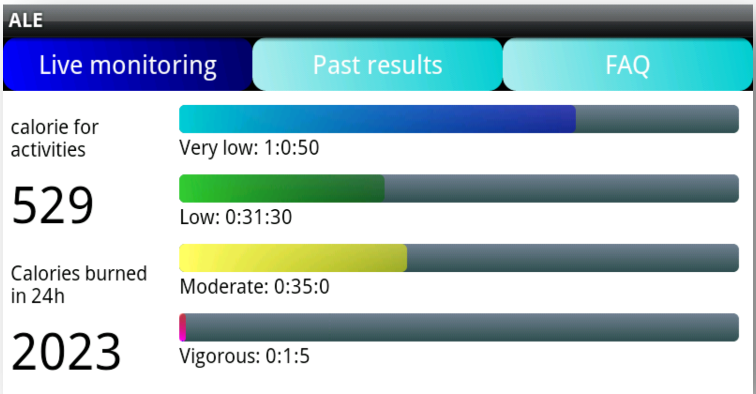
Activity Level Estimator

December 2010



Thesis for a Master of Science degree in Information Systems, performed at the Advanced Systems Group chair, Department of Information Systems, University of Geneva, Switzerland

Author: ir Jody Hausmann

Supervisors: Prof. D. Konstantas
Dr. Katarzyna Wac

Abstract

On a growing scale we use mobile phones for diverse activities in our daily life, for example for entertainment, education or information purposes. In our present research we assess the feasibility of using a mobile phone to track its user physical activity and estimate his energy expenditure. Activity Level Estimator (ALE) is an application developed for Android mobile phone. ALE analyzes and calculates how much time the user spends per activity level and gives estimation of energy expenditure. ALE is designed to be operational as a non-intrusive, background application and was tested with a set of users wearing their device in their pocket's pant. Via an extensive user tests, we assessed the accuracy of ALE against a dedicated BodyMedia Sensewear device. We conclude that ALE is accurate on average 86% for different levels of walking and it underestimates user's energy expenditure of 23% during a period of 24 hours.

Acknowledgments

Thanks to Katia who helped me for the English redaction and supported me during this thesis.
Thanks also to Mitch, Camille, Hélène and Nizar for taking the time to walk for science!
Finally, I want to thank Kate for her high level supervision and for her help during this research.

1. Introduction

This paper presents our research on Activity Level Estimator (ALE), a mobile phone application that detects and monitors user's activity level without any inputs from this user. We developed a system able to estimate energy expenditure along day including physical activity. With the mobile phone ALE transforms body movements into energy expenditure and displays it into kilocalories for the user. Kilocalories are estimation for physical activity made by the user during the day resulting in our estimation for a period of 24 hours. After evaluation, results for ALE showed accuracy with an average of 15% for activity like walking and an average of 23% for 24 hours.

We defined three criteria as hypothesis for our experimentation:

a. If we deploy ALE on mobile phone, it improves usability by being the least obtrusive to the users, by the way of an all-in-one device like a mobile phone and without any dedicated external device. It is possible to find external devices that detect physical activities, but most of them should be connected on a computer to compute results of the day.

b. If we deploy ALE on mobile phone, it will improve “compliance” that user will use it. Actually, most people wear a mobile phone during the day and the fact to have at proximity their mobile will improve the use of ALE. The trend of research shows that users wear mobile phone more time along years.

c. With the off-the-shelf mobile platforms we can develop ALE at least as accurate as an external dedicated system. It is important that ALE system is at least as accurate as external device built especially for this application.

In the section 2, we answer to the hypotheses a and b with the help of different existing scientific research results. We also present approaches made on the same topic. In the sections 3 and 4, we present ALE scenario and design, then the implementation of ALE in the section 5. In the section 6, we explain how ALE was tested and also its comparison with a dedicated device (to answer the hypothesis c). Finally, we give an overall discussion about our research and a conclusion in the sections 7 and 8.

2. State of the art

In Europe and Asia the ageing population increases. The generation of "baby boomers" will be retired soon. It is actually a big challenge for the Europe to take into account this generation and to determine the social and economical impacts (Europa Summaries of EU legislation 2008), especially on health. The need of assisted living at home will be a problem for countries because human and economical resources will become limited. For that, Europe launches many research programs to avoid this future issue. One of these programs is the development of a system that motivates users to be more physically active to avoid the need of assistance at home for this population. It is important to define what a physical activity is. Then how to measure it?

The next section defines physical activity and also explain metabolic equivalent task (MET), the unit to measure energy expenditure (section 2.1). To monitor physical activity, we need different technologies. Via researches, we found approaches that used portable devices. Some of them were complete external systems, with user tests and others showed technology approach (section 2.2). Then we explain the advantage of using an all in one device without any external system on daily basis (Section 2.3). Finally, we give a short introduction about ALE and our experimentation.

ALE is a mobile application that detects daily physical activity of the user and estimates the energy expenditure by the quantity of calories burned. It works with accelerometers provided by the mobile device and recorded movement of the owner when the mobile phone is in the pant's pocket.

2.1 Activity level estimation (ALE)

To estimate physical activity, we need to understand what it means. We need to quantify physical activity and give simple information that shows the level of physical activity of the user. We tried to understand what a physical activity is - in the medical terms - and how it is possible to measure and quantify it. Then we analyzed different solutions that are used by medical personal but also by end users. At the end, we made an accuracy's comparison of these methods to see which one was the most accurately evaluating the physical activity.

2.1.1 Physical activity definition and Metabolic Equivalent of Task (MET)

This section defines what a physical activity is - from a medical view - and also mathematical approach to evaluate it.

The medical definitions of physical activity:

1. *"Athletic, recreational or occupational activities that require physical skills and utilize strength, power, endurance, speed, flexibility, range of motion or agility"(Segen 2006)*

2. *"Bodily movement that is produced by the contraction of skeletal muscle and that substantially increases energy expenditure"(Bouchard, Blair et al. 2007)*

But we can also say that a daily walk from home to workplace is a physical activity. All body movements that need strength are a physical activity. Medical research created a methodology to measure a physical activity as energy expenditure and also gives a unit called *Metabolic Equivalent of Task* (MET) to quantify this energy. MET also simply called "Metabolic Equivalent" is a way to determine the energy cost of a physical activity. We used this unit to estimate the activity level of the user and also to calculate how much calories he burned during the day.

Definition of MET:
"The ratio of the work metabolic rate to the resting metabolic rate. One MET is defined as 1 kcal/kg/hour and is roughly equivalent to the energy cost of sitting quietly. A MET also is defined as oxygen uptake in ml/kg/min with one MET equal to the oxygen cost of sitting quietly, equivalent to 3.5 ml/kg/min."(Ainsworth 2002)

2.1.1.1 Resting Metabolic Rate (RMR) and Basal Metabolic Rate (BMR)

To explain the definitions above, we need also to understand what a Resting Metabolic Rate (RMR) and the Basal Metabolic Rate (BMR) are.

“BMR and RMR are estimates of how many calories you would burn if you were to do nothing but rest for 24 hours. They represent the minimum amount of energy required to keep your body functioning, including your heart beating, lungs breathing, and body temperature normal.”(caloriesperhour 2007)

A BMR is more complicated to measure, because “*measurements are typically taken in a darkened room upon waking after 8 hours of sleep; 12 hours of fasting to ensure that the digestive system is inactive; and with the subject resting in a reclining position*”.(caloriesperhour 2007) A RMR is easier to measure because the subject does not need to spend the night sleeping. Following equations were developed to estimate BMR and RMR without any clinical approach.

The Harris-Benedict equation for BMR (caloriesperhour 2007):

- BMR_men (kcal/day) = $(13.75 * w) + (5 * h) - (6.76 * a) + 66$
- BMR_women (kcal/day) = $(9.56 * w) + (1.85 * h) - (4.68 * a) + 655$

The Mufflin equation for RMR (caloriesperhour 2007):

- RMR_men (kcal/day) = $(10 * w) + (6.25 * h) - (5 * a) + 5$
- RMR_women (kcal/day) = $(10 * w) + (6.25 * h) - (5 * a) - 161$

Where:

- w = weight in kg
- h = height in cm
- a = age in years

These equations are only an estimation of BMR/RMR. We saw that many researches didn't made distinctions between RMR and BMR. A lot of time, the appellation RMR was used instead BMR even when the Harris-Benedict equation was used. We also made the choice in this report to talk only about RMR because it was the most used on scientific papers.

