myHealthAssistant: A Phone-based Body Sensor Network that Captures the Wearer's Exercises throughout the Day

Christian Seeger*, Alejandro Buchmann
DVS Group, TU Darmstadt, Germany
{cseeger,buchmann}@dvs.tu-darmstadt.de

Kristof Van Laerhoven
ESS Group, TU Darmstadt, Germany
kristof@ess.tu-darmstadt.de

ABSTRACT

This paper presents a novel fitness and preventive health care system with a flexible and easy to deploy platform. By using embedded wearable sensors in combination with a smartphone as an aggregator, both daily activities as well as specific gym exercises and their counts are recognized and logged. The detection is achieved with minimal impact on the system's resources through the use of customized 3D inertial sensors embedded in fitness accessories with built-in pre-processing of the initial 100Hz data. It provides a flexible re-training of the classifiers on the phone which allows deploying the system swiftly. A set of evaluations shows a classification performance that is comparable to that of state of the art activity recognition, and that the whole setup is suitable for daily usage with minimal impact on the phone's resources.

1. INTRODUCTION

The World Health Organization predicts that chronic diseases will become the most expensive problem faced by current health care systems and sees the integration of prevention into health care as the main solution for this problem [14]. A paradigm shift towards integrated, preventive health care as well as equipping patients with information, motivation, and skills in prevention and self-management are described as essential elements for solving this problem. As body sensor network (BSN) systems are capable of continuously monitoring a person's physiological and physical state, they form a promising tool that equips users with the required information and motivation.

Many BSN-based projects in health care [5, 9] focus on monitoring of a particular disease or set of physiological signals. They benefit from the independence from stationary in-hospital observations, allowing patients to freely move and live their daily life while being monitored over longer times and under more realistic conditions. We focus on preventive health care and present a system that helps to reduce a person's physical inactivity which is one of the risk factors of many costly and disabling health conditions [14]. Studies [4, 12] have shown that an Internet and phone-based user motivation system can significantly increase and maintain this level of physical activity. Therefore, capturing a person's activities throughout the day is an important task of BSN-based preventive health care applications. The captured information is then used for motivating the person, could be shared with friends via social platforms or sent to a workout database which, in return, calculates a new workout plan based on completed workouts.

This work contributes to the field of BSN-based preventive health care applications. It performs daylong activity recognition and heart rate monitoring and adapts to given requirements on activity recognition. Pre-processing on the sensors saves the system's resources. In a base setup for daily activity monitoring, a set of a single customized accelerometer, a smartphone, and a heart rate sensor are used to detect five different activities, monitor the heart rate and calculate the calorie expenditure. When a person wears additional fitness accessories during a workout a more detailed activity recognition that gives precise workout information is provided. In this gym workout setup, two more customized accelerometers are added which allow detecting sixteen activities and counting of individual weight lifting exercises.

This paper is structured as follows: First, related work is presented. Afterwards, a system overview of *myHealthAssistant*, a preventive health care application that monitors both daily activities as well as very detailed gym exercises including repetition count is given. Both scenarios are presented and evaluated in Sections 4 and 5. The whole system's performance is finally discussed in Section 6, after which conclusions and a summary of our main results are made.

2. RELATED WORK

In [6] the authors propose an activity recognition system that utilizes phone-based accelerometers for detecting a user walking, jogging, climbing stairs, sitting, and standing. Labeled accelerometer data from 29 users were collected and 10-second intervals of training data used to induce a predictive activity recognition model. By implementing the activity recognition system on a cell phone, the daily habit of a huge amount of users can be collected. In that paper a real-time detection was not supported. The system presented in [3] supports real-time activity recognition. Three *MotionBand* sensors attached to a person's wrist, hip, and

*Supported by the German Research Foundation (DFG) within research training group 1362: Cooperative, Adaptive, and Responsive Monitoring in Mixed Mode Environments

BodyNets 2011, Beijing, China, 7–8 November 2011,
ISBN (978-1-936968-29-9).

ankle provide accelerometer, magnetometer, and gyroscope measurements to the user's phone via Bluetooth. By using feed-forward backpropagation neural networks the system distinguishes among six different activities, named resting, typing, gesticulating, walking, running, and cycling.

In [11] a combination of five accelerometers and one heart rate sensor is used. This combination allows not only recognizing fifteen exercises but also detecting the intensities of four of them. The real-time recognition is done on a laptop computer. Besides using the heart rate sensor for detecting the intensity of exercises, the authors also used the sensor for increasing the exercise recognition accuracy. Unfortunately, the resulting improvements were very low.

