Activity Recognition Using Non-intrusive Appliance Load Monitoring

Olaf Wilken, Oliver Kramer, Enno-Edzard Steen and Andreas Hein
University of Oldenburg, Oldenburg, Germany

Keywords: Activity Recognition, Non-intruisve Appliance Load Monitoring.

Abstract: The recognition of sequences via non-intrusive appliance load monitoring has an important part to play for various applications in healthcare. In our work, we present a system for the detection of daily activities based on the use of appliances. The objective of our activity monitoring system is to maximize the time elder people can stay in their own domestic environment. We propose a system that is able to detect comparably complex activities that may be interrupted by other activities. In the experimental part of our work, a one-month and a half-year field study demonstrate the capabilities of the proposed approach.

1 INTRODUCTION

In the context of the demographic change, new technologies become more and more important to preserve the independence of elder people. Here, the focus lies on the recognition of activities of daily living (ADL, e.g. toileting) (Katz et al., 1963) and instrumental activities of daily living (IADL, e.g. cooking) (Katz, 1983) and the detection of deviations from these usual activities. When deviations are detected, the alert states can be used to inform assistants like relatives or nurses. The main objective of such a system is to let elder people live in their domestic environments independently as long as possible.

Various approaches for activity recognition with different types of sensors are known in literature: body-worn sensors as RFID reader (Philipose et al., 2004), ambient intrusive sensors as vision sensors (Nguyen et al., 2005; Oliver et al., 2002) or microphones (Chen et al., 2005) or ambient non-intruisve sensors as motion sensors (Virone et al., 2008; Barger et al., 2005; Guralnik and Haigh, 2002), state sensors (Kasteren et al., 2008) or power sensors (Noury et al., 2011). The classification of the sensor types in the above categories is given in (Ni Scanaill et al., 2006). Furthermore all systems, which are not based on vision sensors are called sensor-based activity recognition systems. A detailed overview of sensor-based activity recognition systems is given in (Chen et al., 2012). An overview of vision-based systems is given in (Poppe, 2010; Moeslund et al., 2006). Most activity recognition systems try to infer predefined activities with the help of probabilities. Therefore, unknown individual activities can be filtered out by predefined activities. This can lead to a loss of information. Furthermore, a lot of systems were evaluated by simplified scenarios with single activities, but in the real world the activities are complex (parallel/interrupting activities) (Chen et al., 2012). The system we propose in this work detects daily individual activities without inference of any predefined activities and the evaluation was executed in two field studies. The activity recognition based on power sensors installed in a fuse box that first classifies appliances by decomposition of total load. These systems are called *non-intrusive appliance load monitoring* (NIALM) cf. (Hart, 1992). Our algorithm is able to detect possible activities without specified presettings. In most cases, the labels of detected activities can be inferred with the help of the associated appliances. Furthermore, the developed system is able to handle noisy data, e.g., wrong classified appliances or little variations in the sequences of the same activities, and is able to detect complex activities, which can be interrupted by other activities and can also recognize parallel/interrupting activities. The main components of the system are NIALM and activity recognition. In contrast to the system we propose in this work, the approach by Noury etal (Noury et al., 2011) requires the manual specification of activities with the association of appliances for each installation and one sensor for each appliance to be classified is used (called *intrusive appliance load monitoring* (IALM) cf. (Hart, 1992)).

This work is structured as follows. The NIALM procedure of our system is introduced in Section 2 de-

Wilken O., Kramer O., Steen E. and Hein A..
Activity Recognition Using Non-intrusive Appliance Load Monitoring.
DOI: 10.5220/0004700300400048
In *Proceedings of the 4th International Conference on Pervasive and Embedded Computing and Communication Systems* (PECCS-2014), pages 40-48
ISBN: 978-989-758-000-0

scribing the sensors, the edge detection and the appliance recognition procedure. Section 3 describes our approach to detect activities as sequences of switching events. In Section 4, the behavior of our system is shown in two field studies. The article closes with a summary of the most important results in Section 5.

2 NON-INTRUSIVE APPLIANCE LOAD MONITORING

2.1 Sensor

From the employed power sensor (CRD5110 from CR Magnetics (Magnetics, 2013)), the electrical parameters voltage (V_{RMS}), current (I_{RMS}) and real power (P) are streamed with a sampling rate of 5 Hz. In Figure 1, the typical signal of an active appliance is shown. The signal has a transitive noise (transient) at the turn-on phase. Afterwards, a plateau (steady state) follows with only small variations in contrast to the transient. The turn-off phase is only an edge.

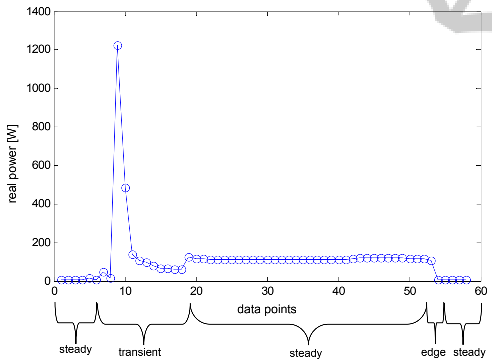


Figure 1: A signal of a running appliance from power sensor.