2.1.1.2 Activity levels and MET

MET classification was not developed to determine precise energy cost of physical activity but rather as an activity classification system (Ainsworth, Haskell et al. 2000). The MET level is defined as multiple of the standard resting energy values. When you are sitting, the standard MET value equal 1 (1 kcal-kg^{-1}-h^{-1}), an activity like walking at 5.6 km/h equal 3.8 MET (3.8 kcal-kg^{-1}-h^{-1}). In (Byrne, Hills et al. 2005), we found that the basal MET (1 MET) overestimates the energy expenditure by 20%. They conducted two studies, the first one with 593 subjects and the second one with 98 subjects. They measured the RMR of each person with different methodologies, like indirect calorimetry using a ventilated hood system and respiration chamber (see section 2.1.2) and also with different equations that predict RMR. Results showed an overestimation of the standard RMR used by MET classification. They proposed to adjust the MET level with the measured RMR with this equation:

$$MET\_adjusted = MET\ level * (1MET/\ RMR\ (kcal\text{-}kg^{-1}\text{-}h^{-1}))$$

To validate this equation, they measured the energy cost for an activity (walk at 5.6km/h) with medical techniques and compared with the estimate energy cost calculated with the equation (RMR was estimated with the Harris-Benedict equation). Results showed a difference of 0.2%.

With this adjusted equation, it is possible to calculate the energy expenditure during a physical activity in kcal. (Ainsworth, Haskell et al. 2000) listing more than 500 physical activity on a table with a corresponding MET values. The following Table 1 resumes the most frequent activities from the list of 500 elements.

Physical activity	**MET**
Light Intensity Activities	
Sleeping	0.9
Watching television	1.0
Reading, talking on telephone	1.3
Writing, desk work, typing	1.8
Walking, less than (3.2 km/h), level ground, strolling, very slow	2.0
Moderate Intensity Activities	
Walking downstairs	2.5
Bicycling, stationary, very light effort, walking (less 4.0 km/h)	3.0
Home exercise, light or moderate effort	3.5

Bicycling less 16 km/h, for leisure, walking at a brisk pace	4.0
Slow swimming	4.5
Walking at a very brisk pace	5.0
<u>Vigorous Intensity Activities</u>	
Slow jogging	6.0
Jogging	7.0
Calisthenics (e.g. pushups, situps, pullups, jumping jacks), heavy, vigorous effort.	8.0
Running (less 10 km/h)	10.0
Running (less 13 km/h)	13.5
Running (less 16 km/h)	16

Table 1 MET per kind of activities

2.1.1.3 Energy expenditure

METs on the previous Table 1 give the multiple energy expenditure per kg and per hour. With these values, it is possible to estimate in kcal the energy cost for an activity. For example, we want to estimate the energy cost for a men (31 years old, 67 kg, 178cm) during 30 minutes of jogging.

First we calculate the RMR (Harris-Benedict equation):

1. *(13.75 x 67) + (5 x 178) - (6.76 x 31) + 66 = RMR_men 1667.69 (kcal/day)*

Then, we adjust MET with RMR:

2. *1667.69 kcal/day = 1.037 kcal-kg*$^{-1}$*-h*$^{-1}$

3. Jogging MET values (on the previous table) = *7.0 kcal-kg*$^{-1}$*-h*$^{-1}$

4. *MET_adjusted = 7 * 1/1.037 = 6.75 kcal-kg*$^{-1}$*-h*$^{-1}$

Finally, we calculate in kcal the energy cost during the activity:

5. *6.75(MET_adjusted) * 0.5(30minutes) * 67(kg) = 226 kcal*

With this methodology, it is possible to calculate the total energy cost for a whole day by adding all kinds of activity.

2.1.1.4 Online tools

The web site (Health-calc 2006) proposes a tool to calculate the total energy expenditure of the day (Fig. 1). The tool asks different parameters, like gender, age, weight and height. Then it proposes different activities from intense exercise to sleeping. For each activity, you give time that you spend for. The tool calculates the RMR and also the total energy expenditure depending of your activities for this day. The result can be displayed in kJ or kcal. This kind of tools is very interesting to estimate how many calories you burn in 24h with taking into account activities that are easy to forget, like sitting or sleeping.

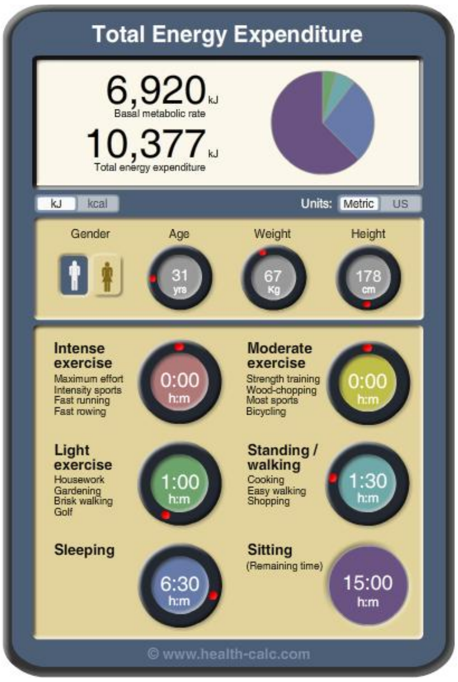


Fig. 1 screenshot of Total Energy Expenditure (Health-calc 2006)

2.1.2 Laboratory-based methods

As we saw in the previous section, MET is only estimation but determined after realistic tests in laboratory. The most common system to measure energy expenditure is the respiration chamber. It is a closed room with different sensors. *"Outside air is* *continuously drawn into the chamber and the flow rate of air at the outlet is measured using a pneumotachograph"* (Jequier and Schutz 1983). The paper cited above describes a room - built in the Institute of Physiology in Lausanne - of 5 m long, 2.5m wide and 2.5m high. It was a room with different elements for the user like toilet and bed. It was made for study energy expenditure of a human during 24h. The measure - from the air at the outlet - was filtered and analyzed with different techniques. The final result gave the RMR of the user in a very accuracy way. This chamber was also used - combined with indirect calorimetry - for the massive user test made by (Byrne, Hills et al. 2005).

Of course, this technique was not very accessible for many laboratories. The review of literature of (Jakicic) described techniques used also to measure energy expenditure:

- The Doubly Labeled Water (DLW) is considered as a standard system for the evaluation of energy expenditure without a huge system in a laboratory. It was also considered as one of the most accuracy systems. The user consumes a concentration of stable isotopes and hydrogen then the urine is measured over a 7 to 14 day period to determine the elimination rates of these two isotopes, from which carbon dioxide and the respiratory quotient can be estimated and energy expenditure determined. (Montoye, Kemper et al. 1996) The disadvantage of DLW is that it requires the use of water with stable isotope and it can be very expensive. The fact to ask the subject to collect his urine during a long period is also an issue for this experimentation.
- Indirect calorimetry (IC) (Fig. 2) is a very common system to measure energy expenditure. Definition: *"the measurement of the amount of heat generated in an oxidation reaction by determining the intake or consumption of oxygen or by measuring the amount of carbon dioxide or nitrogen released and translating these quantities into a heat equivalent."* (Mosby's Medical Dictionary 2009). It can be realized in a medical center but it is possible to find portable system. It is very expensive and the portable system is not able to measure energy expenditure more than a few hours. Results are accuracy and close than DLW

Fig. 2 On the left, a complete system for indirect calorimetry. On the right, same system but portable

2.1.3 Pedometer

Pedometers can be useful to estimate physical activity by counting steps and estimate walk and run speed. MET are determined by walk speed and the duration of the activity, but pedometer was also build for others reasons.

