Enabling Efficient Time Series Analysis for Wearable Activity Data

Kristof Van Laerhoven, Eugen Berlin and Bernt Schiele
Department of Computer Science
Technische Universität Darmstadt
Darmtadt, Germany
Email: {kristof, berlin, schiele}@cs.tu-darmstadt.de

***Abstract*—Long-term activity recognition relies on wearable sensors that log the physical actions of the wearer, so that these can be analyzed afterwards. Recent progress in this field has made it feasible to log high-resolution inertial data, resulting in increasingly large data sets. We propose the use of piecewise linear approximation techniques to facilitate this analysis. This paper presents a modified version of SWAB to approximate human inertial data as efficiently as possible, together with a matching algorithm to query for similar subsequences in large activity logs. We show that our proposed algorithms are faster on human acceleration streams than the traditional ones while being comparable in accuracy to spot similar actions, benefitting post-analysis of human activity data.**

I. Introduction

Due to recent advances in microelectronics, research in wearable healthcare sensing has undergone a tremendous boost, facilitating small and comfortable sensors that can be worn for longer periods of time (weeks up to months). These new types of devices run on lightweight batteries, and contain the sensors and storage space to record physical motion properties of the person wearing it. Many potential applications have been mentioned for this form of activity recognition research, many of them for the monitoring, revalidation and treatment of patients (e.g. [1]).

Recent work has started to examine the possibility of an extension to actigraphy, where a person's level of physical activity is logged over time to be analyzed periodically. Instead of measuring mere amount of activity, this new research instead aims at detecting what type of activities are present in the captured data. The inertial information produced by these activity sensors is detailed enough to distinguish several types of physical activities such as "*running*", "*taking the stairs*", or "*playing tennis*".

This paper's focus is the challenge that lies in the interactive analysis of large amounts of recorded high-frequent inertial data, where users or researchers can view the activity log as a time series, zoom into interesting areas, and select subsequences that exhibit characteristic patterns for particular activities. The algorithms proposed in this paper then allow the system to search in the rest of the data for similar subsequences, in a reasonable time frame for an interactive application. Figure 1 illustrates a prototype of such an application, where the selecting of subsequences in the time series results in the system finding closest matches elsewhere in the data. This would be beneficial for offline annotation of the data, identification of motion gestures, and detailed comparison of similar activities.

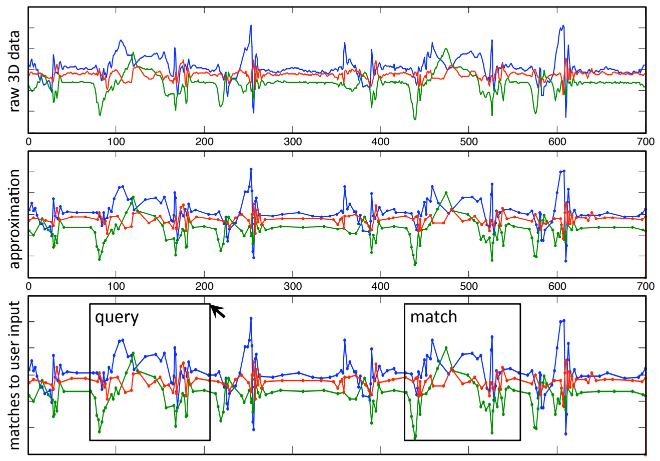


Figure 1: This paper presents approximation and matching algorithms to analyze 3D inertial activity data. Fast matching allows to interactively select a query subsequence in the time series, directly after which the system returns similar ones.

The long-term recording of data tends to result in large databases of sensor samples however, and the analysis techniques on this type of data have thus far been limited to resource-hungry and offline prototypes based on machine learning algorithms such as boosting, topic models, and hidden Markov models [2]. Figure 2 shows part of such an activity data set of 24/7 recording, along with one of the wrist-worn sensors that is used to capture it. The bottom plot shows the motion patterns in the sensor data while the subject was walking.