The authors of [1] present a comparison between two approaches for each, detecting and counting weightlifting exercises. For recognition they chose Naïve Bayes Classifiers and Hidden Markov Models and for counting they implemented a peak counting algorithm and the Viterbi algorithm with a Hidden Markov Model. An accelerometer glove and a posture clip serve as the data source for detecting nine weightlifting exercises. All calculations are done off-line. One outcome of this work is that the counting algorithm has to be adapted to different exercise speeds in order to improve its accuracy.

Compared to the related work, our approach provides day-long real-time activitiy recognition for different sets of sensor configurations. This allows detecting daily activities as well as specific gym exercises and repetition information.

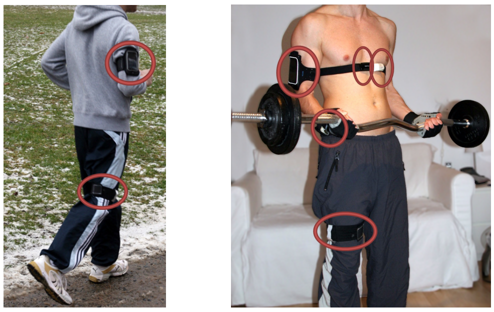
3. SYSTEM OVERVIEW

The application, *myHealthAssistant*, focuses on automated activity recognition and works on different granularities. For monitoring a person's daily activity, a coarse-grained activity recognition that detects only a few fitness-relevant activities is sufficient and does only require a small sensor network. For detecting all aspects of a gym workout, more precise activity recognition is necessary and additional information like the repetitions of weight lifting exercises is desired. This fine-grained activity detection needs a larger network of body sensors and increases the complexity of the system. The following fitness diary recognizes both the coarse-grained daily activities as well as the fine-grained gym exercises including additional repetition information. It stresses different aspects of a flexible body sensor network platform such as adaptability, seamless switching between sensor configurations, and multi-modal data processing. Figure 1 shows the sensor configurations of our case study. The calorie expenditure calculation shown in Figure 3 is based on a study from [15] using age, gender, weight and heart rate.

3.1 System Setup

3.1.1 Android Platform

Android is an open source operating system for smartphones and tablet PCs that uses a modified version of the Linux kernel (www.android.com). Software can be written in Java and executed in a specialized virtual machine. The number and functionality of Android devices grow rapidly and fit very well to the area of BSNs. A smartphone is unobtrusive and, hence, it can be used for daily (patient) monitoring whereas a tablet PC at the doctor's office can be used for better visualization of the patient's health parame-



(a) (b)

Figure 1: BSN consisting of a smartphone, a heart rate monitor, and a setup for daily activity recognition (a) and gym exercise detection (b).

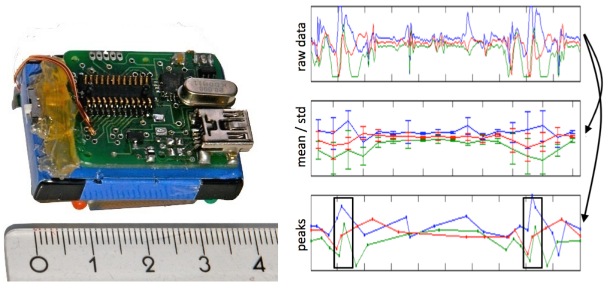


Figure 2: The inertial units were designed to operate as long as possible on a lightweight battery. A dedicated microcontroller calculates basic statistics and the peak features on-board before transmitting them wirelessly to the smartphone.

ters. Both devices are running the same system and allowing a seamless switching. The Motorola Milestone phone serves as an Android 2.1 device for our case study.

3.1.2 Sensors

Much of the early data processing in the proposed system is done as close as possible to the sensors. For the inertial sensor, a custom platform was implemented that samples the data from a 3D accelerometer (the ADXL330 from Analog Devices) at 100Hz and calculates per axis the mean and variance over a sliding one-second window, as well as characteristic peak features to enable a robust exercise counting (as shown later in Section 5.2). The microcontroller, a fast PIC 18F4550 from Microchip, operates at 48Mhz when calculating and is put to a low-power idle mode between samples being taken and processed. Figure 2 shows the uncased prototype of our wearable inertial sensor, with the battery wedged between the inertial sensing board and a ConnectBlue SPA 311 OEM module. On a small 360mA Li-Ion battery that is fully charged, the sensor can operate under the aforementioned conditions for over 50 hours. Recharging the module can be done over a standard USB port.

