Mobile Netw Appl (2009) 14:82–91
DOI 10.1007/s11036-008-0112-y

An Activity Recognition System For Mobile Phones

Norbert Győrbíró · Ákos Fábián · Gergely Hományi

Published online: 7 November 2008

Abstract We present a novel system that recognizes and records the motional activities of a person using a mobile phone. Wireless sensors measuring the intensity of motions are attached to body parts of the user. Sensory data is collected by a mobile application that recognizes prelearnt activities in real-time. For efficient motion pattern recognition of gestures and postures, feed-forward backpropagation neural networks are adopted. The design and implementation of the system are presented along with the records of our experiences. Results show high recognition rates for distinguishing among six different motion patterns. The recognized activity can be used as an additional retrieval key in an extensive mobile memory recording and sharing project. Power consumption measurements of the wireless communication and the recognition algorithm are provided to characterize the resource requirements of the system.

Keywords motion recognition · motion sensors · mobile computing · lifelogging · neural networks

N. Győrbíró (✉)
Spatial Media Group, University of Aizu, Aizu-Wakamatsu, Fukushima-ken 965-8580, Japan
e-mail: ngyorbiro@acm.org

Á. Fábián
Budapest University of Technology and Economics, Budapest, Hungary

G. Hományi
Nokia Siemens Networks, Köztelek utca 6., Budapest 1092, Hungary

1 Introduction

Many systems have been developed to collect data from a person's everyday life for later retrieval (see, e.g., [7, 21]). Their main purpose is to enhance human memory by utilizing the capabilities of computers. They can record e-mail and chat conversations, documents, location information, photographic, audio, and video audio content using cameras and microphones, and many other types of personal and environmental data capture.

In our research project, we have focused on data that can be acquired on mobile phones. In recent years mobile phones have become very personal communication devices. Besides storing contacts, messages, and calendar appointments, the newer generations of mobile devices, called *smartphones*, can be used for taking photos, recording audio clips, browsing the internet, and accessing e-mail. Therefore, an enormous amount of personal data can be collected even during a short period of time, so the recall of a particular event is very difficult. Storage of additional metadata can aid navigation through recorded content. The activity of a person can be a useful cue for retrieving memories. When it is used for annotating content data, one can build elaborated search queries like "*Who called me while I was jogging in the park?*"

In this paper we present a mobile system that recognizes motional activities of a person. This information is recorded along with a personal memory archive. Other novel application possibilities are also described. We have focused attention on power-efficient approaches since it needs to be running on mobile devices offering realtime motion recognition capabilities.

2 Related work

The idea of enhancing memory retrieval with additional cues is not new. Lamming and Flynn [12] utilized physical context information such as location, phone calls, and the interaction between different PDAS as retrieval keys. Kern et al. [9] annotated meeting recordings with motional activities such as standing and sitting to differentiate between presentation and discussion sessions. In order to recognize postures, they utilized body-worn accelerometer sensors.

The usage of accelerometers for physical activity recognition is widely described in the literature. Randell and Muller [19] used a single biaxial accelerometer to calculate the RMS and integrated values over the last two seconds for both axes. This input data was used for classifying six activities (walking, running, sitting, walking upstairs, walking downstairs, and standing) utilizing a neural network. Mäntyjärvi et al. [16] also applied neural networks to human motion recognition. Their feature vector was created with PCA (Principal Component Analysis) and ICA (Independent Component Analysis) from a pair of triaxial sensors, attached to the left and right hips. Lee and Mase [14] developed an activity and location recognition system using a combination of a biaxial accelerometer, compass, and gyroscope. Their classification technique was based on a fuzzy-logic reasoning method.

The above studies relied upon wired sensors, which could be uncomfortable to wear. Therefore it may be difficult to perform outdoor or long-term experiments. Lately, wireless accelerometers have become available, enabling measurements in more comfortable settings. Bao and Intille conducted an extensive study [2] with twenty subjects using five wireless biaxial sensors. Sensory information was processed by FFT (Fast Fourier Transform) to extract means, energy, frequency-domain entropy, and correlation features. Recognition of twenty different everyday activities from these features was performed using decision table, instance-based learning, C4.5 decision tree, and naïve Bayes classifiers from the Weka Machine Learning Algorithms Toolkit. Decision tree classification delivered the best results on the derived feature vector. Utilizing the same toolkit but only a single triaxial accelerometer worn in the pelvic region, Ravi et al. [20] studied the performance of base-level classifier algorithms as well as the meta-classifiers, including voting, stacking, and cascading frameworks. Plurality voting performed the best in those settings.