A pedometer is "*an instrument that gauges the approximate distance travelled on foot by registering the number of steps taken*" (Health-calc 2006)

Since last years, pedometers became very popular for personal wellness. This kind of device has a unique function, counting steps when you walk or run. But the fact to have the possibility to quantify the effort motivates a lot of people. Different studies show that pedometer's users change their daily behavior to walk more steps per day. (Tudor-Locke 2002; Tudor-Locke, Bassett et al. 2004; Sunny, Katherine et al. 2006; Anderson, Maitland et al. 2007)

Pedometer started to be popular in 1965 in Japan. At this time, Hatano started a program to help people start thinking about lifestyle and wellness and said that we need to walk 10000 steps per day (Hatano 1993). The slogan and the pedometer were well accepted by the public and became an international standard. For the public, this slogan was easier to understand and to remember than to know how much calories you burned per day. (Tudor-Locke and Bassett 2004)

There are many commercial pedometers available (Fig. 3) using different technologies. Some are mechanical and must be oriented; others detect steps on a 3D space (Fig. 3). Manufacturers add also software layer on device to give steps' count, but also different information like the quantity of calories burned. It was not possible for us to evaluate these software and specifically mathematical formula used because it was usually proprietary (Tudor-Locke 2002). We found same problems to understand and evaluate the accuracy of these devices. The only way to estimate the accuracy of those devices was to walk *n* steps and compare with the device result. In (Jakicic), we found that the accuracy was variable and depended on the speed of the user and also quality depended a lot of the prices. In (Schneider, Crouter et al. 2004), they compared the step values of 13 models and results showed that five models underestimated steps by 25% and three overestimated by 45%. Pedometer had also a module that detected energy expenditures in kilocalories, but the study (Crouter, Schneider et al. 2003) showed that pedometers (tested for the study) overestimate energy expenditure within 30% accuracy compared with indirect calorimetry. In conclusion, "*a pedometer can be used as a tracking device, a feedback tool (providing immediate information on activity level), and as an environmental cue (reminder to be active)*" (Tudor-Locke 2002).

Fig. 3 Commercial pedometers

2.1.4 Accelerometry

Accelerometry is a technique that uses uni-axial or try-axial accelerometer. It is possible to find many types of accelerometer, mechanical (old system) and electronical (MEMS, piezo-electric) (Fig. 4). Many researches tried to detect body movements with this kind of sensors (see section 2.2) and translated these movements in energy expenditure. In the literature review of (Jakicic), we found different studies that estimated the accuracy of these devices. Some of them showed an overestimation by an average of 9-13% and others with a higher percent of errors. Again with the paper (Jakicic), conclusion about accelerometry indicated that these devices were only accurate for specific activity like walking. Like pedometers, accuracy depended of models.

Fig. 4 Accelerometer device

2.1.5 Accuracy issue

Previous sections described different techniques to estimate the energy expenditure. We saw that laboratory systems were the most accurate but very expensive and inconvenient. About external device like accelerometer and pedometer, studies showed that accuracy depends on the devices. Tests showed that in many cases, pedometers and accelerometers overestimated the energy expenditure. Studies showed also that these kinds of external devices could be only used for specific activities like walking, running and lying. For the other activities like cycling, the accuracy falls to 50% (Jakicic). Accuracy of these portable devices like pedometers depended also on what is in use; medical or wellness. Few portable devices had medical validation and considered as accurate, the section 2.2.3 presents one of these device.

2.2 Existing systems for ALE

In this section, we described different papers that researched systems to estimate activity level with different mobile technologies. Most of these researches tried to encourage users to increase daily physical activity by first quantifying it and used different technologies to record steps or movements. Pedometers were used and users had often to put manually data directly into the mobile phone. Results from these papers were very similar and the fact to see daily physical activity encouraged a lot of potential users. A second functionality was often tested, the peer sharing of data with social interactions. This functionality created competitions state between participants and encouraged them to increase daily activity. We described also software approaches with techniques used to analyze recorded signals by accelerometers and how detect steps during a walk activity. We also analyzed an open source accelerometer and finally presented two accurate commercial devices.

2.2.1 Research deployments

Papers are described individually and split in different sections to classify different objectives, such: as

- Study goal
- Methodology for users' study
- Input / Output of algorithm
- Clinical or Wellness use
- Software / Hardware/ core algorithms
- If Peer sharing capability
- Results and comments

Shakra (Anderson, Maitland et al. 2007)

Goal
Shakra was a mobile phone application that tracked daily exercise activities of people with a mobile phone (Fig. 5). The system was able to detect different kind of activities, like walk, run or simply travel in a car. The application used an algorithm based on an Artificial Neural Network (AAN) to analyze the GSM cell signal and estimate the user's movement. The primary goal was to motivate users for daily exercise and tried to help them to reach the recommended level of activity for an adult, 30 minutes of moderate activity, five times per week. This research also tried two ways of motivating. The first way was to display a daily rate of activity to the user and the second way was to share information.

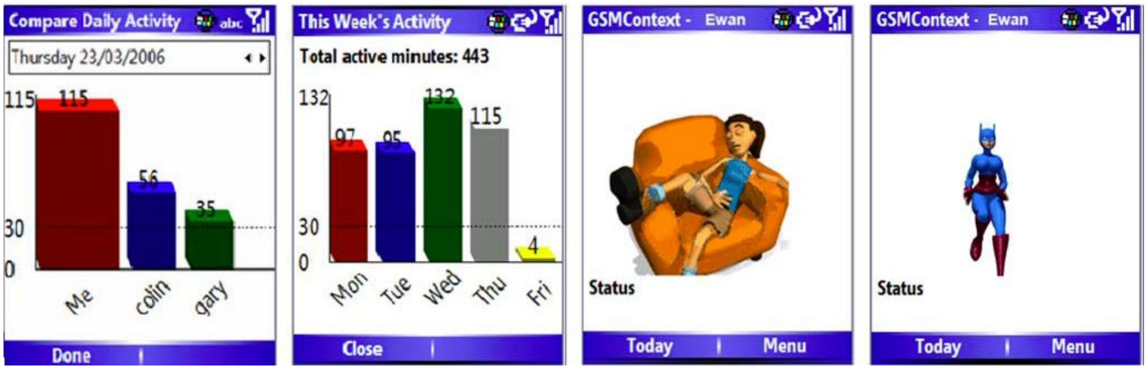


Fig. 5 Screenshots of the application

Methodology for users' study

To evaluate the application, three groups were studied during a week. Before the test, each participant used and tried the application for a 2 days training. These 2 days of training were useful to test the system and calibrated it if needed. Participants were groups of friends or coworkers. Each group was composed by users that had daily interaction. One group had 2 participants, the second three, and the third four. Each group was composed by people that had different level of activity (fair, moderate, high). Users were asked to wear the phone every day. After the trial, all data were analyzed, and all interaction between users as well.

Input / Output

Shakra can be used without any input from the user. The system detected the current activity of the user and every 30 minutes uploaded the data to the server. At the same time it downloaded data from other participants. The interface displayed different information to the user to compare the week's activity with a graph. It was also possible to compare the daily activity with another user. The user could also display the current activity.

Clinical or Wellness

Shakra was mostly used for wellness. The main goal was to motivate users to be more active on a daily basis.

Software / Hardware

The application was developed for windows mobile 5 and implemented in C#. The web service (that store data for the sharing functionality) was running in a Windows Server 2003 with a MySQL database. All mobile phones had a SIM card installed and used GPRS for the uploading / downloading. The physical activity was analyzed by the GSM cell signal.

Peer sharing

The peer sharing was one of the main functionality of Shakra. Data were automatically synchronized with all participants per groups.

Results / Comments

The user study was too short (10 days) to be reliable and to determine if the application had an impact for users, but results from users were similar. Users where very motivated and happy to use Shakra, especially the peer sharing functionality. Groups automatically created a kind of competition and where motivated to have better results among participants. The Result of individual interview returned that it was the major functionality for all users. Participants where very tolerant for the trial, because the AAN had issues and sometimes detected wrong activities. These problems' where detected and taken into account for results. Shakra was an interesting pilot, not for the technology, but for the kind of application. Users where motivated to be more active with the competition, but the user study was too short to be conclusive. There was no continued research on the platform.

Chick Clique (Connelly, Faber et al. 2006)

Goal

Chick Clique was a small mobile phone application and was designed for a small teenage friend's group. The main objective was to create a competition between users and compare daily physical activities. Participants used a pedometer all day and entered manually the result directly to the mobile phone. The result was uploading to a server and sharing to other participants. The final goal of this research was to test if persuasive technology could motivated user to be more active along the hypothesis that "Persuasive technology can be used to change people's behaviors in non-commercial domains such as preventative healthcare and fitness" (Fogg 2002). Persuasive technology can be defined by the use of advertisers like pop-up ads on web site.