We propose a two-tier set of algorithms: one that *approximates* the inertial data so that it can be stored in less memory and be processed faster, and one that *matches* subsequences of these approximations to find similarities within a large time series of human activity data. To achieve our above-mentioned goal, several requirements need to be fulfilled: First, as both researchers and subjects have proven to be able to browse time series plots of inertial data well (as

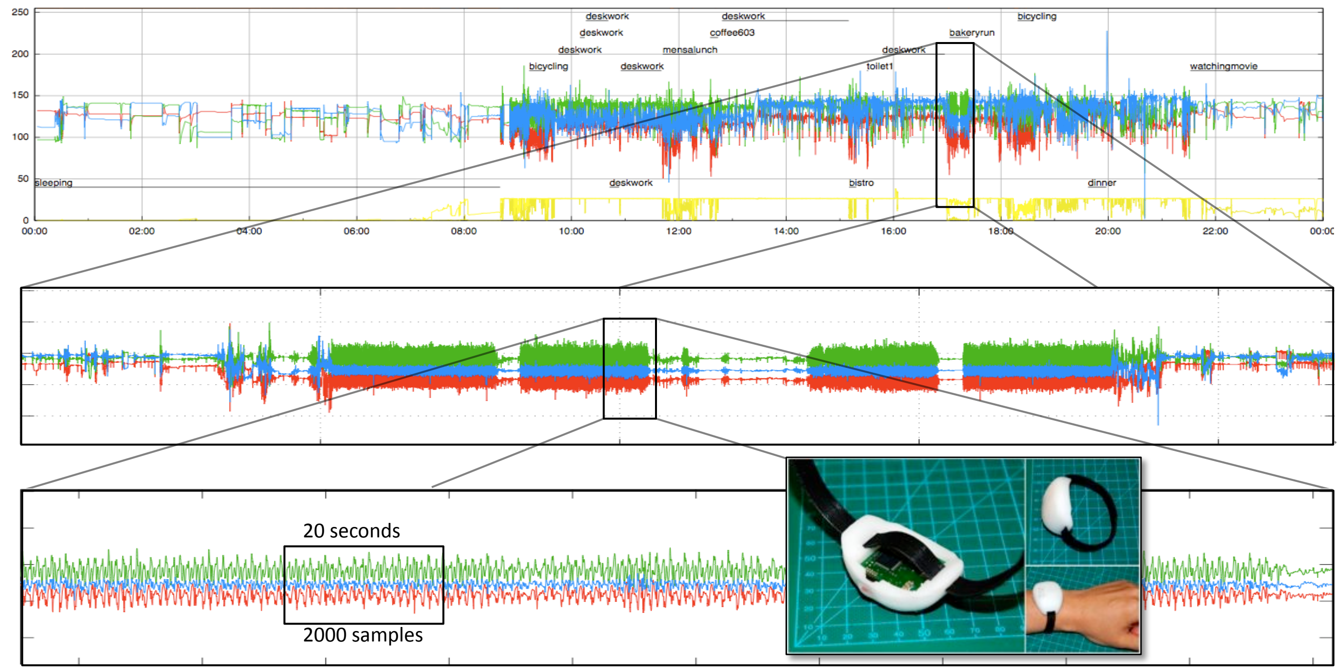


Figure 2: Typical time series plots from a long-term activity dataset, provided by a wrist-worn sensor (in-set picture). Each day, about 8.6 million 3D inertial (accelerometer) samples are complemented with light, temperature, and time stamp data. Acceleration data is focused on as the most descriptive modality; It contains both posture (see for instance the sleeping postures during the night segments) and motion data. The bottom zoomed-in region focuses on data from the subject walking.

mentioned by [3]), the data needs to be approximated so that it can still be represented visually as a time series. Second, the algorithms that do the approximation of data need to be fast enough so that it does not hinder the loading of data, and accurate enough to capture the essence of the motion patterns. Finally, the matching needs to be fast enough to allow interactive applications, while being accurate enough to do qualitative matching.

II. Related Work on Inertial Activity Data

Previous studies have applied a large variety of approximations and features to human accelerometer data. Among the more prominent are (usually calculated over a sliding window) mean and variance, as well as Fourier coefficients, wavelet matches [4], and several approaches applying conversion in sequences of symbols.