2.2 Edge Detection

The edge detection module recognizes switchings of appliances with real power $P(k)$ above a threshold θ_2 in the steady state. This threshold separates events generated by appliances from noise. In order to determine the real power in the steady state, two different thresholds θ_1 and θ_2 and two time slots of different lengths (σ_1 and σ_2) are employed. The first threshold θ_1 determines the beginning of a possible switching. For the determination of θ_2, two time slots are used, one for the turn-on phase and the other for the turn-off phase. The turn-on time slot is longer, because the corresponding transient noise takes longer time. For example, the noises (signal one, two and five) in Figure 2 are filtered out, and the correct switches from the third signal will be detected by this procedure. The detection of turn-on and turn-off is computed with equation 1, where “1” represents turn-on and “-1” represents turn-off. For a robust detection of turn-offs, two time slots with the same size are used. In order to eliminate noise in the stable state, the median is computed over a short period of one second.

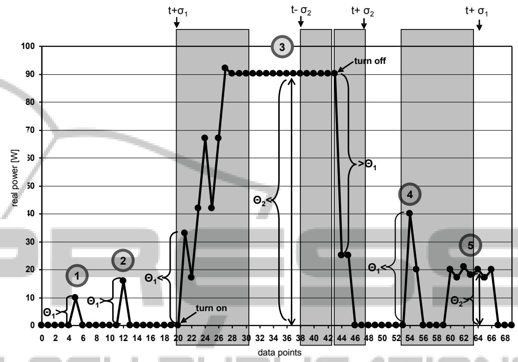


Figure 2: Example data for edge detection from turn-on/off event of an appliance ($\theta_1 = 25\,W$, $\theta_2 = 45\,W$,$\sigma_1 = 10$ datapoints, $\sigma_2 = 5$ datapoints,). The length of the turn-on phase is σ_1 and the length of turn-off is σ_2, respectively.

$$f(k) = \begin{cases} 1 & P(k+1) - P(k) \geq \theta_1 \wedge \\ & P(k+\sigma_1) - P(k) \geq \theta_2 \\ -1 & P(k+1) - P(k) \leq -\theta_1 \wedge \\ & P(k-\sigma_2) - P(k+\sigma_2) \geq \theta_2 \\ 0 & \text{else (no switch)} \end{cases} \qquad (1)$$

For each data point step with a switch event, the device recognition module is run.

2.3 Feature Extraction

The device classification is described in the following. The question arises, which features are appropriate for the device recognition task. The streamed electrical parameter voltage V_{RMS} is not stable enough: if appliances are running in parallel, the voltage varies (voltages variations have also been detected by Hart (Hart, 1992)). After each switching of an appliance, the voltage changes a little bit. This has negative implications for the recognition of the second appliance. As the power $P(k)$ depends on the voltage,

$$P(k) = V_{\mathrm{RMS}}(k) \cdot I_{\mathrm{RMS}}(k) \cdot \cos\alpha \qquad (2)$$

with phase shift α, it is no appropriate stable feature. In contrast, the voltage-independent effective resistance $R(k)$ is a suitable feature for the appliance recognition problem. It can simply be computed from the sensor data with

$$R(k) = \frac{V_{\text{RMS}}(k)}{I_{\text{RMS}}(k) \cdot \cos\alpha} = \frac{V^2_{\text{RMS}}(k)}{P(k)}. \quad (3)$$

This equation is also employed for the values of the turn-on/turn-off signal from a running appliance. However, if two or more appliances are running simultaneously, the computation of effective resistance values R_A has to consider that the appliances are connected in parallel. Therefore, the following equation has to be applied:

$$R_A(k) = \frac{1}{\frac{1}{R(k)} - \frac{1}{R(k_0-2)}}; k \in [k_0+2, k_0+n] \quad (4)$$

$$R_A(k) = \frac{1}{\frac{1}{R(k)} - \frac{1}{R(k_1+2)}}; k \in [k_1, k_1-n] \quad (5)$$

with data point k_0 for the beginning of turn-on, data point k_1 for the beginning of turn-off and $n \in \mathbb{N}_0$. In order to distinguish appliances with similar effective resistance (and similar real power), the resistance values after the turn-on are not sufficient. Since the sensor (cf. Figure 1) provides a transient signal when an appliance is turned on, it can be used for a better distinction. During the analysis of the transient signals from our appliances, we observed that the first two measurements after different turn-ons of the same appliance can be disturbed and vary too much. Therefore, they are unsuitable for a robust recognition and left out in the classification process. Mean value and standard deviation of the turn-on phase are employed for classification of similar appliances. Turn-off events do not have transient signals. The median of the last measured values before turn-off is used in combination with the information, which device is running, i.e., the turn-on information. In contrast to feature extraction real power for edge detection is sufficient because the impact of variations is here not so important.