The Zephyr HxM Bluetooth sensor serves as our heart rate sensor (www.zephyr-technology.com/consumer-hxm). It monitors heart specific parameters including heart rate, calories burned, and R-R intervals as well as the wearer's step counts, speed, and distance. The sensor operates for 24 hours with a full charge.

Bluetooth is used for the communication among the sensors and the smartphone since it is well integrated in current smartphones and supported by most Android devices. In addition, there are already various Bluetooth-enabled (health care) sensors as consumer electronics products. The runtime of approximately 12 hours for the entire system, consisting of a wireless heart rate sensor, wireless custom-built accelerometers, and the Android smartphone, largely depends on the phone itself.

The sensor setup for daily activity detection consists of one accelerometer attached at the user's leg, a heart rate sensor, and a smartphone running *myHealthAssistant*. For a more detailed detection of the weight lifting exercises, including counting, two more accelerometers are needed: one in a weight lifting glove, and the other integrated in a chest-strap. All a person has to do is to switch on the sensors, wear them and the system connects to the newly available sensors and begins the fine-grained gym exercise detection.

3.2 Software Implementation

Our fitness diary application is built upon an event-based middleware we developed for BSNs [10]. The event-driven architecture inherently supports ad-hoc connections which is an important feature since BSN configurations change over time. Our case study for instance consists of two network configurations, one for daily activities and one for gym exercises. The event-driven architecture provides a seamless adapting from one to another configuration. Furthermore, having sensor- and application-specific modules as well as a layered structure in our middleware increases extendibility and adaptability. The bottom layer consists of sensor modules, the intermediate layer of an event handler, a database and application-specific modules, and the top layer consists of the user interface. The following describes the architecture in more detail.

3.2.1 Layered, Event-driven Architecture

Sensors connected to the Android phone are linked to sensor-specific modules at the bottom layer. Raw sensor data is sent to the corresponding module which translates the raw data to events. An event basically consists of an event ID, producer ID, timestamp, and sensor-related information. For instance, *HeartRateEvents* additionally consist of the current heart rate and *AccelerationEvents* consist of the mean acceleration values and variances per axis.

At the intermediate service layer, an *EventHandler* consumes both *HeartRateEvents* and *AccelerationEvents* which are forwarded to a *SQLite* database, the *ActivityRecognition* module and to the *PulseMonitor*. The *ActivityRecognition* consumes *AccelerationEvents* and produces *ActivityEvents* after performing the activity detection described in the next sections. Upon receiving a *HeartRateEvent*, the *PulseMonitor* performs a simple algorithm to check whether the current heart rate matches with the last series of *ActivityEvents*. If the current heart rate is above or below an activity-specific threshold, an alarm is triggered. A *CalorieExpenditure* module performs calorie calculations based on incoming *HeartRateEvents* and user specific parameters such as age, weight, and gender. Events produced by modules are always sent to the *EventHandler* which then forwards them to subscribed event consumers (modules). The *SQLite* database is used for logging.

An interesting property of our architecture is that modules can be started and stopped during runtime. Furthermore, sensors such as the leg sensor are re-used between detection modes without user interaction. Having another module running e.g., the gym exercise detection, means having just another event consumer. The *EventHandler* automatically forwards the leg's *AccelerationEvents* to the new consumer (e.g., gym exercise detection) without any impact on other event consumers (e.g., the daily activity detection).

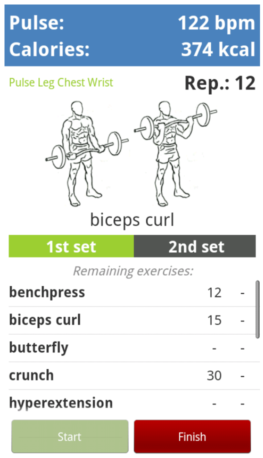


Figure 3: User interface of *myHealthAssistant* showing current heart rate, calorie expenditure, repetition count, exercise, and workout details.

Figure 3 shows our application's user interface including heart rate, calorie expenditure and gym workout monitoring. The current pulse and calorie expenditure are displayed on the top, followed by indicators for sensor connectivity on the left side and the current repetition count on the right side. A picture and the name of the current activity are sketched below this. On the lower half of the screen, workout information, such as finished sets, performed repetitions, and remaining exercises, is displayed.

The next two sections describe and evaluate the different granularities of fitness activity detection in detail.

