Alberto Calatroni, Daniel Roggen, Gerhard Tröster
ETH Zürich
Wearable Computing Laboratory
Zürich, Switzerland
{*alberto.calatroni, daniel.roggen, troester*}@*ife.ee.ethz.ch*

***Abstract*—This paper describes and evaluates the challenging feature of an opportunistic activity recognition system to train a newly discovered sensor with the available sensing devices to recognize activities at runtime. The term "opportunistic" means that the system does not operate with a fixed set of sensor devices, but uses and configures the currently available sensors that just happen to be available. Therefore, the paper presents a reference implementation of an opportunistic system, referred to as *OPPORTUNITY Framework*, and demonstrates the transfer of recognition capabilities from a fused multi-sensor ensemble to an untrained sensing device within the system in a real-world setup. Main contribution of the paper is the evaluation of the approach by describing an experimental setup and presenting results in terms of accuracy and recognition rate from the machine-learning perspective as well as from the framework and system perspective by comparing predicted classes from the teaching sensor set and the newly trained sensor to obtain QoS parameters.**

***Keywords*-Activity and Context Recognition; Opportunistic Sensing; Senor Networks.**

I. Introduction

Activity and context recognition systems utilize sensing devices that are available in the environment, on objects, or on persons to sense the world in terms of inferring activities. Traditionally, machine learning technologies that interpret the sensor datastreams are trained in an offline mode, at the design time of the system [1]. In contrast to that, an opportunistic system does not specify the set of required sensor systems a priori, instead it utilizes sensor nodes that just happen to be available to recognize the person's activities. Another major characteristic of an opportunistic system is the fact that recognition goals are defined dynamically at runtime by an application or a user, and the available sensing devices are configured to an *ensemble*, which is the set of accessible sensors that are best suited to execute this goal [2]. The type and modality of the involved sensor systems that are utilized to execute a recognition goal cannot be pre-defined as sensors are used that just happen to be available. Therefore, the system has to handle physical, logical, and other types of sensors [3] in order to execute a recognition goal. Given these definitions, some characteristic features and application cases of an opportunistic activity recognition system can be identified, like: (i) *sensor appears*, (ii) *sensor disappears*, (iii) *sensor reappears*, (iv) *sensor delivers reduced-quality data* and (v) *sensors are trained for an active recognition goal at runtime* [4].

This paper presents the feature that a newly appeared sensor (*Learner*) can be trained with the existing ones that are executing a recognition goal (*Teacher(s)*) by providing the predicted activity class to incrementally train the new sensor and calculate QoS parameters [5] on the fly to estimate to what extent the learner will be able to contribute to a future, similar recognition goal. Therefore, we use the *OPPORTUNITY Framework*, a reference implementation of an opportunistic activity recognition system (see [3] [4] [5]). By utilizing a *system-supervised learning* approach on the new sensor node [6] we compare the predicted label of the teacher to the label predicted by the learner to calculate a *degree of fulfillment (DoF)* [5] metric that indicates to what extent a sensor can fulfill a certain recognition goal. This information together with the dynamically obtained machine learning parameters (e.g., classifier model) is stored persistently in the sensor's self-description. We present and evaluate the approach by operating the OPPORTUNITY Framework in a real-world scenario with four body-mounted sensor devices that deliver triaxial accelerometer data. We transfer the capability to recognize the locomotion of a person (i.e., *WALK*, *SIT*, *STAND*, *LIE*) from a 3-sensor ensemble to a single sensor and compare the teacher with the learner output class to evaluate the approach. This is radically different from standard settings using offline training, since we cannot gather groundtruth labels here to assess the performance of the learner. We instead have to rely only on the teacher-learner comparison.

The remainder of the paper is structured as follows: Section 2 provides an overview on related work. Section 3 presents a description of the technical details and the realization of the sensor learning approach. Sections 4 and 5 describe an experimental setup with on-body sensors and the results of the training. Section 5 closes with a conclusion and an outlook.