Some newer models of smartphones are now equipped with motion detection technology, either by including a small accelerometer (e.g. Nokia N95) or using the built-in camera (e.g. Panasonic P904i). Such solutions are convenient for measurements, but their range of applicability is limited by the single sensor. Ji Soo Yi et al. [22] conducted a context awareness study to detect changes in mobility under various lighting conditions using the output of a single tri-axial accelerometer attached to a handheld computer. For human activity recognition, the deployment of multiple accelerometers not only improves the results but is rather mandatory, as noted by Kern et al. [10]. Further work [11] utilizes recognized activity and meta-information from audio for contextual cues in live life recording using a laptop and a wired sensor-based system.

One of the important areas of motion recognition applications is healthcare. Chen et al. [4] implemented a mobile phone-based system for multiple vital signs monitoring. The system can detect using a wireless accelerometer if a monitored patient falls and can alert care providers. In [15], accelerometers were applied to detect symptoms of Parkinson's disease.

Feature extraction techniques such as FFT, PCA, or ICA are rather computation-intensive and are better suited for desktop workstations. Mobile phones are weaker in terms of processing power. They also lack the constant supply of power and must rely on their limited battery resource. Thus we are looking for a method that could provide the best possible classification rate considering the special characteristics of mobile devices. To support continuous recording of motional activity, a realtime recognition system is sought.

3 System design

Our activity recognition system comprises the following components: wireless body sensors, a smartphone, and a desktop workstation. The in-house MotionBand [13] devices have been used for measuring sensory data. As Fig. 1 shows, they resemble a wristwatch and can be strapped conveniently to one's arm. A MotionBand contains an accelerometer, a magnetometer, and a gyroscope. There is also a button on the top which can be used to flag arbitrary events. It can be connected to other devices via wireless Bluetooth connections.

The smartphone is responsible for collecting data from the Motionband sensors. This is a natural approach since generally people carry their mobile phones on their person. A phone kept in one's pocket can continuously record motional activity.

However, supervised learning of activities could be a heavy burden for even powerful smartphones. Therefore we have separated the learning and recognition

Figure 1 The MotionBand device

processes. After collecting enough sensory data, the system is trained on a desktop workstation using feed-forward neural networks. Such networks are powerful at pattern recognition [3] and after training classification can be performed quickly. This enables us to implement a neural network on the phone that is fast enough to classify the activities in real-time. The parameters of this neural network are set to be equal to the trained neural network on the workstation.

The measurement method and the detailed description of the system are presented in the following sections.

4 Measurement method

The MotionBand sensors are triaxial. For each axis the accelerometer provides 16-bit data accurate to ± 6 G. The measured sensor value is affected by the gravity of the Earth. The magnetometer measures deflection from magnetic north and can be used while the device is in a still state. The gyroscope measures angular velocity, i.e. speed of rotation, within a range of $\pm 300°$/s. Sensory data is arranged into 28-byte packets containing additional bytes for synchronization and checksum purposes. Packets also include the status of the button. When connected wirelessly, approximately 50 packets are transmitted every second. Due to limitations of Bluetooth technology, at most seven MotionBands can connect at once to a single mobile phone or a computer.

MotionBands are designed to be comfortable and thin enough to be worn under clothes. A MotionBand is attached to a user's body with flexible straps. The weight of the device is 30 grams including the battery. The device has an internal battery with an approximate active operating time of 5 h and 20 h in idle mode. The battery can be recharged by a standard Nokia phone charger.

Using the three sensors it is possible to track both the orientation and movement of a body part. However, for activity recognition purposes, the accelerometer is the most valuable sensor, giving information about the forces describing the movements.