The appeal of using the mean and variance as features for acceleration is particularly high because of their efficient implementation. The mean tends to capture the local posture of the body, and variance describes how much motion is present in the signal. The combination of mean and variance has also been used with much success in detecting high-level activities by calculating them over large sliding windows [5]. These features have been used effectively when combining multiple body-worn sensors [6] or in short sliding windows with an HMM-based approach [2].

Several features are in contrast more costly to calculate but have resulted in better performance. Autocorrelation, Discrete Fourier Transform, and filterbank analysis can be expected to work especially well on activities with dominant frequencies, and have been identified as superior in several comparison studies (e.g., [7]).

Other approaches quantize the data in symbol strings and look for sequences in this data. Minnen et al. [8] proposed to discretize the inertial data first by fitting K Gaussians to achieve a roughly equal distribution of symbols, after which repetitive sub-sequences in the data, so-called motifs, are searched that in turn train HMMs to allow unsupervised learning of activities. Later work used the symbolic aggregate approximation (SAX [9]) algorithm as a way to represent the time series, which was further improved to iSAX [10]. Symbolic methods are known to be fast, but are not as easy to visualize as the ones covered in this work.

Apart from [11], where the authors use the SWAB algorithm on fused acceleration and gyroscope data to detect relevant gestures made by the wearer of the sensor, piecewise linear approximation of wearable inertial data has thus far not often been explored. Work on matching has thus far used mostly Euclidean distance over features or dynamic time warping (DTW) [12] to compare subsequences in inertial sensor data, or window based classification, gesture spotting [6], or motif discovery [8].

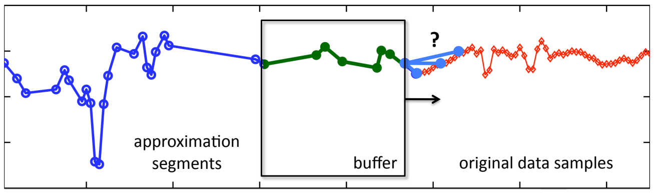


Figure 3: SWAB approximates a time series to a set of segments, with a sliding buffer in which local bottom-up segmentation is done [13]. Our adaptation replaces the Sliding Window phase by moving the buffer up to the data point where the slope changes sign.

III. Approximation

This section introduces the algorithm proposed in this paper to approximate human acceleration time series efficiently, so that it is represented by a smaller amount of data without losing the intrinsic nature of the underlying activity.

A. SWAB

Key to our approach is the approximation of a time series of human acceleration data, into a representation of linear segments that is efficient to manipulate and faster to process than the raw sensor data. The linear segments can be visualized in an identical way to the original data in a time series plot, while the number of data points is reduced. The work presented here is based on SWAB [13], and is in essence an adaption of this algorithm specifically for physical activity data from accelerometer data.

The original algorithm works by approximating the time series by well-chosen linear segments that as [13] showed are closer to the data than the online Sliding Windows approach, while being still an online approach. Figure 3 sketches the basic operation of SWAB: A buffer window is slid over the original time series, in which Bottom-Up segmentation is performed. The left-most segment is then produced as the next approximated segment, while the buffer window is moved to the right, to the next original data point for which the Sliding Window approximation cost overruns a threshold. More precisely if the segment between the ith and jth data points, x_i and x_j respectively, is called S, the cost of approximation of the subsequence $(x_i, x_{i+1}..., x_j)$ by S is calculated by

$$c(x_i..x_j, S) = \sqrt{\sum_{n=i}^{j}\left(x_n - (x_i + (n-i) * \frac{x_j - x_i}{j-i})\right)^2},$$

which is then done for every new data point x_j until the cost c overruns the cost threshold. In that case, the new buffer is extended to x_{j-1} and the next approximation segment is searched with the Bottom-Up approach in the buffer.