 ISBN: 978-1-61208-156-4

II. Related Work

Traditional activity recognition systems have to define the classes that shall be recognized, the sensors and their operating characteristics at design time of the system. The activity recognition chains that process sensor signals to infer activities involve different steps: datastream preprocessing, feature extraction, classification and multi-sensor fusion. The classification step (mapping feature vectors to a defined set of output classes) involves offline-trained machine learning algorithms [1] [7] [8].

As opportunistic sensing and opportunistic activity recognition draw from the characteristic to use sensor nodes that just happen to be available to execute a dynamically-stated recognition goal (see [2] [3] [4] [5]), an approach to make sensors ready to recognize activities on the fly at runtime of the system is necessary. This transferring of recognition capabilities from one (or more) sensor(s) to another sensor can be done on the classifier level, where the model is transferred directly from one sensor node to another [6]. This approach suffers from the problem that both nodes have to operate on the same feature space, which limits the approach. Another way is to provide the feature space independent classes to the learner, which can incrementally train the baseline machine learning technologies [6]. In [9] the authors showed the transfer of activity recognition capabilities from one smart home to multiple different systems operating in the same domain. Again, the transfer relies on a common feature space. The transfer of capabilities across multiple feature spaces is also shown in [10].

This paper takes on the approach that is presented in [6] and shows that the transfer of recognition capabilities in an opportunistic environment is possible in a real-world scenario. As no ground truth is usually available, an estimate of the QoS for the learner has to be calculated. This is done in form of a DoF that is calculated by comparing the predicted class from the teacher with the predicted class from the learner and stored as part of the sensor self-description. Details of these self-descriptions are presented in [5]. This paper extends the *system-supervised learning* approach that operates independently from the feature spaces of the teachers and learners as described in [6]. There, the approach is described and evaluated on a rich dataset [11] and by using Nearest Class Center (NCC), k-Nearest Neighbors and SVM classifiers with the advantage to have a groundtruth available. Main contribution of this paper is the application of the approach in the OPPORTUNITY Framework, and the empirical calculation of the accuracy in form of the DoF during the training phase at runtime.

III. Technical Details and Realization

This Section provides technical details of the OPPORTUNITY Framework, the applied sensor self-description concept, and how the transfer learning approach is realized within the framework.

A. The OPPORTUNITY Framework

A reference implementation of an opportunistic activity recognition system (the *OPPORTUNITY Framework*) was developed and is used within this paper. The framework is written in Java/OSGi and is a step towards a ready-to-use middleware for building opportunistic activity recognition applications in different domains [3] [4] [5]. An opportunistic system utilizes sensors in a way to configure the best set of sensors according to a recognition goal. If we assume that the set is not static, then our system needs to react on changes in the sensing infrastructure (e.g., a node might disconnect when running out of power). The following more or less challenging features can be identified that characterize such an opportunistic system in terms of (self-) adaptation:

(i) *Sensor appears*: a new sensor joins the sensing infrastructure. If the sensor is already capable of contributing to the recognition goal, the system has to assess whether or not the sensor is able to increase the overall ensemble's contribution to the recognition goal. If the sensor is still untrained, it can be trained by the other sensor(s). The system gets the knowledge of the sensor's capabilities by parsing its self-description.

(ii) *Sensor disappears*: when a sensor disconnects, the system reaction depends on whether the sensor was or was not active in a configured ensemble. In case the sensor was not active, the current sensor ensemble does not have to be reconfigured. In the other case, a reconfiguration could be needed.

(iii) *Sensor reappears*: same as *sensor appears*, but the system already knows the sensor's capabilities as it has parsed its self-description on previous connections.

(iv) *Sensor delivers faulty data*: when a sensor is shifted or broken it could be that it still delivers (reduced-quality) data. The system has to recognize this and reduce the trust indicator metric of the sensor [5] and - if necessary - reconfigure the sensing ensemble.

(v) *Transfer of recognition capabilities to a sensor*: this is the main focus of this paper. A newly connected sensor has to be trained by the other sensors in terms of recognizing activities and an estimation needs to be provided for the achieved accuracy.