The following measurement method is based on the idea that complex activities of the human body can be described by their smaller components, i.e. the intensity of body part motions. Our goal was to capture this motional activity by deriving an appropriate intensity measure from raw accelerometer data. We were seeking for a measure that can be calculated and processed in real-time on a smartphone. The intensity of motion is directly proportional to the variation of acceleration. Building on this observation we defined the intensity value at time t as shown in Eq. 1:

$$Intensity(t) := \frac{1}{N}\sum_{i=0}^{N-1}\left|\frac{s(t-i)-s(t-i-1)}{\Delta x(t-i)}\right|, \tag{1}$$

where the measured acceleration value at time t is $s(t)$ and N is the number of samples. Since the input data is discrete, a simple numerical derivative is calculated to reflect jerkiness. The absolute values of the derivatives are summed and normalized by the number of samples. Features are extracted using a sliding window of two seconds across $N = 100$ samples. Since the 50 Hz sampling frequency of the MotionBand is approximately constant, the $\Delta x(t)$ sampling interval can also be treated as constant to simplify the calculations. In a window a single intensity value is calculated for each axis.

In our system, events interesting to the mobile phone user are recorded, e.g. when a new photo is taken or a phone call is made. Every captured event is augmented with metadata information describing the context. The current activity of the person is one such piece of contextual information. Since we need to provide the metadata instantly, the calculation of intensity values are done continuously, i.e. the window is sliding by one sample at a time. Note that this calculation can be executed quickly, as expressed by Eq. 1. When metadata is recorded, the current intensity values are classified to produce the activity state. In the following section the recognition method is presented.

5 Activity recognition

Three MotionBand devices were attached to test subjects to collect body-part intensity values. We wanted to differentiate between various activities in both standing and sitting positions, so attaching the devices to the dominant wrist, hip, and ankle seemed a promising choice. These sites are also suggested by research study [2].

The six activities to be recognized were resting, typing, gesticulating, walking, running, and cycling. Resting was defined as an activity with very low intensity values, such as sitting on a chair or lying down on a couch. For realistic recognition, brief movements such as stretching or changing posture were allowed. The task of processing e-mail on a computer was classified as typing since it involved typing on the keyboard and pointing with the mouse. Gesticulating involved vigorous hand gestures while standing still or walking slowly. The above tasks were measured in an office environment. For a natural setting, walking and running exercises were performed outdoors at various speeds, including turning to avoid obstacles. The cycling activity was recorded in a gym.

MotionBands were connected to a smartphone to capture data from the accelerometers. The benefit of this configuration is that it enables collecting data unobtrusively and is very mobile, i.e. the subject could freely engage in either outdoors or indoors activities. During recording, subjects used the button of the MotionBand on the wrist to mark the boundaries between the different activities. Using these markers and the noted sequence of activities, our software could partition and annotate the recording samples. Approximately one second of samples before and after the separators was cut out to ignore transitional movements between activities.

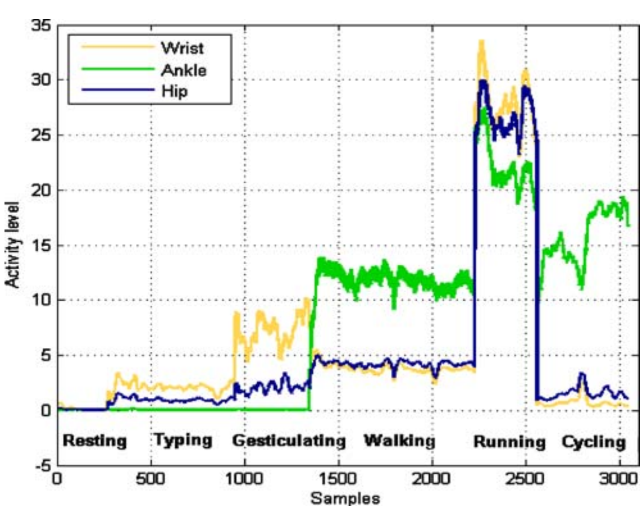

Figure 2 Sample intensity values

Figure 2 shows a sample stream of intensity values. The intensities are considerably different for the examined activities. Our recognition method is based upon this observation.

First, we considered using decision trees due to their simplicity. The C4.5 classifier of the WEKA toolkit was utilized for the learning phase. Once the training is finished, implementing the classifier corresponds to writing some nested *if–then* statements. Such task can be automated using a program code generator. The C4.5 algorithm performed well on the test data, but the recognition rate greatly degraded during live experiments. We suspect the problem comes from the overly precise intervals used for partitioning the classes which are fragile since input from the sensors always contains some noise. Another possible drawback of the classifier approach is that the output is always dichotomous: either the activity belongs to a class or not. In some cases it is useful to recognize partial or fuzzy membership of an activity across the classes. This can be used to resolve ambiguous cases by setting various threshold values or allowing an activity to belong to multiple classes.