Methodology for users' study

Four friends where engaged for the user study. Each participant used a cell phone and a pedometer. Every day, users recorded the quantity of steps directly into the application, then the system shared data and displayed some information, like if the daily goal was reached or not. The system displayed also positive feedback to the user for motivation. All users in the study were girls and teenagers.

Input / Output

Users recorded manually steps count from the pedometer into the mobile phone. Then, at specific time, the system uploaded data for sharing information. The user could see the daily results and also friends' results.

Clinical or Wellness

This research was most for wellness, because they tried some theoretical approach to help people to be more active.

Software / Hardware

The paper didn't give enough information about the software and hardware. For the pilot study, they used a cell phone and a pedometer.

Peer sharing

The peer sharing was the main objective of this research.

Results / Comments

The user study was not complete. We don’t know length of the test during, and also the results. We just know that users where motivated to share information with friends.

Houston (Sunny, Katherine et al. 2006)

Goal

Houston was a software for mobile phone used with an external pedometer. This software was a prototype to encourage people to be more active. The pedometer was used to count the steps and then users recorded information on the mobile. Finally, data were shared with other users (Fig. 6). The main goal of this research was to represent four design requirements for technologies that encourage physical activity. The paper presented also three versions of Houston:

- A basic version
- A personal version with daily goal
- A complete version with peer sharing functionality

Fig. 6 Screenshot of Houston

Methodology for users' study

The pilot study was composed by three groups of female friends (4 – 5 participants per groups) during three weeks. To recruit those people, they used a questionnaire (Sample Physical Activity Questionnaire to Determine Stage of Change (Division of Nutrition & Physical Activity 1999)) to classify them. This method classifies people into pre-contemplation, contemplation, preparation, action and maintenance stages based on stages through which people progress when modifying an addictive behavior (Prochaska, DiClemente et al. 1992). With this classification, they selected some users per categories. Only users classified in the precontemplation categories were not invited, because this kind of user had no intention to do physical activity for the next six month. Other people wanted to maintain, or increase physical activity. Participants were divided in three groups. For the first week, all three groups used the basic version of Houston, and then two groups used the sharing version and the third, the personal version during two weeks. Each participant was interviewed three times.

Input / Output

Participants recorded individually steps count every day. They could also record a daily goal. User could also see the results for the week, how many steps to reach the goal, and for the sharing version, results from another user.

Clinical or Wellness

Houston was most designed for wellness.

Software / Hardware
A pedometer and a mobile Phone. Huston was implemented in Python for the Nokia Series Platform.

Peer sharing
Only the third version of Houston had the peer sharing functionality.

Results / Comments
Results were interesting, because they found that users from group 1 and 2 (with peer sharing functionality) reached goals more often that the third group (without sharing). They also found that messages of encouragement were also very appreciated by participants and most of them tried to do more physical activity.

Actually, there are no recent researches on the same topic. Except for this paper (Anderson, Maitland et al. 2007) , we didn't find a all-in-one system. But we have to keep in mind that these researches were in the mobile phone emerging time and these devices were not built with many sensors as we can find now with smartphones. About results of these experimentations, all papers assumed that the period of the test was too short to be reliable. But results promoted to continue in this way. The peer-sharing functionality was a success for all these researches. The fact to create a competition between users could be a very good source of motivation.

2.2.2 Software systems

In previous sections, we defined methodologies - like indirect calories and respiration chamber - to estimate energy expenditure and research made to estimate physical activities. In this section, we focused on software approaches made for the accelerometry methodology. Then we described the functionality of an open-source pedometer.

The human movement is very complex to understand by a machine but recently, many scientific studies tried to monitor movements with different sensors. These studies are mostly medical research. Many clinics wanted to monitor physical activity for patients, by example, create a system that detected if an old person falls, or detect if the patients were physically active. To detect a human movement, studies used different sensors placed directly on the body. They used principally an accelerometer, a sensor able to detect acceleration on a direction. Actually, we can find this sensor in many cases, on a mobile phone, on a pedometer and also for video games, but detecting acceleration was not enough to evaluate the kind of movement. This section presented different scientific papers who described solutions that detected and evaluated movement.

2.2.2.1 Software comparison

The paper (Barralon, Vuillerme et al. 2006) talked about different approaches that detected in near real time if the user was walking or not. The papers described 4 different approaches and evaluated them with the help of 20 old participants (average of 79 years old). To compare these approaches, participants were asked to walk on the flat, sitting on different chairs, standing, lying, and picking objects at their own usual place indoor. All trials were recorded by a camera. They also defined that a "*walk phase is labeled when the subject perform two steps in a row*".

The Hardware was composed by an ACTIMOMETER (Barralon 2005) able to measure activity and mobility. This device was composed by 3 accelerometers (ADXL213, Analog Device) placed on the user's chest to detect anteroposterior acceleration, mediolateral acceleration and vertical down acceleration. The sampling frequency was fixed to 20Hz. The sensor was placed on chest and oriented orthogonally (x, y, z). The orientation was not very important because data from x, y, z values were normalized (dot product).

The four algorithms were developed to detect walking period:

1. Short Term Fourier Transform (STFT). This first approach analyses the movement in three times:
    a. Segmentation in temporal windows
    b. Frequency analysis (with equation STFT)
    c. Adaptative threshold

   At the end, the algorithm returned if the user was walking or not
2. Short Term Fourier Transform with Threshold (STFTT). Same approach as before but add a constant to reduce noise that could be detected as a walk movement.
3. Discrete Wavelet Transform. A method "*based on a ratio between the power of the detail signals and the total power in the anteroposterior direction*".
4. Continuous Wavelet Transform was a decomposition of the signal at different scales. The approach is the same as before.

These algorithms gave results with accuracy and indicated that only one accelerometer was enough to detect a walk movement. The best method using Discrete Wavelet Transform was the most efficient (78.5% in sensitivity and 67.6% in specificity)

The paper (Wang, Ambikairajah et al. 2007) talked about a method to evaluate and detect five different walking patterns. They used a tri-axial accelerometer attached at the waist. The different patterns analyzed:

- walking flat
- walking slope-up
- walking slope-down
- walking stairs-up
- walking stairs-down

Each pattern was recorded 10 times for all 52 participants. To record these data, they used a single accelerometer with a dynamic range of +/- 6 g. The sampling rate was 50Hz. They divided data in window of 128 samples (2.56 seconds) with half window length overlapping between consecutive windows. The sensor was attached at the waist above the iliac spine. Signals (X, Y, Z) were analyzed separately, that's why the sensors must stay always oriented at the same position on the user. They used a Wavelet Packet Decomposition as algorithm. This approach had results with accuracy (for each patterns) of 92.05%.

The paper (Foerster and Fahrenberg 2000) talked about researches on movement body. They recorded 31 participants for 13 motions and postures with 5 uni-axial sensors. To records data, they used a small computer carried by the user on belts. Sensors (IC Sensor Model 3031) were uni-axial, with a high sensivity and with a sampling rate at 32Hz filtered at 20 Hz. 3 of them were placed at the sternum and each was oriented as vertical, sagittal and lateral. Two were fixed on thighs (left and right) and sagittal direction oriented. All sensors were connected by cables to the small computer on belt.

They tested different configurations with 5, 3 and 2 sensors for each movement patterns. They found - with their approach - that 2 sensors was enough to found the four basic movement as sitting, standing, lying and moving. For the other movement patterns they needed to have more sensors. They also commented about the positioning sensors issues. Sensors must be well oriented and fixed to have the best accuracy.

The paper (Akay, Sekine et al. 2003) talked about the evaluation of effects of rehabilitation training for patient with Parkinson's disease and poststroke hemiplegic with "*portable acquisition system based on accelerometry*". They used matching pursuit algorithm to "*decompose signal into several already-known time-frequency patterns*". The acceleration signals were analyzed by a wavelet-based method. They used a tri-axial accelerometer (3031-010 IC-Sensors) placed on belt in the lumbosacral region and connected to a portable data logger.

The paper (Mathie, Celler et al. 2004) tried also a system which detected different movement, especially, the falls movement. They wanted to create a system as personal alarm for people and detected if the user falls. They developed an algorithm with decision node as a tree. The tree was binary. They collected data from 26 subjects. Each participants performed different movement. To monitor activity, they used a tri-axial accelerometer fixed at the waist. The algorithm was tested with these data. Data were classified on period of rest or activity. Then, algorithm tried to distinguished movement.