(vi) *Ensemble configuration at runtime*: this is a key aspect in an opportunistic system. Whenever a recognition goal is stated to the system, the set of sensors that are best suited to execute this goal are configured. This process [5] has to be executed whenever something happens in the sensing infrastructure.

The next Section III-B explains the concept of sensor self-description and how this can be used to perform real-time transfer learning, and III-C describes how transfer learning is implemented in the OPPORTUNITY Framework and how its performance can be measured in an online setting.

 ISBN: 978-1-61208-156-4

B. Sensor Self-Description

The sensor self-description is an important aspect in an opportunistic activity and context recognition system. It provides the technical details of the sensor to the system and builds the connection point between the high-level framework features and the machine learning technologies on the lower levels. We have split the self-description into two parts: (i) the technical description that holds the physical characteristics as well as technical aspects of a sensor (e.g., power requirements, communication interface, update rate, size, weight, ...), and (ii) the dynamic description that lists the sensor's capabilities according to recognition goals [5]. Both parts of the sensor self-description follow the OpenGIS SensorML specification, an XML standard definition. The concept of *ExperienceItems* defines a complete recognition chain of a sensor together with the required signal processing and machine learning techniques (feature extraction, classification, fusion) to recognize activity classes. ExperienceItems are part of the dynamic self-description and there can be multiple items describing multiple activity classes or recognition chains [5]. These ExperienceItems and the defined methodologies can be invoked and configured by the activity recognition system at runtime on demand. The DoF defines to what extent a sensor together with the recognition chain can execute a recognition goal. Figure 1 shows a clipping of an ExperienceItem where the DoF is defined for the activity class *WALK*. This means, that the sensor and the corresponding machine learning techniques as defined in this very ExperienceItem can execute the recognition goal *WALK* with a DoF of 0.75. This degree of fulfillment metric reflects the expected accuracy in the recognition of a certain activity and can be generally calculated in two ways:

(i) *Statically by using a labeled groundtruth for learning*: normally a system is trained by using a groundtruth that defines the sensor signal and patterns and the corresponding activity classes. The classification mechanisms are therefore trained with the labeled groundtruth data to autonomously detect significant and similar patterns in the datastream. This approach presumes the initial knowledge of the used sensors, their modalities, the feature space, the exact position and location and the activity classes that shall be recognized (and of course a groundtruth of appropriate size/length).

(ii) *Dynamically by using transfer learning*: this method is the core contribution of the paper and will be described in detail in the following Section III-C.

To avoid the need for a labeled groundtruth and thus the training in offline mode, this paper presents and evaluates the approach of training new sensor nodes at runtime of the system and calculating the DoF by comparing the predicted class from the teacher with the predicted class from the learner. The next Section describes the approach of real-time transfer learning and the calculation of the DoF in detail.

```
...
  <swe:field name="labellist">
   <swe:DataRecord>
    <swe:field name="WALK">
     <swe:Text definition="dof">
      <swe:value>0.75</swe:value>
     </swe:Text>
    </swe:field>
   </swe:DataRecord>
  </swe:field>
...
```

Figure 1. A clipping of an ExperienceItem as part of a sensor self-description together with the DoF metric [5].

C. Application of Transfer Learning

When a new sensor appears, the system parses its self-description to check whether it can contribute to a running execution of a recognition goal, or if not, whether it can be regarded as a learner candidate for one of the goals that are currently active. A learner candidate has to be defined so by a human expert. For example, one could define an accelerometer on the shoe as possible sensor to detect the modes of locomotion. Learner candidates have in their self-description a feature extraction and a classification method, which form a default ExperienceItem template. Their initial DoF is set manually to ”0.0”. Thus, the system recognizes a learner candidate for a recognition goal when a sensor appears in the sensing environment with an initial zero DoF value (for this very goal). This means that the sensor can be picked for learning, and as soon as an ensemble that executes the recognition goal is configured and active, the learning process is initiated. The predicted classes from the teacher are used by the learner to generate the classifier model (persistently stored in form of a JSON file, see Figure 2 for an example) on the fly by assigning the activity class to the extracted features from the datastream.