Listing 1: Here the original SWAB algorithm [13] abstracting $timeseries$ with cost threshold T, has the Sliding Window heuristic modified to increase the algorithm's speed. To create less data, segments with similar slopes are merged.

```
[segs] = mSWABsegs(timeseries, len, T)
  win_left=1; win_right=bufsize;
  while(1) % while new data:
    swabbuf=timeseries[win_left:win_right];
    % Bottom-Up segementation of buffer:
    BUsegs(swabbuf,bufsize,BUsegs,T);
    % add left-most segment from BU:
    segs = [segs; BUsegs(2)]; n = size(segs);
    % merge last segments if slope is equal:
    if slope(segs(n)) == slope(segs(n-1)),
      merge_last2(segs); n = n-1; end;
    % shift left of buffer window:
    win_left = BUSegs(2).x;
    % shift right of buffer window:
    if (win_right<len),
      i=win_right+1;
      s = sgn(slope(i,i-1));
      while sgn(slope(i,i-1))==s),
        i=i+1;
      end;
      win_right=i;
      bufsize=win_right-win_left;
    else
      % all done, flush buffer segments:
      segs = [segs; BUsegs];
      break;
    end;
  end;
```

B. mSWAB

The authors of [13] mention that optimizations are possible for particular data, by for instance incrementing the sliding window with multiple samples instead of one (which showed beneficial in case of ECG data). SWAB's standard version moves a sliding window, recalculating an approximation cost and matching it to a threshold for every additional sample of raw data (Figure 3).

Our adaptation exploits the property of accelerometer data, which tends to heavily fluctuate by producing characteristic peaks in the time series, and instead moves the window on to the next data point when the slope's sign changes between positive and negative, or zero. This means that instead of having to iteratively calculate c, one simply has to calculate the slope between adjacent data points x_j and x_{j+1} and stop when the signum function changes value, or $sgn(x_j - x_{j-1}) = sgn(x_{j+1} - x_j)$.

This speeds up the process as it requires a single test per sample ($O(n)$ with n the samples the buffer is shifted over), instead of recalculating costs over the segment ($O(n^2)$ regardless whether sum of squares or the L_∞ norm is used for the cost calculation). Although the Bottom-Up part of SWAB remains still costly, substituting the Sliding

Window approach leads to a significant effect when the accelerometer data is sampled at a high frequency or in constant subsequences (i.e., when no movement occurs). The latter occurs very frequently, especially in long-term data which include resting and sleeping segments.

A second change to the original algorithm uses a suggestion made by [14] to merge the last two produced segments if their slope is the same. Listing 1 shows the source code, highlighting the differences to the original SWAB algorithm.

C. *Evaluating the Approximations*

In order to test the performance of mSWAB, we verify two claims: (1) whether mSWAB does indeed approximate the original accelerometer data faster than the generic SWAB, and (2) whether the quality of approximation is indeed comparable to that of SWAB.

In order to obtain meaningful results for our proposed application, three data sets of activity data were used that are similar in structure and configuration to the one depicted in Figure 2. Each data set contains the continuous 3D acceleration data from a wrist-worn sensor, and spans between 24 and 48 hours. The subjects that were monitored in these can be characterized as leading a regular life with normal levels of activity. To have a comparable execution speed comparison, the Sliding Window, the SWAB, and the mSWAB algorithms were implemented in C++[1], and compiled with full optimization turned on. The tests were done on a standard Pentium 5 processor running at 3.2GHz.

The left plot in Figure 4 shows the residual error (i.e., the sum of squares of the vertical differences between original data and approximation segment, for all segments) for the Sliding Window, the SWAB algorithm, and the proposed mSWAB algorithm, for a cost threshold set between 1 and 50. The initial buffer size was set to 100, which is essentially the amount of raw data streaming in per second. Prior verification has shown that this value works well and contains the recommended 5 to 6 segments mentioned in by Keogh. There is little difference between the performances of SWAB and mSWAB, confirming that the Bottom-Up buffer within the algorithm works identically for both implementations. The results also further confirm those from [13], showing that the approximation segments from the Sliding Windows algorithm are further apart from the original data than those of SWAB and mSWAB.

The right plot in Figure 4 shows the execution speed in seconds, for the cost threshold of the approximation algorithms between 1 and 50. Identically to the residual error plot, the initial buffer size for SWAB and mSWAB was set to 100. Sliding Windows can be seen to be in the same range for a cost threshold of one, then veering off and steadily increasing as the cost threshold increases. The

[1]The code for these and further discussed algorithms, as well as the evaluation scripts and data sets, can be obtained by downloading from http://porcupine2.sf.net or mailing the paper's authors.