For the classification process, K-nearest-neighbors (Cover and Hart, 1967), naïve Bayes (Mitchell, 1997) and decision trees (C4.5) (Quinlan, 1993) have been used. A comparison of the employed classifiers will be shown in the experimental section.

3 ACTIVITY RECOGNITION

The activity recognition is based on sequence recognition. A sequence is an order of letters which represent classified appliance switchings. Each appliance switching AS_i is coded by a letter in the sequence. An uppercase letter represents a turn-on event, and the corresponding lowercase represents a turn-off. For example, the coded letters for the event "toaster on" is T and for the event "toaster off" is t. The information that an appliance switching pair (turn-on/turn-off) can directly be mapped to an activity is considered as simple and robust case (e.g. turn-on/turn-off of "TV" are associated to activity "watching television"). Activities can be divided into two types:

1. A *simple activity* is represented by an interlaced sequence containing complete pairs of appliance turn-on and corresponding turn-off events that belong together. For example, in Figure 3 S_2 is a *simple activity* (closed sequence).
2. A *complex activity* contains at least two non-closed sequences, with a *begin sequence* and an *end sequence*. The *begin sequence* contains at least one turn-on event, but no corresponding turn-off. The *end sequence* contains all missing turn-off events. Figure 3 shows an example. Sequence S_1 is a *begin sequence*, S_3 is a intermediate sequence, and S_4 is an *end sequence* of a complex activity. S_2 is a parallel/interrupting activity.

The steps of the activity recognition process are described in the following.

3.1 Detection of Sequences

Our observations show that a sequence with a large number of switching events corresponds to an activity. Hence, switching events that occur in quick succession potentially belong to the same sequence. A switching event AS_j is assigned to a sequence S_i employing the following criterion:

$$\{AS_j \in S_i \mid time(AS_j) - time(AS_{j-1}) < T \wedge AS_{j-1} \in S_i\} \quad (6)$$

with threshold T being the maximum time between two switchings that belong to the same sequence. After this time-based clustering, each sequence is classified as closed or not. Again, Figure 3 illustrates the creation of sequences.

3.2 Detection of Related Sequences

In this section, the *begin sequences* BS_i, the *end sequence* ES_i and *intermediate sequences* IS_i of the complex activity candidates CAC_i are determined. The result of the created structure from the complex activity candidates is

$$\text{CAC}_i = \begin{cases} BS_i^j \mid BS^j \in \{S_1, \dots, S_n\} \\ ES_i^k \mid ES^k \in \{S_1, \dots, S_n\} \\ \{IS_i^p, \dots, IS_i^s\} \mid \{IS^p, \dots, IS^s\} \subset \{S_1, \dots, S_n\}. \end{cases} \quad (7)$$

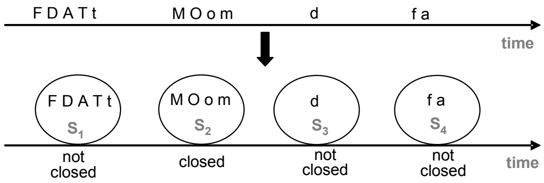


Figure 3: Example data for creation of sequences. F = "ceiling light (living room)" on; D = "TV (living room)" on; A = "table lamp (living room)" on; T = "standard lamp (living room)" on; M = "ceiling light (floor)" on; O = "ceiling light (bathroom)" on. The lowercase letters represent turn-off events. Sequences S_1, S_3 and S_4 represent the complex activity "watching TV & reading" and S_2 represents the simple activity "toileting".

Every closed sequence is a candidate for a *simple activity* EAC_i. In Figure 4, an example of two complex activity candidates CAC_1, CAC_2 and one simple activity candidate EAC_1 is shown. In this example the deteremined candidates are the ground truth, but it may occur in real situations that the *complex activity* may contain the two complex activity candidates CAC_1, CAC_2 (transitive dependency), and the sequence S_2 can be an intermediate sequence of a complex activity. These dependencies are solved in the following steps.

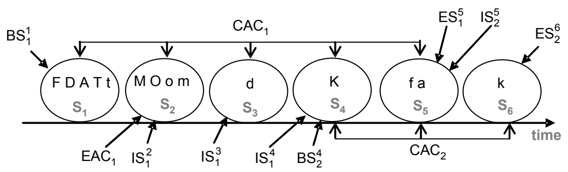


Figure 4: Example data for detected *start sequences*, *intermediate sequences* and *end sequences* of two complex activity candidates. The letters represent the same appliance switchings as in Figure 3 with additonal K = "desk lamp" on and complex activity "desk work" consisting of S_4 and S_6.