After further experimentation, feed-forward neural networks were chosen as the tool for supervised learning of the activities. The learning phase was done offline on a desktop workstation using Matlab. A design choice had to be made between a single large neural network and multiple small networks. The large network could be trained to classify all activities, while the small ones could recognize one particular activity each. During classification each small network takes the current intensity values as input and calculates its output. The network claiming the highest confidence value is considered the winner.

Neural networks assigned a single classification task are believed to perform better than those with multiple tasks [5]. Therefore we chose the small networks architecture over the large one.

In theory, the training phase is computationally heavy but the recognition is fast. In practice, each perceptron classifier corresponds to an evaluation of an exponential function. However, most mobile phones do not have an FPU (floating-point unit), so they can only simulate real-valued arithmetic, increasing the processing time. Our goal was to find a network with the smallest number of perceptrons in the hidden layer that provides adequate performance. Performance was measured with F_1-*measure*, which considers precision and recall capabilities evenly. This scoring method is described in detail in Eq. 4 of the next section. The results are shown in Table 1. Networks with more than three hidden neurons did not provide significant increase in recognition performance.

Table 1 Increasing the number of hidden neurons

Activity \ Neurons	F_1-measure (%) 1	3	6	12	24
Resting	64.30	61.92	65.37	67.34	60.18
Typing	42.01	72.14	72.44	72.43	65.36
Gesticulating	87.22	81.46	81.64	85.25	87.10
Walking	84.58	92.35	96.02	97.88	97.99
Running	87.28	94.71	98.69	99.13	99.90
Cycling	69.73	75.96	73.68	71.11	70.89
Average	72.52	79.76	81.31	82.19	80.24

Based on this observation, the network consists of nine neurons in the input layer, fully-connected with the three neurons in the hidden layer and one neuron in the output layer. Six such networks were deployed, each responsible for recognizing one particular activity.

5.1 Classification performance

Training of activities was done using the Neural Network Toolbox in Matlab. The input set consisted of three complete sets of activities recorded by the authors of this paper. The average number of samples per activity was 1776, and all three sets had roughly the same size.

Networks were trained in multiple rounds using data from one subject at a time. For each network the training set was composed of all the corresponding positive inputs and an equal number of negative inputs chosen randomly from the other five activities, properly balancing the number of positive and negative training samples.

Before the training, the input patterns were normalized so that the occurring values were better defined. Matlab offers two methods for normalization. The *mapstd* function provides statistical normalization, i.e. the expected value and the distribution can be specified. The simpler *mapminmax* adjusts the input values to a given interval. We opted for the latter, since realtime data needs to be processed so that the minimum and maximum values can be bounded and difference in the input intensity sequence can be calculated.

The networks were trained with ten-fold cross-validation method using the Levenberg-Marquardt method as the backpropagation algorithm. This method is a very accurate learning algorithm [8], but due to its $O(n^2)$ memory and speed requirements it can only be used with relatively small networks.

The *tansig* function of the Neural Network Toolbox defined in Eq. 2 was utilized as the transfer function:

$$tansig(x) := \frac{2}{1+e^{-2x}} - 1. \tag{2}$$

This is mathematically equivalent with the hyperbolic tangent sigmoid function– *tanh* defined in Eq. 3 –, but can be computed faster.

$$tanh(x) := \frac{sinh(x)}{cosh(x)} = \frac{\frac{e^x-e^{-x}}{2}}{\frac{e^x+e^{-x}}{2}} = \frac{e^{2x}-1}{e^{2x}+1}. \tag{3}$$

Occasionally it may produce a slightly different numerical value due to the limited presentation of floating point numbers on computers.

The common MSE (Mean Squared Error) function was used to measure convergence and for adjusting the parameters of the network. In order to achieve a good output error rate, the learning factor of the network was lowered to $\mu = 10^{-6}$ and the target error rate was set to 10^{-3}. These changes resulted in a more accurate recognition without overfitting, because more epochs can be performed by the iterative batch learning algorithm, i.e. the whole training set was presented to the network more times, enabling a finer iteration.

To evaluate recognition performance, testing was done using all data (excluding transition periods) from the subjects. Utilizing the training data is justifiable since only a fraction of that had been presented to the network earlier.

The input training data was fed to our custom Matlab script. The script balanced the input patterns, implemented the ten-fold cross-validation, and executed the learning algorithm. After the training completed, the mobile recognition code was generated using the parameters of the trained networks.