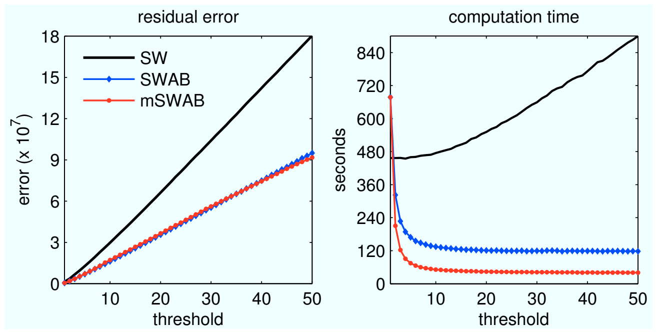


Figure 4: mSWAB evaluated against the Sliding Windows and SWAB algorithms on the long-term data. Left: the residual error for a varying cost threshold. Right: the time in seconds needed to approximate the data. Initial buffer size for SWAB and mSWAB was set to 100.

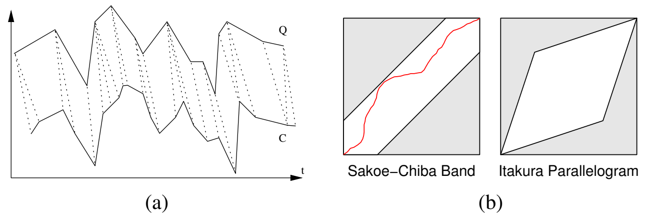


Figure 5: a) Matching subsequences Q and C with dynamic time warping (DTW) aligns data points to the optimal counterpart (dotted lines). b) DTW is often bounded, for instance with the SakoeChiba Band or the Itakura Parallelogram, restricting the warping paths by local or global constraints (e.g., white areas in the plots).

mSWAB algorithm does indeed display a faster execution speed compared to SWAB, owing to the sliding heuristic of the buffer window: Instead of steadily increasing the segment and re-doing the cost calculation over an increasing set of data points, the change of slope between successive data points is monitored.

IV. Subsequence Matching

After the approximation of the original inertial data, this section covers the matching in nearest neighbors-based classification of these approximations. As a faster alternative to dynamic time warping, a heuristic using Euclidean distances between the K longest segments is introduced.

A. *Dynamic Time Warping*

Dynamic time warping (DTW) is a widely used technique used in speech recognition, information retrieval and machine learning, to overcome small distortions in time between two time series. Given two subsequences, DTW optimally aligns or 'warps' the data points between the two time series (Figure 5a) and returns their distance, which then can be used in classifiers as a similarity measure.

To align two time series $Q = q_1, ..., q_n$ of length n and $C = c_1, ..., c_m$ of length m with DTW, an n-by-m matrix with squared distances of the time series elements q_i and c_j is created, and an optimal 'warping path' that characterizes the alignment of Q and C and minimizes the warping costs is computed. The warping path cost for distance matrix entry (i, j) can for instance be recursively computed with the distance function $\gamma(i, j) = d(i, j) + min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\}$.

The general approach, which computes all the squared distances in the matrix and chooses the minimal continuous path, is of high time complexity ($O(n \cdot m)$). In practice, different local or global constraints can be used to decrease the number of paths that will be computed during alignment process, thus significantly speeding up the calculation. In our implementations we considered two common bounding techniques: the SakoeChiba Band and the Itakura Parallelogram, where only paths are considered that lie within certain bounds (see Figure 5b). Other lower bounding techniques as well as new approaches for DTW have been also recently discussed and presented in [15] and [16].

B. *K Longest Segments*

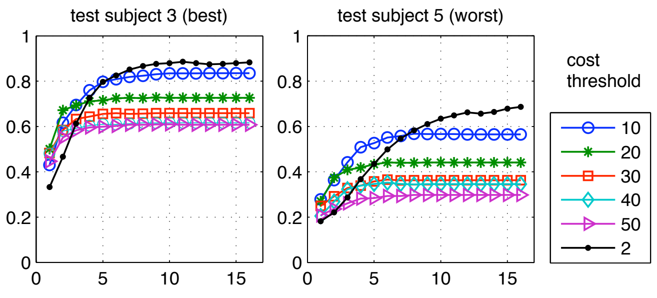
Most subsequences of interest tend to have a high number of segments, resulting in slow matching when done with Euclidean matching or dynamic time warping. In order to speed up matching, we propose to limit the subsequences to those segments that are likely to be most descriptive. These in our data are assumed to be the K longest segments per dimension. We argue that this is sensible as the large segments tend to cover either large peaks or large stable regions in the subsequences for our accelerometer data, both of which are important for characterizing motion patterns within physical activities.