3.3 Clustering of Sequences

In this section, all sequences are clustered based on the feature similarity considering variations of the same sequences. For example, the variations can be wrong classified appliances, permutations of switchings or missing events. For the detection of such variations, the distance of two sequences is computed by an extended *edit distance* (Oommen, 1997). The *edit distance* is defined as the transformation from one sequence to another sequence with a minimum number of operations (Levenshtein, 1966). The operations are *deletion*, *insertion*, and *replacement*. With the extended *edit distance* the additional operation *transpositions* allows the detection of adjacent transpositions. Furthermore, the *extended edit distance* supports substitution matrices to recognize wrong classified appliances, and the weights of all operations can be chosen freely. It is computed employing dynamic programming techniques. The two sequences S_1 and S_2, which have to be compared, constitute columns and rows of a matrix $\mathbf{D}$ defined as

$$D_{i,j} = min \begin{cases} D_{i-1,j} + W_d & \text{(deletion)} \\ D_{i,j-1} + W_i & \text{(insertion)} \\ D_{i-1,j-1} + s(S_1(i), S_2(j)) & S_1(i) \neq S_2(j) \\ D_{i-1,j-1} & S_1(i) = S_2(j) \\ D_{i-2,j-2} & S_1(i) = S_2(j-1) \\ & \quad \wedge \\ & S_1(i-1) = S_2(j) \end{cases} \tag{8}$$

with $n = |S_1|$ and $m = |S_2|$. Function $s(\cdot)$ computes the weights of replacements (e.g. from the substitution matrix), and W_d, W_i are weights of the operations *deletion* and *insertion*. The matrix is initialized with $D_{0,0} = 0, D_{i,0} = i, 1 \leq i \leq m$ and $D_{0,j} = j, 1 \leq j \leq n$. The entry $D_{m,n}$ defines the distance computation.
The agglomerative hierarchical clustering method is used for the clustering process. The distance measure that we employ computes the relation of similarity between two sequences with the help of the extended *edit distance*

$$\delta = 1 - \frac{D_{m,n}}{max(|\ S_1\ |, |\ S_2\ |) * max(W_d, W_i, W_s)} \tag{9}$$

with weights W_s of replacement. The results of the clustering step are similarity activity clusters $ACl_i = \{S_j \dots S_n\}$. For example when complex activity from Figure 3 appears on different days then all three different sequences are in three different clusters.

3.4 Detection of Activity Candidate Clusters

Not all sequences in a similarity activity cluster ACl_i have to be associated with the same activity. For example, a start sequence of a complex activity candidate CAC_1 may be an element of ACl_i, and the associated end sequence may be element of ACl_j. Another start sequence of CAC_3 may also be element of ACl_i, but the associated end sequence is element of ACl_k. Hence, the two complex activity candidates are associated with different activities (cf. Figure 5). The correct association activity is solved by computing activity candidate cluster CAC_i^{Cl} with

$$|\ \{CAC_i \mid BS_i^m \in ACl_j, ES_i^q \in ACl_l\}\ | \geq H. \tag{10}$$

The threshold H is the minimum number of occurrence of a daily activity. Furthermore, it can happen that a similar activity cluster ACl_i may contain either closed sequences that are associated to a *simple activity* candidate or contain non-closed sequences that can be associated with a complex activity candidate. These clusters are *simple activity* candidate clusters EAC_i^{Cl}, if all closed sequences exceed the threshold S_E and all sequences, except the sequences which are associated to a complex activity candidate, on different days are larger than threshold H. The result of the correct solved association is shown in Table 1. In the next step, intermediate sequences of complex activity candidate clusters are determined.

Table 1: Assignments of similarity cluster and activity candidates.

Activity candidate cluster	Similitary activity	Activity candidate
CAC_p^{Cl}	ACl_i, ACl_j	$\{CAC_k, \ldots, CAC_m\}$
CAC_q^{Cl}	ACl_i, ACl_r	$\{CAC_f, \ldots, CAC_n\}$
...	...	...
EAC_t^{Cl}	ACl_j	$\{EAC_w, \ldots, EAC_z\}$

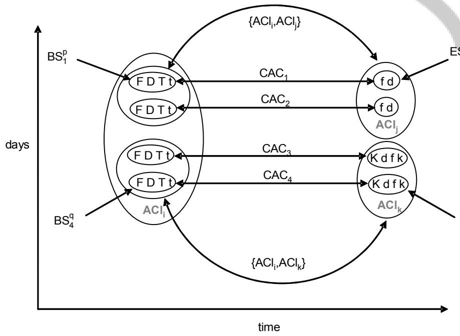


Figure 5: Example data for correct association of similarity activity clusters ($CAC_1, \ldots, CAC_4$) with two different complex activity candidate clusters. The letters represent the same appliance swichtings as in Figure 4. CAC_1 and CAC_2 are associated to activity (candidate cluster) "watching TV at night" and CAC_3 and CAC_4 are associated to "watching TV & desk work at afternoon".