4. DAILY ACTIVITIES

For monitoring a user's daily activity level, a simple distinction is made between being idle (e.g., sitting, standing), doing moderate movements (e.g., walking, cycling) and doing sports (e.g., running). The detected activity is then correlated with the user's current heart rate. If the heart rate does not fit to the current activity, an alarm is sent which provides a basic patient monitoring functionality. Showing the calorie expenditure already gives feedback about the level of physical activity.

4.1 Experiment Setup

For the activity recognition, we use a three-dimensional accelerometer attached above the right knee (cp. Figure 1 (a)). The sensor samples with a frequency of 100Hz and sends Bluetooth packets including the variances and mean values of the last 100 readings per axis every second. The low sending frequency was chosen in order to save energy. For the training data we collected for each walking, running, and cycling 120 data samples and for each sitting and standing 18 data samples from a male subject (age: 28 years,

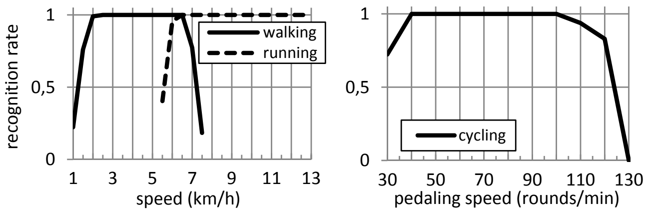


Figure 4: Stress-testing the daily activity recognition: Accuracy for different speeds, showing the regions where the activities are reliably detected.

height: 1.80m, weight: 67kg). Based on these samples of mean and variance values for each axis, we modeled the six-dimensional Gaussian distribution for each class and use this information for the activity detection directly on the phone. The closest distance to one of these classes of an incoming sample decides on the current activity. Based on this, we achieve the highly accurate recognition for the same subject as suggested by previous work [8, 7, 13].

4.2 Subject-dependent Evaluation

To evaluate the limitations of the proposed system, it is important to know in which ranges (e.g., speed) of a given activity the system still recognizes the correct activity and at which points it tends to fail. Therefore, we stress-tested our system with different walking and running speeds as well as different pedaling speeds for cycling. The tests were done with the same subject as for the training data. Figure 4 depicts the results for walking/running performed on a treadmill. The solid line shows the recognition accuracy for walking. For a very slow walking speed of 1 km/h the system does not detect walking properly since the movements are too slow. Only a recognition accuracy of 22% is achieved and in most cases the detected activity was "standing". By accelerating the walking to a more realistic speed the accuracy increases very fast (76% for 1.5km/h and 99% for 2.0km/h). In a range between 2.5km/h and 6.5km/h the activity "walking" was always detected correctly. With a very fast walking speed of 7km/h the accuracy dropped again. The dashed line shows the results for running. In this case, we started with 5.5km/h and an accuracy of 40%. At a speed of 6.0km/h the algorithm already achieved 96% of accuracy and reached 100% accuracy with a speed of 6.5km/h. We stopped the test at 13.0km/h with still 100% recognition accuracy. Figure 4 shows the results for cycling. The algorithm needs at least a pedaling speed of forty rounds per minute in order to reach a 100% accuracy and drops above a speed of one hundred rounds per minute.

Our tests have shown that the activity detection is robust even if the activities are performed in other speed ranges than they were trained for. In addition to indoor treadmill tests, the subject performed several outdoor runs as well, with similar recognition results: The figures show that the detection is accurate for realistic walking, running, or cycling speed. Since these tests were done by the same subject as for the training data, we will now test for subject-independency.

4.3 Subject-independent Evaluation

Ideally, an activity recognition system for preventive care or fitness applications is deployable without any additional training procedures. We therefore tested the performance of our system for subject-independency. The chosen subjects vary in age, height, weight and gender (cp. Table 1) and every subject differs at least in one factor from the person used for the training data (male, 28, 67kg, 1.80m). Every subject was doing the exercises in three different speeds (4 minutes each). They had to cycle with 70, 80, and 90 rounds per minute and to walk with 3.0, 4.0, and 5.0 km/h. For running they could choose their own three speed levels whereas levels between 7.0 and 10.0 km/h were chosen. Sitting and standing were not varied, as they were detected correctly.