These researches showed how to create systems that recorded and detected physical activity, but approaches were too theoretical and not often applicable. Accelerometers were always fixed on unusual position and algorithms were not estimating physical activity in real time.

2.2.2.2 Commercial applications for mobile phone

About commercial software on mobile phones, we mainly focused on dedicated market for Android phone and iPhone. It was possible to find other applications for other platforms like Blackberry, Symbian or Microsoft Mobile. But actually, Android and iPhone are actually the most popular and provided also the most applications. On the Android Market, we found 23 pedometers available. For all of them, it was not possible to find accuracy information and also instructions about functionality of the step counter. The applications could be evaluated by the users with a rate scale from 1 to 5. On average these applications had a rate of 2.3 out of 5. Only three applications had a rate above 3 out of 5. In a big majority, these applications were free and the most expensive was sell at 2$. Only one of them was an open source application that we describe in the following section. We found two kinds of comments for these applications. The most current comment talked about the accuracy and a lot of users said that the applications did not count with enough accuracy steps, some of them said also that the application did not work. Another common comment was to about batteries, a lot of user said that these kind of application drained to much the battery. Few of these applications warned the users - in the application's description - about the battery.

It was not possible to count all step-counter available on the Apple Store because Apple divided the store per country and some applications were only available for specific countries. For the Switzerland's store in 2010, we found 57 applications. A lot of them were same but light version. Most of them gave description like that they were the most accurate available. A lot of time, a web site was given on the application description. We didn't find in these web sites scientific information about accuracy even how it was developed. We also found that some applications were based on trade mark system and we thought that it was only for marketing causes because we didn't find any information about these trademarks. Most of these applications were not free with an average of 1$. Regarding rating and comments, a lot of them had an average rate (from 1 to 5) of 3 and the majority had no comments from users. But we have to keep in mind that we only saw ratings and comments for the Switzerland's store.

2.2.2.3 An example of an open source pedometer

On the android market, we found only one open source pedometer application (Fig. 7). It was developed by Levente Bagi and was the first one to provide this kind of application for this Android platform. In this section we analyzed the algorithm used to count steps. We did not conduct deep study to evaluate the accuracy of this pedometer, but small tests showed that accuracy was pretty good. To make this possible we had to find the right parameters calibrated for each user. The application allowed to change the sensibility of the algorithm to be more sensible or not for detecting steps. References (Bagi 2009; Bagi 2010)

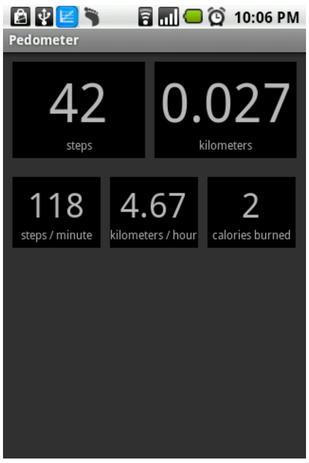


Fig. 7 Screenshot of the main GUI from the Pedometer application

We described here the algorithm in two different ways. In first we described it with pseudo code and then with a schema (Fig. 8) that described the life cycle.

This algorithm detected complete acceleration on a time window. When some condition was true, it calculated if the acceleration could be a step or not. Acceleration was defined with a start and an end when the signal started to increase until it decreases.

The algorithm used constant values

- `Limit = 30` → This value fixed a limit of sensibility (from 10 to 40). It was possible to change this value directly on the user interface. 30 was the default value and defined as a medium sensibility. The values represented the sensibility of the algorithm for detecting steps. That means that a step was count only if the acceleration analyzed was bigger than this limit. We didn't know how the author determined these values.
- `YOffset = 240` → We didn't find any information about this variable
- `Scale = 240 * (1 / STANDARD GRAVITY * 2)` → This values represented the constant of the earth gravity (g = 9.81 m/s) multiplied by the offset. The result was 12.244.

Pseudo code of the algorithm

1st step: Calculation with values from the raw accelerometer data (`X, Y, Z`)

VA^t was the average of each value (`X, Y, Z`) multiplied by the scale and the offset at each event from the sensor.

```
For (i = 0, i < 3, i++)
 VA^t += YOffset + values[i] * Scale
End For

VA^t  = VA^t / 3
```

2nd step: Detection of the acceleration

The algorithm determined the direction (increase of decrease) of acceleration with the last value record. There were 3 kinds of direction with result: -1; 0; 1.
-1 was a decrease direction when VA^t was smaller than the last value VA^t.
1 was an increase direction when VA^t was bigger than the last value VA^{t-1}.
0 was in case of equality between VA^t and last value VA^{t-1}.
The value direction gave a way of the acceleration.

```
if (VA^t > VA^t-1) then
 direction^t = 1
else
 if (VA^t < VA^t-1) then
   directiont^t = -1
 else
   direction^t = 0
```

```
 End if
End if
```

The next step was to detect if the direction was not the same as before. It meant if the direction was like the previous one ($direction^{t-1}$) an increase (-1) acceleration, the test was false. If the test was true, it meant that it was a new type of acceleration.

```
If (direction^t == - direction^(t-1)) then
```

In case of the test was true, the system needed to define if the change of direction detected was when the signal accelerate or decelerate. We called that `extremeMax` or `extremeMin`. The algorithm needed to find such extremity to estimated values at start and at the end of the acceleration. If the present value was at the end of acceleration, the system knew the values at the start.

```
If (direction^t > 0) then
extremeMax
Else
extremeMin
End if
```

3rd steps: Calculation of the difference between VA^{t} to determine if it was a step

The VA^{t-1} was saving on the array `lastExtreme`. This array provided two values, one per extremity (`extremeMax` and `extremeMin`).

```
LastExtremes[extremeMax] = VA^(t-1)
LastExtremes[extremeMin] = VA^(t-n)
```

or

```
LastExtremes[extremeMin] = VA^(t-1)
LastExtremes[extremeMax] = VA^(t-n)
```

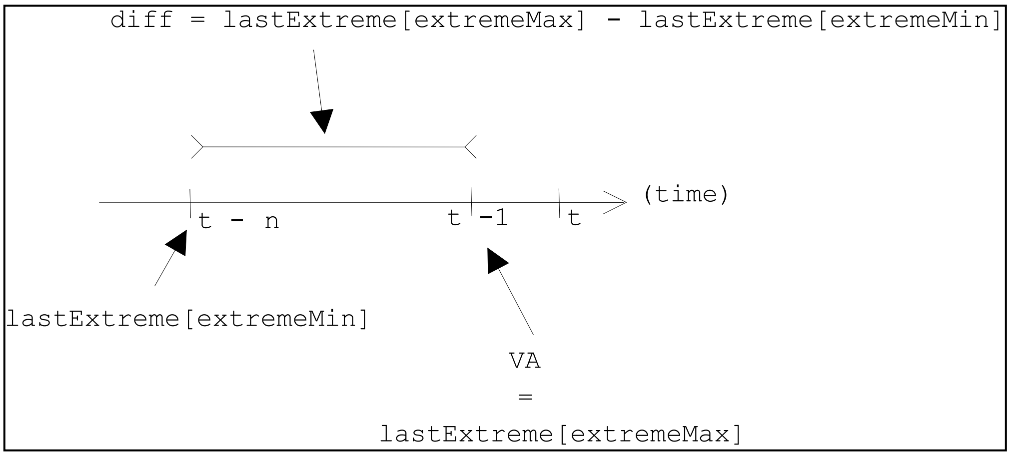
The variable `diff` was the difference between the VA^{t-1} (before the direction change, t - 1) and the VA^{t-n} (the previous direction change, t - n). That means that the code take into account a "window" of the last change of direction (t - n) with the new one (t). Then it calculated the difference with the VA^{t-n} and the VA^{t-1}

```
diff = absolute(LastExtremes[extremeMax] -
LastExtreme[extremeMin])
```

<u>Example</u>

Condition, the direction was opposite to the last one

Variable, `direction`^{t} `= 1`. (If `direction`^{t} `= 1` so `extremeMax = VA`$^{t-1}$)



There, the application tested if the value `diff` was bigger than the `limit`. The `limit` was the constant initialized in the beginning to determine the sensibility of the step's detector.

```
If (diff > limit) then
```