3.5 Detection of Activities

In the last step, the intermediate sequences, which can also be activity candidate clusters, are determined. All the activity candidate clusters, which either are not associated to another activity candidate cluster or in which all the dependencies, the intermediate sequences, are solved, are the result activities A_i (simple activities EA_i or complex activities CA_i). First, every intermediate sequence of $CAC_i \in CAC_j^{Cl}$ is examined, if it is element of all other $CAC_n \in CAC_j^{Cl}$. Figure 6 shows the sequence $d \in ACl_t$ that is associated to all CAC_n. The number of elements in ACl_t is equal (or approximately equal) to elements of CAC_j^{Cl}, therefore, ACl_t with its elements is associated to CAC_j^{Cl}. However, the sequence $Kk \in ACl$ is not associated to CAC_j^{Cl}, because this one occurs only once. Second,

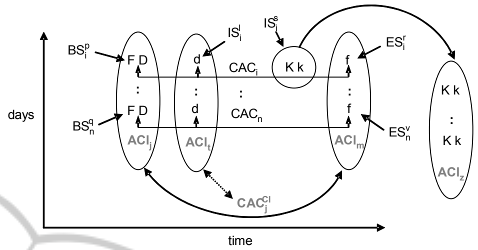


Figure 6: Example data for association of intermediate sequences to complex activity candidate cluster CAC_j^{Cl}. The letters represent the same appliance switchings as in Figure 4. Here the sequence "Kk" (representing activity "desk work") is a parallel/interrupting activity in CAC_i. $CAC_i, \ldots, CAC_n$ represent the complex activity "watching TV".

the transitive dependencies are recursively solved, but this never ocurred in our executed studies. Furthermore, an element of detected activities is parallel or interrupting, if it sometimes occurs in another activity. For example, in Figure 6, the sequence Kk can be a parallel/interrupting activity.

4 EVALUATION

For the experimental evaluation of the system, we conducted two field studies. The first study was carried out in a two-room apartment (cf. Figure 7), the second study in a three-room apartment (cf. Figure 8). In every apartment, an elderly person over 70 years old lived alone during the study. For every electric circuit, a power sensor was installed in the fuse box. Eighteen appliances in the two-room apartment and twenty-two appliances in the three-room apartment were monitored (cf. Table 2). Appliances under the threshold of $P = 35\ W$ have not been monitored (i.e., a radio and two energy saving bulbs in the first study, and eight energy saving bulbs in the second study). During the installation of the system, each appliance was learned by two or three training data in a supervised way. Every resident was asked to log the switchings of appliances and the corresponding activities (e.g. "meal preparation"). Since the logs are usually incomplete (Kasteren et al., 2008), a wireless motion sensor was installed in every room for further manual labeling of appliance switchings. In the first study, 28 days and in the second study, 18 days of activities have manually been labeled. For the activities ground truth does not exist, because the logs were incomplete and the activities could not manually be labeled in contrast to switching appliances.

Table 2: Overview of the installations used in the field studies.

Apartments	2-room apartment	3-room apartment
No. of electric circuit	3	4
No. of appliances	18	22
Period of data collection	5 months	1 month

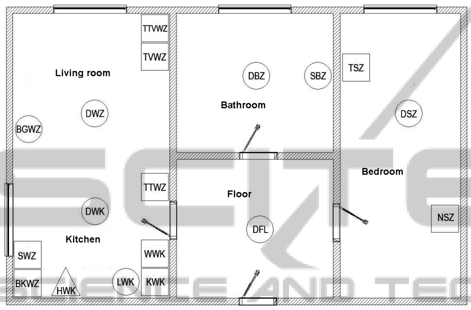


Figure 7: Floor plan of 2-room apartment. The acronyms represent appliances and the shapes (e.g. circle) the associated electric circuits.

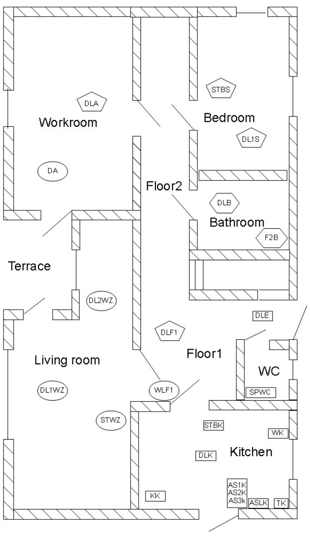


Figure 8: Floor plan of 3-room apartment. The acronyms represent appliances and the shapes the associated electric circuits.

4.1 Appliance Classification

In both field studies, the same thresholds (θ_1 = 35 W, θ_2 = 20 W with window sizes $\sigma_1 = 3\ s$ (turn-on) and $\sigma_2 = 1\ s$ (turn-off)) are used for edge detection. The thresholds have been determined empirically in different tests. In the first study, 40 appliance switchings of $2,661$ manually labeled switchings were not detected correctly. In the second study, the detection of seven switchings of $1,518$ manually labeled have failed. The reason for the larger number of failures in the first study was that two different light switches were installed closely together. The test person could execute two switchings of different appliances within three seconds. But the edge detection can only detect one switching within three seconds (due to the window size). The high number of appliance switchings in first study was related to a defect refrigerator. This one was active (turn-on/turn-off) every hour.