Subject	Gender	Age	Weight	Height
1	female	52	70kg	1.65m
2	female	22	55kg	1.66m
3	female	22	72kg	1.66m
4	male	53	85kg	1.78m
5	male	28	82kg	1.70m
6	male	24	76kg	1.88m

Table 1: Details on the group of test subjects used in the evaluation. Extra care was taken to have a wide variety in especially age and fitness.

	walking	running	cycling	standing	sitting
walking	3208	1	1	0	0
running	0	3094	12	0	0
cycling	0	0	2938	0	0
standing	0	0	0	3120	0
sitting	0	0	1	0	3290
acc.	**100%**	**99.9%**	**99.5%**	**100%**	**100%**

Table 2: The confusion matrix and accuracies for subject-independent recognition.

Table 2 shows the results of the cross test. For all subjects, walking was detected with an accuracy of 100%, running had one outlier and cycling had 14 outliers with still more than 99% of detection accuracy. Those results show that the activity recognition is very reliable and does not need a person-specific training.

5. GYM EXERCISES

The previous section has shown that our application can handle coarse-grained activity recognition robustly on daily data, using a dedicated accelerometer and a heart rate monitor connected to a phone as a basic setup. This is sufficient for monitoring the user's overall daily activity behaviors, but more activities and specific information is desired for specific workouts: In this section we like to demonstrate that our architecture also copes with more complex BSNs at run-time, and that seamless BSN configuration changes are supported in the detection as well. A second workout mode is introduced for a more fine-grained workout diary. In this mode, a separate activity recognition module distinguishes between 16 activities: 5 cardio exercises and 11 weight lifting exercises shown in Table 3. In addition to this, more detailed activity information is given as a counting algorithm detects and counts single repetitions of each weight lifting exercise.

5.1 Exercise Recognition

For the detection of 16 gym workout exercises, the same Gaussian model-based classifier and platform as for the daily activity detection is used, with the only difference being that two more dedicated accelerometers are assumed to be present, to make it easier for the model to distinguish between the exercises: One sensor is attached to a sensor strap around the torso, and a second sensor is attached to the right weight lifting glove (cp. Figure 1 (b)). By embedding the

	Exercise	Posture	Type
1	Walking	-	Cardio
2	Running		
3	Cycling		
4	Rowing		
5	Elliptical trainer		
6	Wide grip lat pulldown	Sitting	Back
7	Barbell rear delt row	Standing	
8	Hyperextensions	Standing	
9	Barbell bench press	Lying	Chest
a	Butterfly	Sitting	
b	Front barbell raise	Standing	Shoulders
c	Dumbell lateral raise	Sitting	
d	Barbell curl	Standing	Arms
e	Cable triceps extensions		
f	Barbell squat	Standing	Legs
g	Table top crunch	Lying	Abs

Table 3: The set of 16 gym exercises consists of 5 cardio workouts and 11 weight lifting exercises.

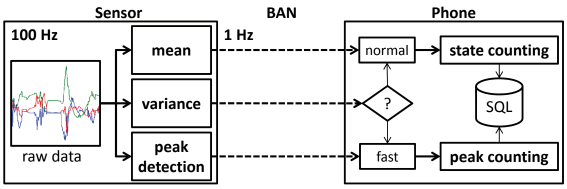


Figure 5: Data flow from raw data on the accelerometer to counting exercise repetitions on the phone.

sensors into the weight lifting outfit, the sensors are always located at the same position and no additional straps are needed. As before, every sensor transmits the mean and variance for each acceleration axis per second, expanding the total input data space for the Gaussian models from six to eighteen dimensions for exercise recognition.

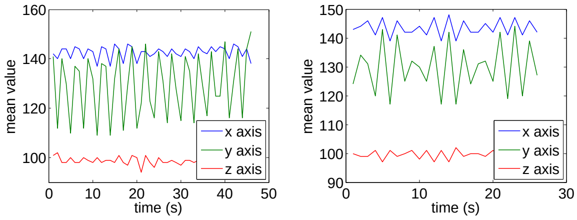
The exercise activities shown in Table 3 consist of 5 popular cardio workouts and 11 popular weight lifting exercises for training the chest, back, shoulders, arms, abs, and legs. The selection of exercises were chosen to provide a realistic and varied full-body workout set that regular gym users would perform. The execution of the exercises, especially their typical speed, is expected to be subject-dependent, so on-line training of the Gaussian models was implemented within the application. All results in this section will therefore also come from person-dependent evaluation.

The evaluation of the gym exercises detection was done in five runs in an actual gym environment. The extended BSN of smartphone, basic accelerometer sensor, and the two gym-specific sensors embedded in glove and torso-strap, were used for both the capturing of the training data, as well as the classification in real-time for the trained setup. Every exercise was performed for about 45 seconds and recognized amidst regular background data.