The calculation of the effective learner DoF is not a trivial task, since groundtruth information is not available at runtime. For the calculation of the DoF, we can only rely on comparisons between teacher and learner, taking into account that the teacher does not provide a perfect groundtruth. We calculate the agreement rate using the cumulative moving average, which is a statistical method to analyze time series data [12]. In detail, the following values and variables are used to calculate the DoF of the learner at runtime:

- DoF_T = DoF from the teacher (known from its ExperienceItem).
- DoF_L = DoF from the learner that has to be calculated on the fly.
- n = count of predicted and compared activity classes.
- ϑ = degree of teacher-learner agreement during the training phase, defined as the percentage of instances where teacher and learner output the same activity class label among the total number of examined instances.
- $[0|1]$ = false and true, indicates the match (true) or

mismatch (false) between the class labels predicted by Teacher and Learner.

DoF_L is calculated as follows as soon as the learning process ends:

$$DoF_L = DoF_T * \vartheta, \tag{1}$$

which means that as soon as the learning process and the comparison of the classes predicted by the learner and teacher are over, the DoF of the learner is calculated by multiplying the DoF of the teacher with the cumulative average of the agreement rate. This multiplication has the goal to rescale the calculated DoF in cases where we have a high agreement between teacher and learner (ϑ close to 1) and an imperfect teacher. In these cases, in fact, we have high agreement between a learner and a wrong teacher, which does not mean at all that the learner is fulfilling the recognition goal properly. As a side effect, the estimated DoF_L will always be smaller or equal to the DoF_T, which is indeed often the case, but the opposite can also happen, as can be seen in a few cases in [6]. By the above calculation we are then accepting the condition $DoF_L \leq DoF_T$ and we are making a pessimistic estimate, which leaves us on the safe side when using the learner in following missions (the learner could perform even better than foreseen by the system).

The end of the learning process in our version of the framework is reached when either the teacher (or sensors within the teacher ensemble) or the learner is disconnected. Alternative approaches based on a time lapse or upon convergence of the DoF will be investigated in future work. The cumulative moving average of the agreement rate is calculated at runtime as follows:

$$\vartheta_{n+1} = \frac{\vartheta_n * n + [0|1]}{n+1}, \tag{2}$$

where n is incremented by one after each iteration. The number that has to be added to the numerator, after having multiplied ϑ with n, depends whether the class predicted by the teacher agrees or disagrees with the one predicted by the learner. In this way, the cumulative moving average of agreeing labels can be calculated over time, building the running accuracy for the learner measured with respect to the teacher. When the training phase is finished, this relative accuracy (ϑ) is multiplied by the DoF of the teacher to get an estimation for the DoF. This value and the newly built classifier models (in form of a JSON file) are written back to the sensor's self-description, where they complete the usual information about the used feature extraction and classifier.

IV. Experiment and Evaluation

We have set up a real-time scenario with body-worn sensors to test and evaluate the transfer of recognition capabilities for the modes of locomotion (*WALK, SIT, STAND, LIE*). We have picked a rather easy activity set that has to be recognized as the goal is not to work with sophisticated and highly complex activity classes, but to test to what extent rather simple and easy to recognize activities are transferrable. We used 4 sensors that delivered triaxial acceleration data. Two Intersense InertiaCube3 sensors were mounted on the right upper-/lower-arm (*motionjacket_RUA* and *motionjacket_RLA*, see Figure 3-1 and 3-2), one bluetooth accelerometer was mounted on the right knee of the subject (*btaccel_RKN*, see Figure 3-1 and 3-3), and one SunSPOT accelerometer sensor was attached to the right shoe (*sunspot_shoetoebox*, see Figure 3-1 and 3-3). The ensemble that was trained and configured to recognize the mode of locomotion activity classes consisted of the *motionjacket_RUA*, the *sunspot_shoetoebox*, and the *btaccel_RKN* sensors. The untrained sensor that was trained with the predicted classes from the teacher ensemble in our test setup was the *motionjacket_RLA* sensor. The setup of the three sensors in the ensemble was defined in an ExperienceItem:

- *Feature Extraction*: Mean/Variance for each sensor/recognition chain.
- *Classification*: NCC Classifier for each sensor/recognition chain.
- *Fusion*: Majority-Voting Fusion to combine the classification results for each sensor/recognition chain.
- *DoF*: 0.792 for *WALK, STAND, SIT, LIE* for the complete ensemble.

Below the picture of the test subject in Figure 3, also some sensor datastreams are shown. The *motionjacket_RLA* sensor was picked as learner as we set its DoF to detect modes of locomotion to zero. The predicted class from the ensemble was not only presented as system result, but also to the learner to incrementally train the classifier (NCC) model. The following Section V summarizes the results that were achieved within the experiment.

V. Results

The learning phase in the experiment was 15 minutes long. So the teacher-ensemble presented its predicted label to


```
{
  "centroids" : [
    [-8.4965,-2.2945,2.5368,0.29659],
    [-9.2508,-2.3725,1.7388,1.3654],
    [-9.3933,-1.9461,1.0396,0.13042],
    [-4.4928,7.9204,-2.242,0.13042]
  ],
  "centroid_labels" : [3,1,5,4],
  "number_of_instances" : [1310,433,476,95],
  "cloud_size" : [
    [1.0241,0.89795,1.8205,0.2136],
    [0.32604,0.56007,0.71504,0.55195],
    [0.26634,0.57792,1.1387,0.11843],
    [1.1746,0.56141,1.5778,0.2331]
  ]
}
```


Figure 2. An example of a JSON file, which provides the configuration of a classifier used for activity recognition (the classifier model) [5].

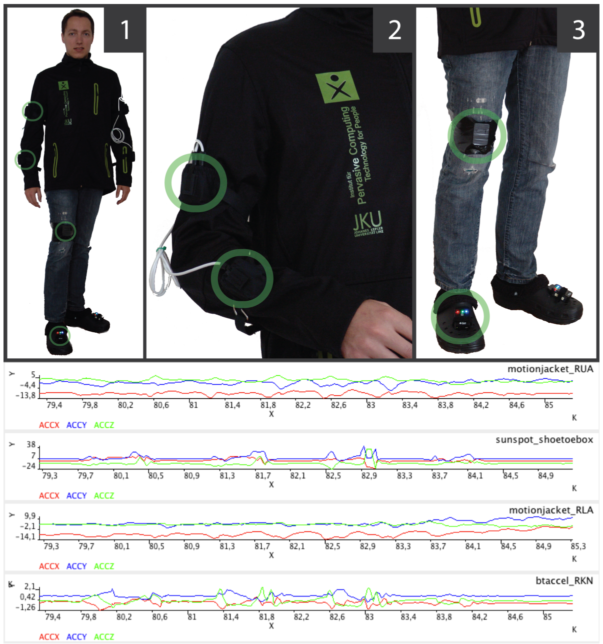

Figure 3. Experimental setup of the on-body sensors together with snapshots of the corresponding datastreams.

the learner for executing the incremental training for 15 minutes, whereas the four activity classes (*WALK, STAND, SIT, LIE*) were executed by the subject approximately equally long but in a random way. That means the subject did not execute *WALK* for 3.75 min, followed by *STAND* sequentially. The activities occurred randomly but (approx.) equally long to ensure enough training samples per activity class. The ϑ value (which is the accuracy of the learner compared to the teacher during the training phase) over time during the real-time training process is shown in Figure 4. During the first few seconds, the value varied substantially (between 1 and 0.2), which can be explained by considering that the value has to settle over time until (i) the learner classifier model is well enough trained, and (ii) enough predicted classes are compared. Therefore, it is important to have a training phase that is long enough to have a good classifier model and a stable DoF calculation for the learner. In our experiment, the ϑ value settled to 0.65. This value cannot be seen as permanently stable and fixed, since it highly depends on (i) the number of observed activity classes, and (ii) the duration of the training phase. By multiplying the DoF_T value with ϑ we get a DoF for the learner of 0.515.