First, we compare the three classifiers K-nearest-neighbors with $K = 1$ (1-NN) (Cover and Hart, 1967), naïve bayes (Mitchell, 1997) and decision trees (C4.5) (Quinlan, 1993). For the classifier C4.5 Weka framework was used and the other both classifiers have been implemented. Tables 3 and 4 show the recognition accuracies w.r.t. different training sets. In the first column of both tables, only the training patterns generated during installation are used. For the test data, all manually labeled patterns are used. Here, the 1-NN classifier shows the best results with over 96% accuracy. With an increase of the training set size (second and third column), the precision of 1-NN only increases slightly in contrast to the other classifiers. But in both studies, 1-NN achieves the best results for each training set size. Since the generation of labeled training data during installation is a realistic scenario, we emphasize that these results are relevant in practice.

Table 3: Classification accuracy of the three classifiers w.r.t. different training sets in the first experimental study (training data/test data). Training data from installation does not contain the not detected edges (40 switchings).

	2 - 3 training data per appliance (92/2621)	data of 1 week (801/1912)	data of 2 weeks (1454/1259)
1-NN	96.9%	97.9%	97.9%
C4.5	80.6%	96.6%	97.2%
naïve bayes	91%	93.9%	94.2%

Table 4: Classification accuracy of the three classifiers w.r.t. different training sets in the second experimental study (training data/test data). Training data from installation does not contain the not detected edges (7 switchings).

	2 - 3 training data per appliance (122/1511)	data of 1 week (816/817)
1-NN	96.2%	96.36%
C4.5	78.1%	95.96%
naïve bayes	91.4%	94.14%

With the extension that the turn-off events are classified depending on known turn-on events, the precision of the 1-NN classifier based on the installation training set is increased. The precision in the first field study becomes 97.62%, the corresponding value in the second study achieves 96.4%. *Sensitivity*, *specificity*, *positive predictive value (ppv)* and *negative prediction value (npv)* have been determined for measure of performance fom 1-NN classifier with extension. *Specificity* and *negative prediction value* were between 97% and 100% for all appliances in both studies. In the first study, there are three major outliers in the *sensitivity* and *positive predictive value* (cf. Figure 9). Appliance "table lamp" was often wrongly classified as "shelf lighting" and vice versa. The reason for this was that the two appliances were very similar and the turn-on of "table lamp" was often interrupted by noise of appliance "TV". Furthermore, appliance "extracted hood" from first and second study (cf. Figure 10) are rarely used (three in first study and five times in second study). This appliance was often wrong classified.

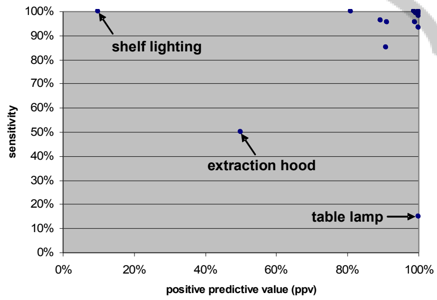


Figure 9: *Sensitivities* and *positive predictive value* of all appliances from first study.

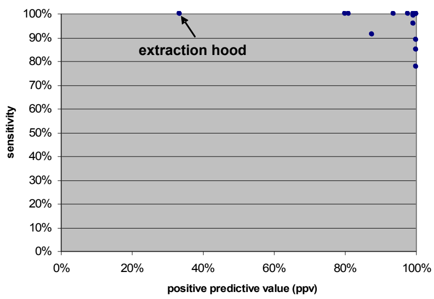


Figure 10: *Sensitivities* and *positive predictive value* of all appliances from second study.

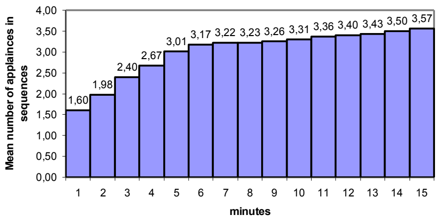
4.2 Activity Recognition

The activity recognition procedure is evaluated with the complete data from both studies. The first study contains data collect on 140 days (about five months), the second study contains data from 27 days. The number of different days with a daily activity (threshold H of the activity recognition procedure) has been determined empirically. This threshold was set to 100 days of 140 days in the first study and 17 days of 27 days in the second study, respectively. The two appliances "table lamp in living room" and "shelf lighting in living room" (cf. Figure 9) are often mixed up by the appliance recognition (probability from the confusion matrix). This was considered in the substitutions matrix, cf. Equation 8.