5.2 Exercise Counting

Previous work [1, 2] has identified the detection of weights and the counting of repetitions for weight lifting exercises as an important feature in a gym diary. For the amount of weights, the authors of [1] propose an RFID-weight mapping utilizing RFID tags on the weights and a glove with an RFID reader. The glove can send Bluetooth packets including the recognized tags to the phone which then could use the mapping for the weight calculations. In this paper, we are only focusing on the number of repetitions and leave the weight calculations as an interesting expansion of this project in the future.

For the counting of the exercises, a two-layer approach



(a) (b)

Figure 6: The 1Hz mean acceleration values from barbell curls in (a) normal workout speed and (b) very fast workout speed.

was followed, depending on the subject's speed of performing the exercises. This is a necessity since the 1Hz communication between inertial sensors and the smartphone might become too slow to count repetitions of faster exercises (such as those performed with lighter weights). We cope with this by detecting the workout speed, and switch to peak detection on the inertial sensor boards for counting the exercise repetitions. For normal workouts, however, the mean values that the inertial sensors are sending by default can be re-used for the counting.

5.2.1 *Exercise Counting on Phone*

Visual inspection (see Figure 6) of the wrist's mean values sent every second by the sensors indicates that straightforward autocorrelation on the variance-dominant axis is sufficient for calculating the number of repetitions for those workouts that were done slowly. Figure 6 (a) depicts for instance the mean values of the wrist sensor for barbell curls: The number of peaks in the y axis shown in the middle section of the plot matches exactly the 15 repetitions the test subject did. In particular, the exercise state counting module on the smartphone calculates the dominant axis, and measures through autocorrelation and variance on the axis with the most dominant variance the number of repeating states as soon as a new exercise has started, in real time.

Some exercises, however, can be executed in a higher tempo, which inadvertently leads to missed counts as is illustrated in Figure 6 (b). This has led to an implementation of a similar peak detection algorithm on the inertial sensors themselves, as explained in the next section.

5.2.2 *On-Sensor Peak Detection*

The exercises presented thus far were done in a usual workout speed, taking on average approximately three seconds per repetition depending on the exercise. In certain conditions, people tend to do their exercises in an accelerated pace, causing the system that was discussed in the previous section to miss counts. To remedy this issue, the inertial wireless sensor module is extended to preprocess not only the mean and variance per second, but also a basic peak detection, as depicted on the left side of Figure 5 and related to the technique presented in [1].

An alternative solution would be to increase the frequency at which the inertial sensors report mean and variance to the smartphone. As our data shows that the faster exercises tend to repeat themselves between 1.5 and 3 seconds, this would mean doubling or tripling the 1Hz sensor messages. This would however influence the power consumption of the whole system to deal with the increased communication speed, as well as cause a significant rise in data processing on the smartphone.

The sensor-based peak detection is done per axis and works as follows: A low pass filter of size five is applied on the last one hundred acceleration samples (which equals the one second time window) in order to filter out small variations that would cause tiny peaks to be reported. Then, peak detection is done on this filtered data by finding local maxima and minima over the last second. The two most pronounced peaks found per axis are then piggybacked on the packet carrying the mean and variance values of the wrist sensor. Thus, instead of being able to detect a repetition lasting at least 2 seconds by using the mean value, this process allows to detect recurring patterns over at least 1 second, using solely the most prominent peaks.

On the smartphone side, the peak counting module has to first decide whether one of these six values (two per axis) was significant for a finished repetition. For this, two parameters are important: 1) the dominant peak axis, and 2) a peak threshold characterizing a significant peak. In order to find these training parameters at runtime, a routine was designed that works completely unsupervised (i.e., the wearer just needs to provide the exercise at training, but without the counts): The dominant peak axis is characterized by the highest absolute sum of peaks outside the band of medians. In order to find the peak threshold, the peak samples from the dominant axis are clustered into two clusters using *kmeans*: From the typical speeds, one resulting cluster will describe the non-characteristic values close to zero, and the other cluster will describe values closer to the characteristic peaks. The value between the median of both clusters' codebook vectors is then defined as the threshold. As soon as both the dominant peak axis and its threshold per axis are found, the peak counting module is able to estimate the faster workout speeds as well.

5.3 Evaluation

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**a**	**b**	**c**	**d**	**e**	**f**	**g**
1	156															
2		210														
3			211													
4	41			176		42		42						42		
5	3				210				1						8	42
6						168										
7							210									
8								168								
9									206							
a				12						211						
b											210					
c												210				
d													210			
e				21										119	30	
f				1										6	163	
g	9															167

Table 4: Confusion matrix for recognition of the 16 gym-specific exercises (cfr Table 3). Overall precision and recall are on average 92% respectively 95%.