In Figure 5 the confusion matrix is shown that compares the predicted activity classes from the teacher with the predicted classes from the learner. There, the calculated DoF value from the learner is reflected, as we can notice a high occurrence of the false interpretation of the *WALK* and *STAND*/*SIT* activity classes. Altogether, during the 15 minutes training phase, we had approximately 350.000 comparisons from the teacher with the learner predicted class, which are contained in the confusion matrix. Table I summarizes the results that were achieved in the experiment.

Our approach of real-time training to realize self-adaption by online training new sensors and the persistent storage of this newly acquired knowledge at runtime works. The calculation of 0.515 as DoF is realistic and reflects the self-adaptation according to real-time transfer of recognition capabilities as not only correct predicted labels are transferred from the teacher to the learner but also wrongly classified activities.

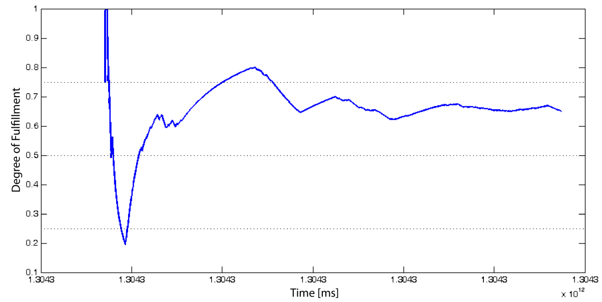

Figure 4. Evolution of ϑ of the *motionjacket_RLA* sensor during the training phase.

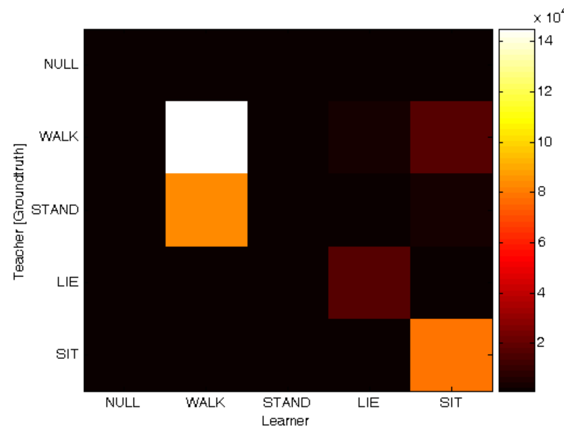

Figure 5. Visualization of the confusion matrix from the learner compared to the teacher (as groundtruth estimation).

Table I
SUMMARIZATION OF RESULTS

Duration of Training	15 minutes
Number of activity classes	4 (*WALK*, *STAND*, *SIT*, *LIE*)
DoF_T	0.792
Number of comparisons	approx. 350.000
ϑ (i.e., cumulative moving average)	0.65
DoF_L	$0.792 * 0.65 = 0.515$

VI. Conclusion and Future Work

We have presented the capability of an opportunistic activity and context recognition system to self-adapt in a way that untrained and newly appeared sensors can be trained by the existing ones. The recognition capabilities of the configured (teaching) ensemble are transferred in real-time to the learner candidates. During the learning process, the quality of the transferred recognition capabilities is quantified to get a stable and reasonable measurement of how good the learner can recognize the activity class. After DoF_L is estimated, this value is stored with the corresponding classifier model in the ExperienceItem of the sensor, to ensure persistency of the newly acquired knowledge. Our approach enables the autonomous transfer of recognition capabilities without relying on a labeled ground-truth to newly appeared sensors and therefore self-adapt to open-ended environments where sensors presumably not known at design time can be trained and used afterwards to extend the system.