Classification performance can be measured as accuracy, i.e. the proportion of the total number of predictions that are correct. However, accuracy may not be a suitable performance measure when the number of negative cases is much greater than the number of positive cases. In our case each network was responsible for classifying only one activity; only one sixth of the samples were considered as positive. If all the negative cases were classified correctly, but none of the positive cases, the accuracy would still be 83.33%. Therefore, *F-measure*, as defined in Eq. 4, may be a more appropriate evaluation method:

$$F_\beta = \frac{(\beta^2+1)\cdot P \cdot R}{\beta^2 \cdot P + R}, \tag{4}$$

where P (precision) is the proportion of correct positives among the predicted positives and R (recall) is the proportion of correct positives among all the positives. Precision can be interpreted as a measure of fidelity, whereas recall is a measure of completeness. The definition of F_β gives the weighted harmonic mean of precision and recall. We set $\beta = 1$ (F_1-measure) so that P and R are evenly weighted.

After the rounds of training and testing, F_1-measures for each activity were aggregated using arithmetic mean. These values can be seen in the first interior column of Table 2.

The average recognition rate was 79.76%. One of the objectives of this research was to study the effectiveness of non-individualized classification networks. Therefore we did not distinguish between the three slightly distinct personal characteristics. Naturally, alterations in the activities degraded the performance. During resting, gentle posture changes beyond resting peacefully caused the relatively low accuracy. Posture changes involving arm movements caused misclassification between resting and typing. Good classification results were achieved where the activity was nearly periodic in the sampling window: walking, running, and cycling. However, the cycling activity was performed differently by each of us: one kept a slow rhythm, one cycled at a fast pace, and one alternated between slow and fast intervals. This variation of intensity reduced the recognition rate.

Proper placement of MotionBands is important for repeatable capture. Small misalignment may have caused changes in the distribution of intensity along the axes. In order to reduce the sensitivity of the system, as an alternative approach, the intensity vector was replaced with a scalar intensity value, i.e. the sum of the triaxial intensities. This simplification made recognition independent of direction. The corresponding three-dimensional input set was directly derived from the nine-dimensional measurements. The same training and testing was performed and the results can be seen in the rightmost column of Table 2. Classification rates for typing, walking, running, and cycling improved; that for gesticulating fell slightly. Posture changes in resting reduced performance even more considerably than in the nine-dimensional case.

Table 2 Classification performance

Activity	F_1-measure (%)	
	9-dimensions	3-dimensions
Resting	61.92	54.29
Typing	72.14	80.28
Gesticulating	81.46	78.63
Walking	92.35	98.09
Running	94.71	99.74
Cycling	75.96	78.74
Average	79.76	81.63

5.2 Recognition on mobile phone

The activity classification system was built as part of our lifelogging research. The goal of the project is to capture personal memories by means of extensive recording. We have focused on data that can be acquired on mobile phones– including phone calls, text messages, photos, videos, etc. An important intention of this research is to enhance the retrievability of recorded events. Thus for each recorded event we store additional metadata, such as time, location, and physical activity of the user.

A prototype recording application was implemented for the Nokia 6630 mobile phone. The phone uses the Nokia Series 60 2nd Edition platform with the Symbian 8.0a operating system. The recording application was written using the Symbian C++ programming language. The activity classification was built as a module to this program. The phone has a multitasking operating system, which allows continuous recording in the background, during which the phone can be used normally for making calls or accessing other functions. The classification module has two main functions: it gathers input from the wireless motion sensors, and feeds this data to the neural classifier to infer the recognized activity.

There is one-way communication from the sensors to the mobile phone using the Bluetooth protocol. Symbian has a microkernel architecture, its networking subsystem runs as a server process in user-space. TCP socket connections are provided by the `SocketServer` process. The functionality of this process can be accessed using the `RSocket` class. The address of a Bluetooth device must be provided in order to establish a connection. This address can be found through the Bluetooth discovery process, in which the mobile phone queries the address of all the Bluetooth devices within the radio range. A dialog window was implemented from which the connection can be selected and established using the `RHostResolver` class. After the socket connection between a sensor and the phone is established, the sensor automatically starts transmitting current measurements. The data in bytes transmitted over a period of time is

$$B_{transmitted} = B_{packet} \cdot f \cdot t \cdot N, \tag{5}$$

where $B_{packet} = 28$ bytes is the packet size, $f = 50$ Hz is the packet sending frequency, t is the connection length in seconds, and N is the number of connected devices. In one minute, the three motion sensors generate $28 \cdot 60 \cdot 50 \cdot 3 = 252000$ bytes of data. Note that from each packet only the 16-bit accelerometer data is utilized and retained over the width of the sliding window used for the recognition.