The next code was a filter of frequency, to being categorized as a step.

```
Boolean isAlmostAsLargePrevious =  diff>(LastDiff*2/3)
Boolean isPreviousLargeEnough = LastDiff > (diff/3)
Boolan isNotContra = extremeMax != LastExtremeMatch
```

or

```
Boolan isNotContra = extremeMin != LastExtremeMatch
```

If all previous tests were true, a step was count.

```
If (isAlmostAsLargePrevious == true AND
  isPreviousLargeEnough == true AND isNotContra ==
  true) then

    setStep +1
    LastExtremeMatch = extremeMin
```

or

```
    LastExtremeMatch = extremeMax
  Else
    LastExtremeMatch = extremeMin
```

or

```
  LastExtremeMatch = extremeMax
```

```
        End if
```

At the end, the algorithm recorded different values for the next turn.

```
LastDiff = diff (if diff > limit)
    direction^(t-1) = directiont
    VA^(t-1) = VA^t
```

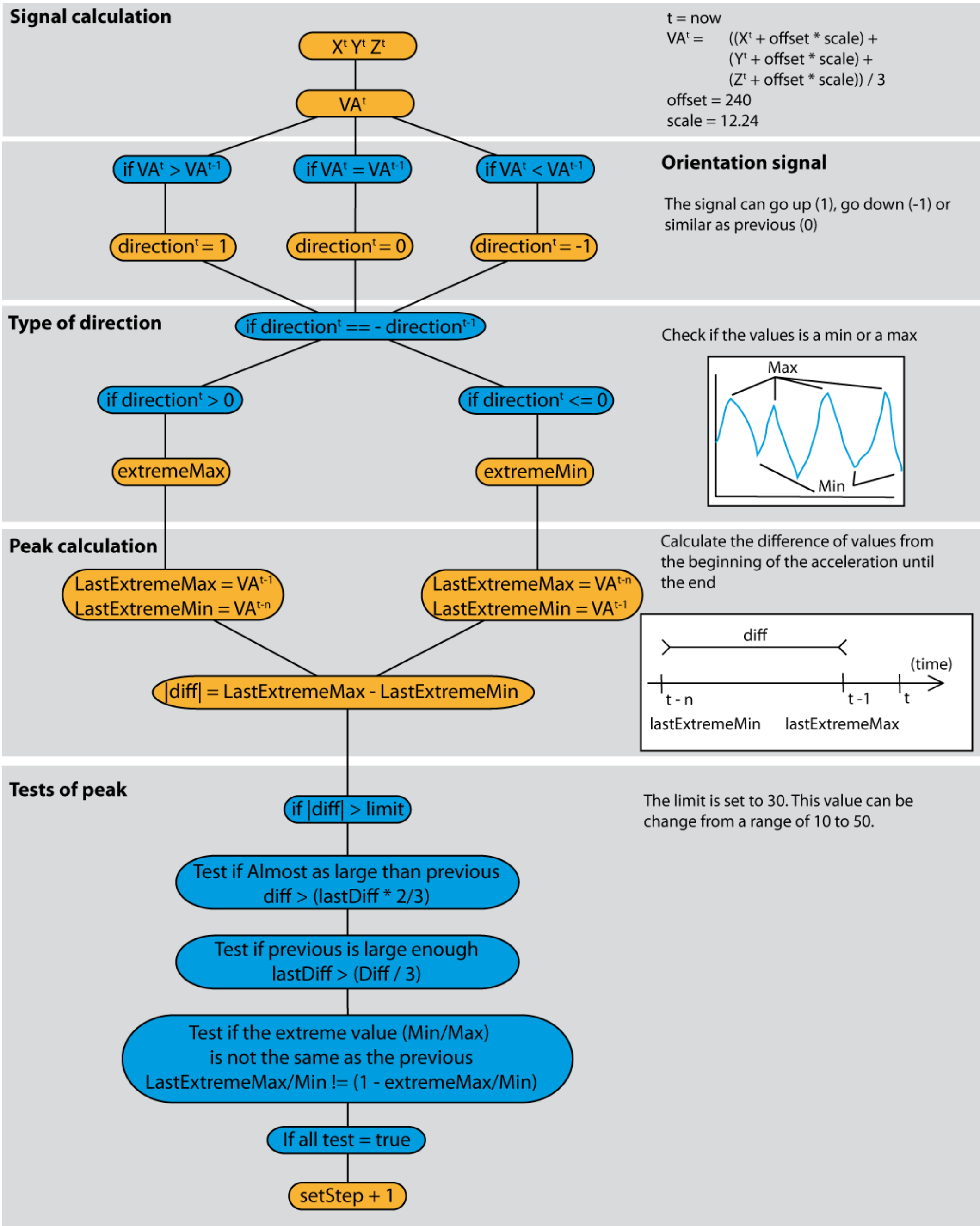


Fig. 8 Life cycle of the pedometer's algorithm by (Bagi 2009; Bagi 2010)

2.2.3 Dedicated hardware

SenseWear is a complete portable device for activity monitoring (Fig. 9) and manufactured by the company BodyMedia (SenseWear 2010). The device is on arm over the triceps muscle and has the capability of capturing the energy expenditure. It is composed by many sensors like: 3D accelerometers, galvanic skin response, skin temperature and heat flux. SenseWear measures the total energy expenditure (MET), physical activities levels, calories burned, steps and sleep efficiency. To see the results, the device must be connected (USB cable) on a computer and synchronized with the dedicated application which analyzes recorded data. On this application, the user can see how much time he spends per day on each level - sedentary, very low, low, moderate, and vigorous. He can also read how many calories he burned during the day.

SenseWear is a clinical product and validated for its accuracy. The medical professional can monitor their patient with the application described above, but with more information. It is considered very accurate by professional people. We found a comparison in (St-Onge, Mignault et al. 2007) between SenseWear and the Doubly Labeled Water methodology (see above section 2.1.2). The user study was made with 45 subjects over a 10 day periods. The results showed that SenseWear underestimated the daily energy expenditure of 117 kcal for a period of 24h than values measured with the DLW methodology. In (Jakicic, Marcus et al. 2004), we found also the same study but with the indirect calorimetry instead the DLW. Results were similar with an underestimation of 8.5%. For our experimentation, we made a comparison with our application and SenseWear (Section 6).

Fig. 9 SenseWear device

Fitbit (Fitbit 2010) is also a complete portable device that track the user energy expenditure. The device is very small and can be clipped on the belt (Fig. 10). It is more for a fitness use but as we can read on the web site, it is 95-97% accurate for the step counting. Fitbit has to be synchronized with a computer system to see the results. Providers elaborated a social network on Internet for sharing results with friends.

Fig. 10 Fitbit device

2.3 A mobile device choice as a platform

We decided to develop the ALE application on a mobile phone because actually, we found many of them that provided powerful computing, storage and communication systems and also different sensors. This kind of device was build with powerful processor and mainly of them had accelerometers. We made a comparison with these devices actually available on market, and then we explained why we selected the Android platform for our development. We tried after too explaining how mobile phones are daily used to explain how ALE could be daily used by users. Finally, we gave a short introduction of our application ALE.

2.3.1 Smartphone comparison

We can find many manufacturers of mobile phone at the market, but today the device is just a support for operating systems. The most common known OS are: Apple iOS (Apple Inc 2010), Google Android with Open Handset Alliance (Open Handset Alliance 2010), RIM Blackberry (Research in Motion Limited 2010), Nokia Symbian (Symbian Foundation 2010) and Microsoft Mobile (Microsoft 2010). These entire platforms are actually systems made for dedicated smartphones with many functionalities. Before 2007, only RIM, Nokia and Microsoft were on the market. At this time, smartphone was more designed for business use and was also not simply to use for the common user. These devices had Internet connection and also Bluetooth but were not equipped of sensors. Some of them had tactical screen used with a dedicated pen.

In 2007, Apple released his first iPhone. It was the first smartphone built with sensors and with multi touch screen (Apple Inc 2010). The release of this mobile phone changed totally the market over the world. Because it was a system very easy to use for every user and also because it was the first one with an original graphical interface that was not derived from computer. Apple made a successful operation and became quickly a very strong concurrent. At this time with others manufacturers, it was possible for developers to create applications, but it was never with a native code and frequently with java code behind a virtual machine. Moreover these applications were not concentrated on a specific market or web site. Users could not find easily applications. Apple with the iPhone introduced a new systems called Apple store. This functionality grouped all applications made by developers from everywhere and provided a complete commercial system for selling applications. Each time users bought an application directly from his iPhone, the developers earned money. For developers, this kind of functionality was a good way to be in front of the market and sell software without the help of big producer. Apple provided also for developer an SDK for developing applications. This SDK allowed creating many kind of applications.