There are many interesting aspects for future research. On one side, a probabilistic framework can be set up in order to provide a more accurate estimation of the learner accuracy given the rate of agreement between teacher and learner and by introducing all the possible knowledge about the teacher (like the confusion matrix). Different estimation techniques can be evaluated on existing datasets, where the groundtruth is not made available to the algorithms, but is used to assess how well the DoF can be estimated. Another interesting aspect is to investigate how to overcome the need for an expert to manually define learner candidates. This can be tackled by evaluating the suitability of a certain sensor due to other elements of its self-description, like the measured quantity (e.g., acceleration), and the placement. The end goal will be to design a planner, which will be able to select learner candidates, operate the transfer of capabilities only where it is meaningful and finally assess as precisely as possible what the learner achieved accuracy is.

Acknowledgment

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.

References

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

[2] D. Roggen, K. Förster, A. Calatroni, T. Holleczek, Y. Fang, G. Troester, P. Lukowicz, G. Pirkl, D. Bannach, K. Kunze, A. Ferscha, C. Holzmann, A. Riener, R. Chavarriaga, and J. del R. Millán, "Opportunity: Towards opportunistic activity and context recognition systems," in *Proceedings of the 3rd IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC 2009)*. Kos, Greece: IEEE CS Press, June 2009.

[3] M. Kurz and A. Ferscha, "Sensor abstractions for opportunistic activity and context recognition systems," in *5th European Conference on Smart Sensing and Context (EuroSSC 2010), November 14-16, Passau Germany*. Berlin-Heidelberg: Springer LNCS, November 2010, pp. 135–149.

[4] M. Kurz, G. Hölzl, A. Ferscha, A. Calatroni, D. Roggen, G. Tröster, H. Sagha, R. Chavarriaga, J. del R. Millán, D. Bannach, K. Kunze, and P. Lukowicz, "The opportunity framework and data processing ecosystem for opportunistic activity and context recognition," *International Journal of Sensors, Wireless Communications and Control, Special Issue on Autonomic and Opportunistic Communications*, vol. 1, June 2011.

[5] M. Kurz, G. Hölzl, A. Ferscha, H. Sagha, J. del R. Millán, and R. Chavarriaga, "Dynamic quantification of activity recognition capabilities in opportunistic systems," in *Fourth Conference on Context Awareness for Proactive Systems: CAPS2011, 15-16 May 2011, Budapest, Hungary*, May 2011.

[6] A. Calatroni, D. Roggen, and G. Tröster, "Automatic transfer of activity recognition capabilities between body-worn motion sensors: Training newcomers to recognize locomotion," in *Eighth International Conference on Networked Sensing Systems (INSS'11)*, Penghu, Taiwan, Jun. 2011.

[7] J. A. Ward, P. Lukowicz, G. Tröster, and T. E. Starner, "Activity recognition of assembly tasks using body-worn microphones and accelerometers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1553–1567, 2006.

[8] A. K. Dey and G. D. Abowd, "The context toolkit: Aiding the development of context-enabled applications." ACM Press, 1999, pp. 434–441.

[9] T. van Kasteren, G. Englebienne, and B. Kröse, "Transferring knowledge of activity recognition across sensor networks," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, P. Floréen, A. Krüger, and M. Spasojevic, Eds. Springer Berlin / Heidelberg, 2010, vol. 6030, pp. 283–300.

[10] W. Dai, Y. Chen, G.-R. Xue, Q. Yang, and Y. Yu, "Translated learning: Transfer learning across translated learning: Transfer learning across different feature spaces," in *Conference on Neural Information Processing Systems 2008 (NIPS'08)*, December 2008, pp. 353–360.

[11] 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 *Proceedings of the Seventh International Conference on Networked Sensing Systems (INSS), Kassel, Germany*. IEEE Computer Society Press, June 2010.

[12] Y.-l. Chou, *Statistical analysis, with business and economic applications*. Holt, Rinehart and Winston (New York), 1975, vol. 2nd Edition, no. 0030894220.