Table 4 displays the raw classification results in a matrix showing the inter-class confusion, for subject-dependent training. Classification accuracy ranges from 71% for the standing front barbell raise (labeled *e*), which is frequently confused with activities *4* and *f*, to an almost 100% accuracy for 9 of the activities. For this evaluation, false classifications at the start and end of the exercises were minimized by ignoring the first and last seconds per exercise (i.e., not classifying them as exercise class or background class). This method for discarding borderline data was also implicitly applied in the training process, where the system waits for 3 seconds after the user presses the training button on the smartphone's screen to start the collection of training data.

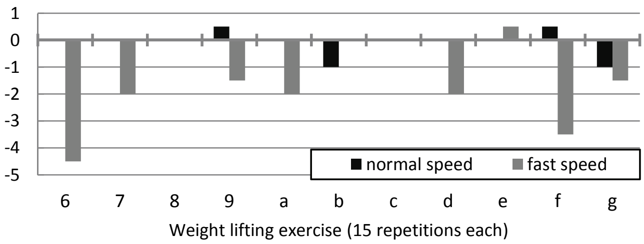


Figure 7: Average error in mis-counts for the counting algorithm, for both normal execution speed (in black) and fast ones (in gray).

The evaluation of the sensor-based peak detection was done in MATLAB, using the accelerometer sensors with embedded peak detection during a speed-up workout set from the same subject on another day. The result of this unsupervised algorithm is a miscount rate of 12.12%. We believe that this is still sufficient since a proper workout should be done more slowly.

Using the normal workout speed counting algorithm for a second training set resulted in four miscounts on the whole set of exercises and an overall miscount rate of 2.42%. The second training set was done by the same subject but on another day. Figure 7 shows the average results per exercise in case of normal and speed-up performance. The decision on which counting approach should be used for a given exercise is done in real-time based on the current variances.

6. SYSTEM EVALUATION

The entire system, with phone and sensors, lasts under realistic conditions (the phone being used frequently, all sensors turned on) for at least 12 hours of activity and heart rate monitoring without a recharge. This is enough time for monitoring a person during the day and charging the system at night. In future, this can be expected to improve, as the system is not limited to Bluetooth and more power-efficient protocols exist, to which our system can easily be adapted.

Figure 8 shows a day-long test of daily activity and heart rate monitoring. It also shows some unusual positive and negative peaks. The negative peaks were resulting from a too dry contact between the sensor strap and the skin which can be resolved by moistening the strap. For the high peaks, we have not found a proper solution so far. Usually the heart rate returned to the actual value after some seconds. In our application, the *PulseMonitor* compares the heart rate against an overall range of healthy heart rate values. Since the peak values in Figure 8 were outside these ranges, the *PulseMonitor* triggered a "dangerous heart rate" alarm and, hence, informed us about the sensor malfunction.

For gym exercise recognition, sensors have to be attached to three positions on a subject's body (cp. Figure 1 (b)). The heart rate sensor together with one accelerometer is attached to the chest. Since the heart rate sensor's strap is very comfortable and both sensors are small, these sensors are unobtrusive and attaching them is easy to do. The wrist sensor is combined with a weight lifting glove which makes it

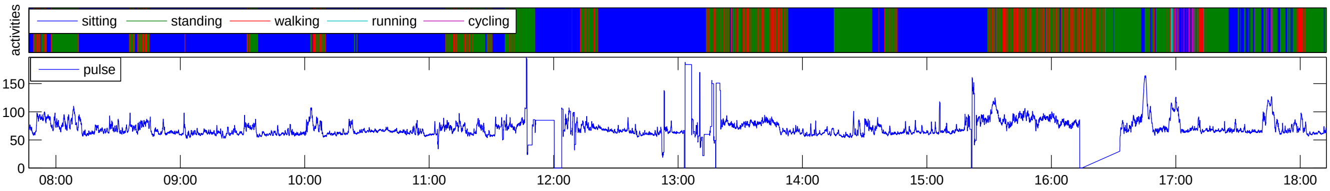


Figure 8: Visualization of the data captured by the BSN system on the phone: detected activities and heart rate values during one of the day-long tests.

also easy to attach. Only attaching the leg's accelerometer happened to be slightly difficult for subjects because it is not clear were the sensor has to be. This could be solved by labeling the strap with left/right indicators and showing a sketch illustrating an attached sensor. Once we had shown how to attach the leg's sensors, attaching it was no more a problem. Some subjects had the problem that the strap was slipping during the cardio exercises, especially during a run. For future tests we will try a dimpled rubber strap in order to avoid slipping. Overall, attaching the sensors was fairly easy and readjustments were not necessary.