On same time, Google created the Open Handset Alliance (Open Handset Alliance 2010), a group of many producer, manufacturer and operator. They developed the system Android. The first release was in 2009. Android was a very similar system like iPhone but based on a Linux system and open source. The difference was that Android was not only for a dedicated devices like iPhone but an operating systems for many different devices from different brands. The first version was touch screen (without multi touch) and now we can find many devices equipped with a lot of sensors and most are multi touch. The next section talked more in detail about the Android platform.

Microsoft Mobile (Microsoft 2010) has longer history than Apple or Google. They released the first version called pocket PC in 2000. It was more a small computer than a mobile phone. They had success mainly for business people. The version dedicated for new generation of smartphone was the Microsoft Mobile 6. It was not a success story but was available with many different devices. It was possible for developer to create applications, but most of them was only with Java mobile (a light version of Java) and under a virtual machine. The principal issues of these applications that it was not compatible for all devices. Microsoft also launched a market called Windows Marketplace for Mobile to allow users to buy the application directly from the mobile phone. This store was only available with the version 6.5 and later. This version was more accessible for developers because Microsoft provided for it an SDK with a native language. This year in 2010, Microsoft just released the new version called Windows Phone 7. This version was design to be a strong concurrent of Android and iPhone with new dedicated mobile - from HTC and Motorola. These devices were equipped of sensors like concurrent and proposed many current functionalities. Developer had also access to an SDK.

RIM - Research In Motion - released the first BlackBerry in 1999 (Research in Motion Limited 2010). It was the first smartphone on the market. During many years BlackBerry was a reference and the different versions of the device was always a best-seller. It was the first device to provide the functionality of email and agenda synchronized and became quickly the standard tools for business man. Actually, RIM is still a very strong concurrent especially in the USA where it is the first sellers. They also provided a market called BlackBerry Store and an API for developer.

It was not easy to do a comparison of all of these systems, but for that, we needed to find which one was the most accessible for research and also for our experimentation. We needed a mobile with accelerometer, with multitask system and also open to install easily homemade application (Table 2). Our choice was Android because it provides all functionalities we needed. iPhone was also a good choice but didn't provide multitask (excepted of the new version) and too close to install our application without agreement from Apple. Microsoft Mobile 7 was also a good choice but the release too late for our experimentation and the SDK 6.5 was not enough stable.

Mobile phone	**platform**	**Multitask**	**Development kit cost**	**Third party application installation**	**Store accessibility**
iPhone	close	no	90 euro	Only from iTunes with a developer account	Take times and constraint by many rules
Android	open	yes	free	From email, sd-card, usb cable	The application is publish instantly
Nokia (symbian)	open	yes	free	Sd-card, PC connect	N/A
Blackberry	open	N/A	free	N/A	N/A
Windows mobile 7	close	no	90 euro	Only from the Microsoft marketplace	Take times and constraint by many rules

Table 2 Development on mobile phone, a comparison

2.3.2 Android platform

The Open Handset Alliance was created by Google in 2007 (Open Handset Alliance). This consortium was composed by many industries like Google, HTC, Dell, Intel and many more. The main idea was to provide an open source system available for different devices without compatibility issues. For that they developed the Android system and released the first mobile in 2009 with the T-Mobile G1 made by HTC. Except for the Google API, all systems are open source. Users like developers can do whatever they want. It is also possible to provide homemade applications through the Market Store (free or not). The interesting thing is that developers don't need to wait for an agreement from Google to provide applications. It is also possible to install these applications directly from a web site or from a computer.

About the Android SDK, it is a Java language but works without virtual machine and is more complete than Java ME. Actually, we found different versions of the SDK, from 1.1 to 2.2. For our experimentation, we decided to use only the 1.6 version because it was the most widespread version when we started our experimentation. At the end of our research it was the 2.1. We select this SDK because it gave easily access to hardware layer and also a lot of functionalities for interaction with sensors. It gives also methods to create separately services on background and applications. For our experimentation, we needed to develop a system that runs on background, but also a user interface to interact with the application.

We selected also Android for this accessibility to share an application through the market store, but also the possibility to share it manually on many devices.

2.3.3 Mobile device usage

Our experimentation was mainly based on mobile phone because this kind of device became very popular these last years for many applications in daily life and not just for making voice calls. We had the assumption that users wear their mobile phone most of the time during the day and therefore it was for us an opportunity to propose a system to measure the daily energy expenditure based on a mobile device.

To support our hypothesis, we presented the proportion of users with their own mobile phone. In (Smith 2010), we found statistics about the percent of Americans adults that own a mobile device. They conducted survey with over 3000 adults. With this sample, they found that 85% of American adults own a cell phone in 2010. For comparison in Switzerland, we found that in 2008 on average 1.15 mobile phone subscriptions for every citizen (Fig. 11) (Office fédéral de la statistique 2010). We found also that 40% of Swiss people own one mobile phone and 50% own two or more mobile phones (Office fédéral de la statistique 2010). For an International comparison, the Fig. 12 presents the annual increase of mobile phone subscription.

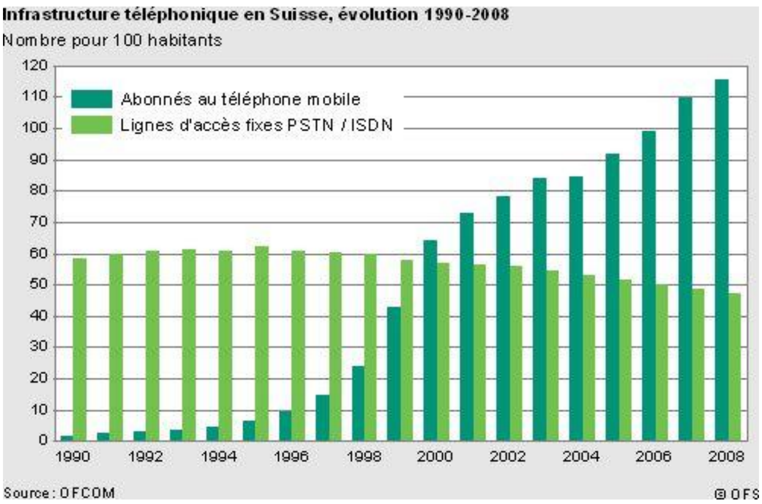


Fig. 11 The evolution (1990 - 2008) of subscription for standard phone (light green) and for mobile phone (dark green). Results are expressed for 100 Swiss citizen (Office fédéral de la statistique 2010)

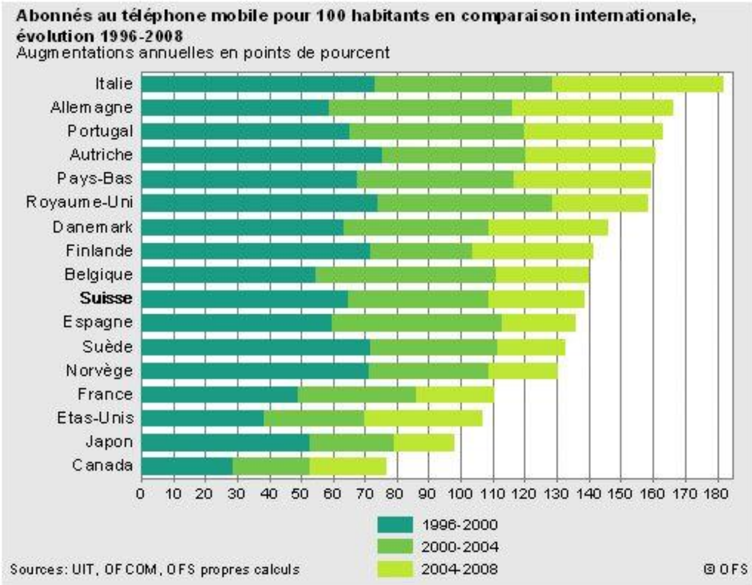


Fig. 12 The subscription increase in different country in percent of 100 citizen. (Office fédéral de la statistique 2010)

For smartphones, it was not possible to found recent statistics due to its novelty. The results for Switzerland didn't make the distinction between standard mobile phones and smartphones. It was possible to find data about subscriptions with Internet connection but only for 2008 and earlier. We knew that in 2010, the smartphone exploded on markets with the emergence of iPhone and Android phone - in 2009. So data for 2008 was not reliable to estimate the percent of users that used Internet with a mobile phone. Yet, for the USA, we found that - in 2010 - 29% of cell phone's users used and downloaded applications. A sample was 1917 users (Purcell, Entner et al. 2010); we assume they used a smartphone.