The battery charge of the mobile phone is a limited resource. Mobile software draining the battery quickly could not be successful. Furthermore, the ARM mobile processor of the phone can only simulate floating-point operations, slowing down mathematical calculations. In order to optimize the use of the processor and battery resources, the complexity of the system had to be reduced as much as possible. This motivated us to employ small neural networks and intensity values based on the numerical derivatives.

The training of the networks was performed in Matlab on a powerful desktop workstation. The parameters of the networks, the weight matrices and bias vectors, represent the classification knowledge. Networks with the same topology were implemented for the mobile recognition system, inheriting the trained parameter values.

Recognition on the mobile phone works as follows. The input intensity values are fed into all six activity classifiers and their outputs are compared. Since tangent sigmoid is used as the transfer function of the networks, the output values range from -1 to 1. The output is accepted if the recognition is at least 95% confident, i.e. the output is greater than or equal to 0.9. If multiple outputs are above this threshold, the activity is ambiguous, and the network with the highest output is considered to recognize "monotasked" activity. More sophisticated schemes might allow multiple activities at the same time, e.g. gesticulating while walking. For our prototype we opted for a single best guess. A video demonstrating the setup and live recognition of activities is available online [6].

The Nokia Energy Profiler [18] tool was used to estimate the power consumption of the recognition task. The profiler runs only on newer Nokia S60 3rd Edition Feature Pack 1 devices, so the Bluetooth communication and calculation parts of our application had to be ported to the Nokia E90 phone which is suitable for measurements. The power consumption is measured as a whole and includes all the energy drawn by the software and hardware components of the phone. To estimate the power consumption of the Bluetooth communication, we first made measurements in the idle mode of the phone, then connected the MotionBands one by one, waiting about 30 seconds after connecting each device. Table 3 shows the average power consumption calculated as the difference between idling and when connected to successive devices.

One set of activity inferring mathematical calculations includes calculating the output of the six neural networks at 50 Hz over one second. However, the additional cost of the calculations was so negligible compared to the communication costs that they were statistically indistinguishable. In a separate measurement, we increased the number of equations by 200 times, which resulted in a power consumption of about 20 mW. Therefore, the average power consumption for one set of calculations is estimated to be 0.1 mW.

Table 3 Bluetooth average power consumption

Number of MotionBand devices	Bluetooth power consumption (mW)
1	95.89
2	129.41
3	157.72

6 Other areas of application

Some scenarios that could benefit from such activity recognition are presented in this section.

During recreational activities like running and cycling, and also walking and resting, listening to music is a pleasant pastime. In general, we would prefer harmonizing the music to the activity in which we are engaged, i.e. listening to quick, moving music during running but enjoying a slow, relaxing song while resting. Since current mobile phones also work like a portable music player, the realtime-sensed activity could be directly utilized to automatically select songs based upon their tempo. Therefore, during running, the user could enjoy music with a higher BPM (Beat Per Minute), and one with a lower BPM during walking. Further analysis could refine the classification. For example, sports activities comprising warm-up, exercise, and warm-down phases could be differentiated and matched with the play of matching songs. In a domestic environment, activity data could be sent wirelessly to the user's computer which can find songs consistent with the activity level and play them back through, e.g. home stereo equipment. During an initial phase, such a system might need a small learning phase with feedback from the listener actor to calibrate the motional intensity level to the preferred musical tempo.

Python is a high-level programming language available freely for a multitude of platforms. The version *Python for S60* can be used on newer Nokia smartphones. It offers an alternative to the Symbian C++ environment, whose complex memory management rules makes development difficult. *Python for S60* has become a popular tool for rapid prototyping of new mobile applications. Features previously only available in the native Symbian language can be now accessed through various Python APIs. NiiMe [1] is an open-source software written in Python that uses the built-in accelerometer in the mobile phone to wirelessly control the mouse of a computer by movements of the phone. The application allows using the phone as a steering wheel for car racing games, flying over landscapes, or even as small drum machine by executing different movements. Sensor streams and calculated intensity values from MotionBand devices could also be made available in Python to foster development of novel applications.