Despite the peaks from the heart rate sensor, our system worked quite reliably. Only during the long-term tests, the leg's accelerometer disconnected once or twice. Until now the system simply informs us about the disconnection. For future work, we will implement an automated reconnecting. Except for the long-term test, there were no Bluetooth disconnections during our tests.

From an end-user's point of view, the system seemed to be easy and intuitive to use. Except for the leg's accelerometer, the sensors are unobtrusive and wearing the system in a gym did not attract attention.

7. CONCLUSIONS AND FUTURE WORK

Many (preventive) health care applications require continuous monitoring of patient's physiological and physical parameters. A body sensor network consists of body-worn sensors that allow monitoring a patient's parameters in real-time and therefore fits to those requirements. We presented a fitness diary that captures a person's heart rate, calorie expenditure, daily activities, as well as specific gym exercises. This preventive health care application intends to motivate patients to increase their level of physical activity and to decrease the risk of disabling health conditions.

The contribution of this work is a fitness diary that adapts to the given detection requirements. By wearing different sets of sensors at different occasions, the recognition system switches between detection of daily activities and specific gym exercises, and in addition to this counts the gym exercises' repetitions. Its underlying event-driven middleware supports a seamless switching between sensor configurations. The application's recognition performance matches that of state-of-the-art methods, while being capable of a reliable day-long activity and heart rate monitoring with real-time feedback to the user.

For future work, we will continue evaluating the gym exercise detection for extended use by expert users. We also plan to integrate uploading of the completed workout to a social platform in order to increase the user's motivation. In addition to this, the usage of activity information increases the precision of the calorie expenditure calculations.

8. REFERENCES

[1] K.-h. Chang, M. Chen, and J. Canny. Tracking free-weight exercises. *UbiComp*, 2007.
[2] R. Chaudhri, J. Lester, and G. Borriello. An RFID based system for monitoring free weight exercises. In *SenSys*, 2008.
[3] N. Gyrbíró, A. Fábián, and G. Hományi. An Activity Recognition System For Mobile Phones. *Mobile Networks and Applications*, Nov. 2008.
[4] R. Hurling, M. Catt, M. De Boni, W. B. Fairley, T. Hurst, P. Murray, A. Richardson, and S. J. Sodhi. Using Internet and Mobile Phone Technology to Deliver an Automated Physical Activity Program: Randomized Controlled Trial. *JMIR*, 2007.
[5] P. Khan, A. Hussain, and K. S. Kwak. Medical Applications of Wireless Body Area Networks. *International Journal of Digital Content Technology and its Applications*, 3(3):185–193, 2009.
[6] J. Kwapisz, G. Weiss, and S. Moore. Activity Recognition using Cell Phone Accelerometers. *Human Factors*, 2010.
[7] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford. A hybrid discriminative/generative approach for modeling human activities. In *IJCAI*, pages 766–772, 2005.
[8] K. Murao and T. Terada. A motion recognition method by constancy-decision. In *ISWC*, 2010.
[9] P. Neves, M. Stachyra, and J. Rodrigues. Application of wireless sensor networks to healthcare promotion. *JCOMSS*, 4(3):181–190, 2006.
[10] C. Seeger, A. Buchmann, and K. Van Laerhoven. An Event-based BSN Middleware that supports Seamless Switching between Sensor Configurations. In *ACM International Health Informatics Symposium*, 2012.
[11] E. Tapia, S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman. Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. *ISWC*, 2007.
[12] D. F. Tate, R. R. Wing, and R. A. Winett. Using Internet Technology to Deliver a Behavioral Weight Loss Program. *JAMA*, 2001.
[13] K. Van Laerhoven and H.-W. Gellersen. Spine versus porcupine: A study in distributed wearable activity recognition. In *ISWC*, pages 142–149, 2004.
[14] World Health Organization. Integrating prevention into health care. `http://www.who.int/mediacentre/factsheets/fs172/en/index.html`, 2011. [Online; accessed 01-April-2011].
[15] J. Yang and Z. Liu. ADACEM: Automatic daily activity and calorie expenditure monitor on mobile phones. In *SenSys*, 2010.