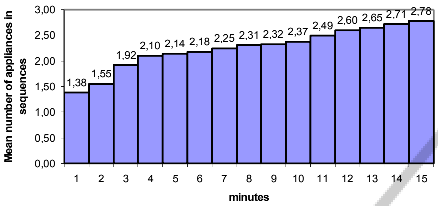
Table 5: Simple activities of the first study (b = bathroom; f = floor).

No.	inferred activities	appliances
A_1	toileting during day	ceiling light (f) on; mirror lamp (b) on; mirror lamp (b) off; ceiling light (f) off
A_2	toileting during day	mirror lamp (b) on/off
A_3	toileting during night	bedside lamp on; mirror lamp (b) on/off; bedside lamp off;
A_4	afternoon tea	kettle on/off
A_5	various activities	ceiling light (f) on/off

The threshold for the similarity of sequences (cf. Equation 9) was set to 70% for both studies. The times for creating sequences (threshold T of Equation 6) was determined by histograms. The histograms of both studies (cf. Figure 11) represent the average number of appliances in sequence by time. The average number of appliances was computed for every time with the help of equation 6. For the threshold T the times (bars from histograms) which show the smallest distance of two adjacent bars have been chosen (for first study $T = 7$ minutes and for second study $T = 8$ minutes). Furthermore, stand-alone running appliances (e.g., the refrigerator) are filtered out manually. Tables 5 and 6 show the detected *simple activities* of both studies. In Tables 7 and 8 the detected *complex activities* are shown with the associated similarity clusters ACl_i. The tables also show the positions (rooms) of some appliances for a clear identification. The names of the activities are inferred from the protocols and the involved appliances in the tables. The order of the appliances in each activity is only one representative of several that have occurred in reality. Some activities are recognized multiple times. For example, the *simple activity* "toileting during day" has been detected twice, because the same activity is executed with different appliances. The algorithms cannot distinguish between both activities. Furthermore, two different activities can be detected as one, e.g., the detected *complex activity* A_7 of the second study is in fact divided into the activities "preparing morning coffee" and "preparing evening



(a)



(b)

Figure 11: Histograms with an average number of appliances in sequences at different times: (a) first study; (b) second study.

Table 6: Simple activities of the second study (b = bathroom; be = bedroom; f = floor).

No.	inferred activities	appliances
A_1	toileting during night	ceiling light (be) on; ceiling light (b) on; ceiling light (b) off; ceiling light (be) off
A_2	toileting during day	ceiling light (f) on; ceiling light (b) on; ceiling light (b) off; ceiling light (f) off
A_3	preparation for sleep	ceiling light (f) on; ceiling light (b) on; ceiling light (b) off; ceiling light (be) on; ceiling light (f) off; ceiling light (be) off
A_4	various activities	ceiling light (be) on/off
A_5	various activities	ceiling light (f) on/off

Table 7: Complex activities of the first study (b = bathroom; be = bedroom; f = floor; l = living room).

No.	Inferred activities	cluster	appliances
A_6	breakfast	ACl_1	ceiling light (l) on; kettle on; kettle off
		ACl_2	ceiling light (l) off
A_7	afternoon nap	ACl_1	bedside lamp on
		ACl_2	bedside lamp off
A_8	afternoon TV	ACl_1	TV on
		ACl_2	TV off
A_9	evening TV, reading and puzzles	ACl_1	ceiling light (l) on
		ACl_2	TV on; table lamp (l) on
		ACl_3	TV off; table lamp (l) off
		ACl_4	ACl_1 of A_{10}
A_{10}	preparation for sleep	ACl_1	mirror lamp (b) on; mirror lamp (b) off; ceiling light (l) off; ceiling light (be) on; bedside lamp on; ceiling light (be) off;
		ACl_2	bedside lamp off
A_{11}	various activities	ACl_1	ceiling light (f) on
		ACl_2	ceiling light (f) off

Table 8: Complex activities of the second study (k = kitchen).

No.	Inferred activities	cluster	appliances
A_6	breakfast	ACL_1	ceiling light (k) on; kettle on; toaster on; kettle off; toaster off
		ACl_2	ceiling light (k) off
A_7	preparing morning coffee/ evening coffee	ACl_1	ceiling light (k) on; kettle on/off
		ACl_2	ceiling light (k) off
A_8	preparing meal	ACl_1	hood light on
		ACl_2	hood light off

coffee". For these activities, the same appliances are used. This confusion can be solved by considering the time of the day. It is possible that one similarity cluster (ACl_i) belongs to different activities, e.g., the detected complex activities A_9 and A_{10} share the same ACl_i. If these two activities should be recognized as one activity, the algorithm can be adapted easily. Furthermore, only a few recognized activities, which are named as *various activities* in the tables, cannot be inferred to an unambiguous activity name, because they appear in different activities. Finally, there are sometimes larger variations (transpositions) of elements in the same sequences, especially in long sequences (e.g., A_3 in Table 6). The algorithm can only detect adjacent transpositions. This issue is currently solved via the threshold H.