One such idea is to utilize the intensity values in a fitness program. Measured activity level corresponds to how much energy is exerted by muscles on which the sensors are placed. Therefore, combining such measurements with information about the person's height and weight could be used to estimate burnt calories during a session. Sessions could be automatically labelled with recognized activities, and visual charts could summarize how much energy is spent on each activity in the course of a day. Adding other contextual information could yield even finer statistics. For instance by taking the input of a GPS, we could record how many calories were expended during jogging in the park or walking to the office. Elevation data could also enhance such analysis. The software could recognize if a person would benefit from more exercises and, based on the gathered information, might suggest possible improvements.

One interesting application area of lifelogging is to automatically learn personal habits and try to provide more convenience for the user. Disturbance in the office could be reduced by "polite calling," i.e. redirecting phone calls to voicemail or other less obtrusive communication channels when the person's state is associated with being too busy. Sensing such high-level state is difficult, but the activity level is one of the salient inputs. Location and time information are also important. When combined with activity recognition, smart queries like *"Where do I go jogging more often?"* or *"What radio station was I listening to while going to work in the morning?"* can be formulated.

7 Conclusions and future work

We have presented the architecture and implementation of a realtime mobile activity recognition system. Considering the resource constraints of mobile phones, we have managed to create a system that can classify physical activities with acceptable confidence. In our experience the recognition was somewhat insensitive to the training subject's individual characteristics, and therefore can be used by other people without the need for re-training. However, for better performance, each user could input personal data to fine-tune the categorization by repeating the learning process on a workstation. This could be especially useful for a personal training application.

As a continuation of this work, other algorithms or input processing methods could be explored. Some algorithms presently suitable only for desktop workstations will soon be utilized on mobile devices. This trend is driven by the increasing processing capabilities of smartphones. For example, recent devices (Nokia N95, Nokia N810, etc.) based on the ARMv6 processor architecture have hardware FPU and SIMD (Single Instruction Multiple Data) coprocessors, facilitating more complex calculations. Adding other mobile sensor input including light, temperature, microphone, *etc.* could further enhance the recognition method, as suggested by the experiments in [17].

We emphasize the practical applicability of the activity classification system. By using lightweight and comfortable sensors and the user's mobile phone, activities can be monitored and recorded throughout a longer period of time. Our research project aims to record all kinds of information that can be gathered on the phone for later retrieval. Archiving the current physical activity seems to be a useful search key for browsing recorded events. The system is designed to be easily extensible with additional sensing and measurement devices. As sensors become easily wearable, e.g. woven into clothing, the smartphone can be thought of as an aggregator, processor, and transmitter of sensory data.

Acknowledgements The authors would like to thank Michael Cohen and the reviewers of this paper for providing valuable insights and suggestions for improvement.

References

1. Arranz A (2008) Niime (Nokia2MovingExperience). http://www.niime.com/
2. Bao L, Intille SS (2004) Activity recognition from user-annotated acceleration data. Lect Notes Comput Sci 3001: 1–17
3. Bishop CM (1999) Pattern recognition and feed-forward networks. In: Wilson RA, Keil FC (eds) The MIT encyclopedia of the cognitive sciences. MIT, Cambridge, pp 629–632
4. Chen W, Wei D, Ding S, Cohen M, Wang H, Tokinoya S, Takeda N (2005) A scalable mobile phone-based system for multiple vital signs monitoring and healthcare. Int J Perv Comput Commun 1(2):157–163