For matching, these K longest segments are selected and compared against the segments in the subsequence it is compared to. The distances to the closest matching segments, using Euclidean distance, are then summed to obtain a distance between the two subsequences. By filling the second subsequence with the contents of a sliding window over the entire time series, closest matching subsequences to a query subsequence can be found.

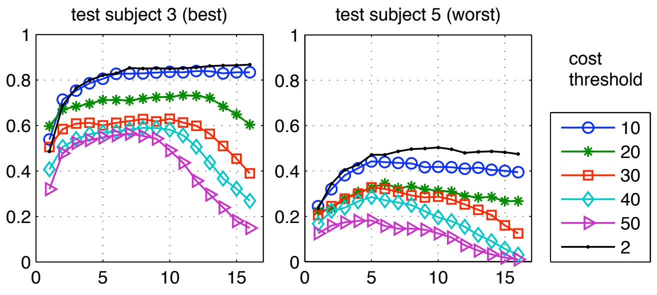
The choice of the number of segments K greatly affects the speed of the algorithm, as well as the accuracy of approximation. The higher K becomes, the more distinctive the resulting set of segments will be in matching and the more time is needed to find closest matches to all segments.

C. *Evaluation of Matching Methods*

To evaluate how accurate the matching works of the DTW and K longest segments methods, we use a data set that contains 15 very similar target classes: For 5 test subjects, three person-specific activities are recorded that are known to be highly challenging in activity recognition: "*walking*", "*climbing stairs*" and "*descending stairs*". The data set incorporates fatigue and sensor strap loosening by recording per test subject all activities 5 times in a row and on two different days, resulting in many examples per class with inter-subject deviations. The entire data set consists of about 1.1 million samples, spanning over 2 hours.



(a) Accuracies using DTW with SC-band from 1 to 16



(b) Accuracies using K longest segments with K from 1 to 16

Figure 6: Accuracy results for the best and worst test subjects in the data set, approximated with mSWAB for a range of cost thresholds, and classifying with nearest neighbours using: a) dynamic time warping (DTW) with SakoeChiba (SC) band, or b) K longest segments.

The above-described DTW and K longest segments algorithms were used to match and classify via nearest neighbors classification the training part of this data set, using 30-fold cross validation, to the remainder testing part. DTW was used together with a SakoeChiba band being varied from 1 to 16, as was the parameter for K longest segments, K. The target classes were chosen to be as challenging as possible, not only containing the activity but also which person performed the activity. Detections in unlabeled ('background') data were counted as false positives.

As illustrated by the best-performing data in the left plots of Figure 6, K longest segments tends to equal the performance of DTW for SC band and K of 10, and using a low enough cost threshold. The right plots in Figure 6 illustrate that DTW performs in few isolated cases considerably better with lower cost thresholds. DTW however, as can be witnessed in Table I, results exactly then in far longer execution times. It is therefore important in future work to consider data from more test subjects.

Table I: Comparing DTW and K longest segments as in Figure 6 with accuracy and execution time, using 10 for both SC-band width and K, and varying cost threshold.

ct	DTW accur.		KLS accur.		time (sec.)	
	worst	best	worst	best	DTW	KLS
2	63.4	88.0	50.3	85.0	88.6	12.7
10	56.6	83.4	41.7	83.5	15.5	7.5
20	44.1	72.6	31.2	72.6	7.0	3.6
30	36.3	65.8	28.6	63.0	4.0	2.2
40	34.5	61.6	19.5	58.2	2.5	1.5
50	29.7	61.0	12.4	49.1	1.7	1.1

V. Conclusions and Future Work

Long-term activity recognition has, due to advances in miniature sensing techniques, become a field in need for fast machine learning techniques. Recordings of long-term inertial sensing trials tend to be many and large, producing time series which are hard to analyze using conventional classification techniques. This paper proposes two algorithms that are suitable for query-by-content browsing of such data.