As parallel/interrupting activities, activity "toileting during day" (A_1 of the first study) appeared in the activities (A_8 and A_9) with longer duration. In the second study, no parallel/interrupting activities were detected. This is certainly associated with the fact that the detected complex activities had a shorter duration.

5 CONCLUSIONS

In our two experimental studies, we were able to demonstrate the capabilities of our system for activity recognition based on appliance switchings. The approach is capable of detecting *simple* and *complex activities*. Furthermore, the algorithm can detect parallel/interrupting activities and can consider noise as wrong classified appliances. A major problem evaluating the activity detection occured in verifying the ground truth since the logs were incomplete. Furthermore, evening activities could not be recognized in the second study since energy saving bulbs have been used. They can not be detected by the appliance detection. An increased use of saving bulbs could lead to future problems of activity recognition, because the recognized activities of the two studies often include lamps. Furthermore, when at the beginning and at the end of the day an appliance is switched (e.g. "aquarium lighting") the presented algorithm for activity recognition would identify the various activities during the day as one activity. This case did not occur in the two studies. One approach to solve this could be a maximum time duration that a valid activity can have. Some recognized activities can be easily determined by specifying significant appliances (e.g. activity "breakfast" often contains appliance "toaster" and "kettle"). But other activities that are not previously known or are very individual turn out to be difficult to detect (e.g. "afternoon nap"). The presented approach is able to recognize such activities in an unsupervised kind of way. In future works, we plan to investigate, if the number of days with daily activities can be increased by recognizing larger transpositions of elements in sequences instead of only adjacent transpositions. Furthermore, it will be investigated, how stand-alone running appliances, e.g., refrigerators, can be detected automatically.

REFERENCES

Barger, T. S., Brown, D. E., and Alwan, M. (2005). Health-status monitoring through analysis of behavioral patterns. *Trans. Sys. Man Cyber. Part A*, 35(1):22–27.

Chen, J., Zhang, J., Kam, A., and Shue, L. (2005). An automatic acoustic bathroom monitoring system. In *International Symposium on Circuits and Systems (ISCAS 2005)*, pages 1750–1753 Vol. 2. IEEE.

Chen, L., Hoey, J., Nugent, C., Cook, D., and Yu, Z. (2012). Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 42(6):790 –808.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21 –27.

Guralnik, V. and Haigh, K. Z. (2002). Learning models of human behaviour with sequential patterns. In *Proceedings of the AAAI-02 workshop Automation as Caregiver*, pages 24–30. AAAI Press.

Hart, G. (1992). Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891.

Kasteren, T. v., Noulas, A., Englebienne, G., and Kröse, B. (2008). Accurate activity recognition in a home setting. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 1–9. ACM.

Katz, S. (1983). Assessing self-maintenance: activities of daily living, mobility, and instrumental activities of daily living. *Journal of the American Geriatrics Society*, 31(12):721–727.

Katz, S., Moskowitz, R., Ba, J., and Jaffe, M. (1963). Studies of illness in the aged: The index of adl: a standardized measure of biological and psychosocial function. *Journal of the American Medical Association*, 185(12):914–919.

Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707–710.

Magnetics (2013). CR Magnetics CRD5110. http://www.crmagnetics.com/Products/CRD5100-Series-P30.aspx. Accessed: 2013-07-28.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.

Moeslund, T. B., Hilton, A., and Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.*, 104(2):90–126.

Nguyen, N. T., Phung, D. Q., Venkatesh, S., and Bui, H. (2005). Learning and detecting activities from movement trajectories using the hierarchical hidden markov models. In *In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 955–960. IEEE Computer Society.

Ni Scanaill, C., Carew, S., Barralon, P., Noury, N., Lyons, D., and Lyons, G. M. (2006). A review of approaches to mobility telemonitoring of the elderly in their living environment. *Annals of Biomedical Engineering*, 34(4):547–563.

Noury, N., Berenguer, M., Teyssier, H., Bouzid, M. J., and Giordani, M. (2011). Building an index of activity of inhabitants from their activity on the residential electrical power line. *IEEE Transactions on Information Technology in Biomedicine*, 15(5):758–766.

Oliver, N., Horvitz, E., and Garg, A. (2002). Layered representations for human activity recognition. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, page 3. IEEE Computer Society.

Oommen, B. J. (1997). Pattern recognition of strings with substitutions, insertions, deletions and generalized transpositions. *Pattern Recognition*, 30(5):789–800.

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

Poppe, R. (2010). A survey on vision-based human action recognition. *Image Vision Comput.*, 28(6):976–990.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Virone, G., Alwan, M., Dalal, S., Kell, S. W., Turner, B., Stankovic, J. A., and Felder, R. A. (2008). Behavioral patterns of older adults in assisted living. *IEEE Transactions on Information Technology in Biomedicine*, 12(3):387–398.