5. Cortez P (2004) MLP application guidelines. In: Summer school NN2004 neural networks in classification, regression and data mining, Oporto, July 2004
6. Fábián Á (2008) Activity recognition demonstration. http://uk.youtube.com/watch?v=qkKXboGsfxo
7. Gemmell J, Bell G, Lueder R (2006) MyLifeBits: a personal database for everything. Commun ACM 49(1):88–95
8. Hagan M, Menhaj M (1994) Training feedforward networks with the marquardt algorithm. IEEE Trans Neural Netw 5(6):989–993. doi:10.1109/72.329697
9. Kern N, Schiele B, Junker H, Lukowicz P, Tröster G.: Wearable sensing to annotate meeting recordings. Pers Ubiquitous Comput 7(5):263–274 (2003)
10. Kern N, Schiele B, Schmidt A (2003) Multi-sensor activity context detection for wearable computing. citeseer.ist.psu.edu/kern03multisensor.html
11. Kern N, Schiele B, Schmidt A (2007) Recognizing context for annotating a live life recording. Pers Ubiquitous Comput 11(4):251–263. doi:10.1007/s00779-006-0086-3
12. Lamming M, Flynn M (1994) Forget-me-not: intimate computing in support of human memory. In: Proc. of the friends, vol 21. Meguro Gajoen, 2–4 February 1994
13. Laurila K, Pylvänäinen T, Silanto S, Virolainen A (2005) Wireless motion bands. In: Position paper at UbiComp'05 Workshop on "Ubiquitous computing to support monitoring, measuring and motivating exercise". Tokyo, 11–14 September 2005
14. Lee SW, Mase, K (2002) Activity and location recognition using wearable sensors. IEEE Perv Comput 1(3):24–32
15. Mäntyjärvi J, Alahuhta P, Saarinen A (2004) Wearable sensing and disease monitoring in home environment. In: Workshop on ambient intelligence technologies for wellBeing at home. Held in conjunction with 2nd European symp. on ambient intelligence. EUSAI, Eindhoven, November 2004
16. Mäntyjärvi J, Himberg J, Seppanen T (2001) Recognizing human motion with multiple acceleration sensors. In: Proc. of the 2001 IEEE Int. Conf. on Systems, Man, and Cybernetics, vol 2. IEEE, Piscataway, pp 747–752
17. Maurer U, Smailagic A, Siewiorek DP, Deisher M (2006) Activity recognition and monitoring using multiple sensors on different body positions. In: Proc. of the int. workshop on wearable and implantable body sensor networks (BSN'06). IEEE computer society, Los Alamitos, pp 113–116. doi:http://doi.ieeecomputersociety.org/10.1109/BSN.2006.6
18. Nokia (2008) Nokia energy profiler. http://www.forum.nokia.com/info/sw.nokia.com/id/324866e9-0460-4fa4-ac53-01f0c392d40f/Nokia_Energy_Profiler.html
19. Randell C, Muller H (2000) Context awareness by analysing accelerometer data. In: Proc. of the fourth int. symp. on wearable computers, pp 175–176. Atlanta, 18–21 October 2000
20. Ravi N, Dandekar N, Mysore P, Littman ML (2005) Activity recognition from accelerometer data. In: Proc. of the seventeenth conf. on innovative applications of artificial intelligence (IAAI). AAAI, Menlo Park, pp 1541–1546
21. Vemuri S, Bender W (2004) Next-generation personal memory aids. BT Technol J 22(4):125–138
22. Yi JS, Choi YS, Jacko JA, Sears A (2005) Context awareness via a single device-attached accelerometer during mobile computing. In: MobileHCI: Proc. of the seventh int. conf. on human computer Interaction with Mobile Devices & Services, pp 303–306. ACM, New York. doi:10.1145/1085777.1085839

Norbert Győrbíró received his B.Sc. and M.Sc. degrees in Computer Science from the University of Szeged, Szeged, Hungary, in 1999 and 2003 respectively. He worked as a researcher and from 2006 as a senior researcher for Nokia Research Center in Budapest, Hungary, and in Helsinki, Finland between 1999 and 2007. He is currently pursuing Ph.D. studies in Computer Science at the University of Aizu, Aizu-Wakamatsu, Japan. Norbert is interested in mobile and distributed computing, fuzzy logic, and soft computing methods. He is a student member of the IEEE and ACM.

Ákos Fábián received his M.Sc. in Electrical Engineering from the Budapest University of Technology and Economics in 2007. As a graduate student working on his thesis, he participated in a mobile lifelogging project at Nokia Research Center. Ákos focused on the motion sensor input and implemented the mobile recognition software. Currently he is working as a software designer on the Telecom Server Platform for Ericsson Hungary Ltd.

Gergely Hományi works as a research engineer at Nokia Siemens Networks in Hungary. He received his M.Sc. degree in Informatics from the Budapest University of Technology and Economics. Previously, he worked at Nokia Research Center on various research projects. The spectrum of his research included planning and management of wireless mesh routers, location-based services, and applications of proximity networking (Bluetooth, ad-hoc WLAN). He contributed to the development of a mobile lifelogging application which provided the basis of the presented work. Recently, he has been involved in mobile media research, especially quality-of-service issues in a ubiquitous environment.