A segment-based approximation algorithm for time series is presented, based on the SWAB algorithm and modified to work faster on human activity data. It was shown on such long-term accelerometer data sets that the residual error of the representation is nearly similar to that of the technique it is based on, while being faster to execute.

Additionally, a matching routine for the approximated data was proposed that uses the Euclidean distance between the K largest segments in the approximation. This method has in an empirical study shown to be about twice as fast than bounded dynamic time warping, while resulting in similar accuracies when the value K is chosen well.

This work will be explored further by using the proposed algorithms with learning methodologies that are capable of modeling richer representations of activities, such as models capable of learning (sets of) typical motion subsequences per activity.

Acknowledgements

This work was sponsored by the Project "Methods for Activity Spotting With On-Body Sensors" from the German Research Foundation (DFG).

References

[1] K. V. Laerhoven, H.-W. Gellersen, and Y. G. Malliaris, "Long-term activity monitoring with a wearable sensor node," in *2006 International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2006)*. IEEE Computer Society, 2006, pp. 171–174.

[2] J. A. Ward, P. Lukowicz, G. Tröster, and T. E. Starner, "Activity recognition of assembly tasks using body-worn microphones and accelerometers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1553–1567, 2006.

[3] K. V. Laerhoven, D. Kilian, and B. Schiele, "Using rhythm awareness in long-term activity recognition," in *Proceedings of the 12th IEEE International Symposium on Wearable Computers*. IEEE Computer Society, 2008, pp. 63–68.

[4] N. Rajpoot and K. Masood, "Human gait recognition with 3dwavelets and kernel based subspace projections," in *Human Activity Recognition and Modeling (HAREM)*, 2005.

[5] T. Huynh, U. Blanke, and B. Schiele, "Scalable recognition of daily activities with wearable sensors," in *Proceedings of the 3rd International Symposium on Location- and Context-Awareness (LoCA)*, 2007.

[6] G. Ogris, T. Stiefmeier, P. Lukowicz, and G. Tröster, "Using a complex multi-modal on-body sensor system for activity spotting," in *Proceedings of the 12th IEEE International Symposium on Wearable Computers*, 2008, pp. 55–62.

[7] T. Huynh and B. Schiele, "Analyzing features for activity recognition," in *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies,*Grenoble. ACM Press New York, NY, USA, 2005, pp. 159–163.

[8] D. Minnen, T. Starner, I. Essa, and C. Isbell, "Discovering characteristic actions from on-body sensor data," in *International Symposium on Wearable Computers (ISWC)*, 2006.

[9] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: a novel symbolic representation of time series," *DMKD Journal*, 2007.

[10] J. Shieh and E. Keogh, "iSAX: Indexing and mining terabyte sized time series," in *SIGKDD 2008*, 2008.

[11] O. Amft, H. Junker, and G. Tröster, "Detection of eating and drinking arm gestures using inertial body-worn sensors," in *ISWC 2005: IEEE Proceedings of the Ninth International Symposium on Wearable Computers*. IEEE Press, 2005, pp. 160–163.

[12] K. M. Hsiao, G. West, and S. Vedatesh, "Online context recognition in mul-tisensor system using dynamic time warping," in *Proc. of the 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2005, pp. 283–288.

[13] E. J. Keogh, Chu, Hart, and Pazzani, "An online algorithm for segmenting time series," in *IEEE International Conference on Data Mining*, 2001, pp. 289–296.

[14] H. Junker, "Human activity and gesture spotting with body-worn sensors," Ph.D. dissertation, ETH Zürich, 2005.

[15] C. Ratanamahatana and E. Keogh, "Making time-series classification more accurate using learned constraint," in *SIAM International Conference on Data Mining*, 2004.

[16] D. Lemire, "Faster retrieval with a two-pass dynamic-time-warping lower bound," *Pattern Recogn.*, vol. 42, no. 9, pp. 2169–2180, 2009.