The paper (Patel, Kientz et al. 2006) described an empirical investigation of the proximity of users to their mobile phones. This study was conducted in 2006 but results were still reliable to explain how much time a user stays in proximity of his mobile phone. During three weeks, 16 participants wore a cell phone and a small Bluetooth device. The small device was connected to the mobile phone and transmitted a signal every 60 second to an application installed on the mobile phone. This application measured the strength of the signal received to estimate the distance between the Bluetooth device and the cell phone. The external device was very small and was worn a maximum of time by the user. Results showed that users had their mobile phone at a very close proximity i.e. at arm level during 68% of the time. It was mean that users had his mobile phone close to his arm during about 16 hours per day.

Unfortunately, we didn't find recent research that made the same or similar study. The trends showed that people used more mobile phone than in 2006 and showed also that the users spend more time with it and wore it longer in a pocket or in a bag.

2.3.4 Lack of external device

In the previous section, we saw that in a big majority, people own a mobile phone and wear it during a long period of time. As we also said, for our experimentation, it was a good opportunity to use this kind of device to measure physical activity of the user without any external device. Of course, it was possible to be connected with an external commercial device to measure steps and derivate data for measuring energy expenditure, but the fact was that users had to wear two devices during a long period of time and this kind of solution seems to be not a way of motivation for the user. Actually it is still an assumption that an all-in-one device was better for our research, except to have a long time study, we were not above to prove that, and therefore in our research, this hypothesis was not experimented. Experimentation cited in the section 2.2 showed that past experiments used always two separated device. The reason was simple, the technology was not yet available. We didn't find recent papers about the use of mobile phone and how this kind of all-in-one device could help research as ours.

2.4 Introduction to ALE and Thesis structure

Activity Level Estimator (ALE) was a prototype made for Android platform. It was developed to estimate continuously and in real time the energy expenditure along activity levels made by the users during 24h. It was designed to analyze signals from the accelerometer build-in to the mobile phone and evaluate them into thresholds that represented activity levels. ALE worked on background until the user selected to stop it. ALE was also built for working when worn in pants pockets, not on the hand's user and was not tested in a handy bag. In the Section 3, we present the case study of ALE and also the scenario that describes the possibility of its use. The Section 4 describes the whole design of ALE with description of its functional and non-functional requirements. Next the Section 5 talks about the implementation of ALE and also how the threshold algorithm was evaluated. ALE was also tested with different users and we made a comparison with a device providing ground truths i.e. with SenseWear device, we present results on the section 6.

3. Case Study

The following scenario describes an example of the utilization of ALE. We've chosen to tell a story about a senior user, as our experimentation was a part of a project to find tools to improve aged people's health. This scenario takes part in a close future. We assume that all mobile phones on the market are smartphone with different sensors, like accelerometers and compass.

3.1 Scenario

Jeanne goes into retirement this month. She was a teacher in primary school. She is married to Robert. Jeanne and Robert are 62 years old. They live in Geneva city center in a small flat. They don't need more space because their children have left home to live in their own flats, also situated in Geneva.

Jeanne is happy to have a lot of free time now, but she doesn't know what to do with it. She stays at home a lot and wants to be more active. Before her retirement, she was walking to and from work and after, she was resting at home with Robert. During her work's day, she was active because she walked a lot – when she taught – and stayed standing for a long period. Now she spends time at home and stays sitting almost all day long. She is affected by the lack of physical activities – mentally and physically – and wants to find a solution to change that.

A friend of Jeanne, Sylvie, retired two years ago, and she is very active. Sylvie talks to Jeanne and proposes her a new system called ALE based on a mobile phone. This system monitors the daily activities and is a good solution to see the effort you've done during the day. Sylvie presents also ALE as a motivating system. Jeanne is a bit afraid because she doesn't know how to use this kind of technology. She knows how to use a mobile phone (just enough to make a call), and how to navigate the Internet, but in many cases she has to ask for her children's help.

Sylvie shows her how to use this system and its options. She helps Jeanne to install, create an account and configure the system. Jeanne is really interested in this software and thinks, that if Sylvie can do it, so can she! She downloads ALE directly from the market store and waits for the automatic installation. Then, she launches it. The application asks directly some information about Jeanne like: gender; length; weight; age and estimation of daily sleep time. Sylvie explains that these data are useful to calculate how much calories the user burns per days. The explication is also displayed above the formularies. The ALE proposes to Jeanne to create a new account but as it is optional she skips the account setting and ALE goes directly to the live monitor section. She can see now two areas; the first one indicates an estimation of calories burned during the day. Sylvie explains that it is an estimation of 24 hours without any activities – only sleeping and sitting. Sylvie teaches also that if Jeanne goes for a walk – during 30 minutes – the estimate calories will be automatically updated. The second area presents a chart with 4 bars of different colors. Each bar represents an activity level: very low; low; moderate and vigorous. For each bar, the category's name and activity's time are displayed. Sylvie explains that the bars grow depending on the activity. As she said before, if Jeanne goes for a walk at a normal speed, the low level bar will grow. Jeanne asks also if it is possible to compare with another day and see if she walks more the than previous day. Sylvie shows to Jeanne the history tab at the top of the interface. Jeanne clicks on it and sees only small bars with different colors. Next to this bar, she notices an empty section. Sylvie explains that it is normal because there is no more data at the moment to display and after using the application during a week, she will see a bar for each week's day. Four small color sections represent the bar of the day. Normally each section represents an activity but at this time, each section is at 0. Sylvie discovers also that she can change the date to see past results. Jeanne asks how Sylvie knows all these functionalities. Sylvie explains that she reads – in the third tab – the FAQ and the help section and that the information was sufficient to understand the basics functionalities. Then she had understood even more after she has used ALE a bit.

The day after Jeanne tries ALE. She decides to see what the system means by very low, low and moderate. For that, she puts the mobile phone in her jeans' pocket and starts to walk slowly in her flat. Then she accelerates more and more until a very fast walk. She walks about 5 minutes with different speeds. At the end, she takes her mobile phone and sees that the application was still open and displayed the four bars. She also sees that three of them have grown. She mentally computes the total time for each bar and found that the result is about 5 minutes. Then she restarts to walk but this time after each different walk, she looks on the mobile phone to see which bar has grown. Jeanne thinks it is a fun application and decides to wait for the end of the day to see how active she was today. She clicks the "Home" button of her mobile phone and puts it in her pocket. At the end of the day, Jeanne takes her phone and sees above the screen an icon that indicates that ALE is still running. She clicks on the icon's application directly on the menu and ALE displayed the live monitoring page. Jeanne sees that her results are poor but she remembers that she didn't move a lot during the day. Then Jeanne clicks on the menu button and selects to quit the application. Now she sees that ALE stopped running.

After two weeks, Jeanne meets Sylvie again. Sylvie asks her how it is going with ALE. Jeanne answers that in the beginning, she was a bit frustrated by the poor results. She thought that she was more active during the day and was very surprised by the low results of her physical activity. She thought that even if she didn’t go out for a walk; she walked a lot at home. Now, she starts to go out every day and walks each time more. Now she fixes herself a goal to walk at least 2 hours per day with a moderate walk. She likes to compare results with previous days because it is a good source of motivation for her.

Jeanne remembers that she skipped the account creation step when she installed ALE and asks what is interesting given this function. Sylvie explains that it is a web page account that displays personal results. At different moment of the day, ALE sends data to the server. It is interesting because if she changes or loses her mobile phone, data are still on the website. Sylvie adds also that a message displayed on the web site informs that new feature will coming soon like the possibility to share results with another users and many more. Jeanne thinks it will be fun to share results with Sylvie because it feels like a challenge to her now.

3.2 Use case

From the previous scenario, we define the next use cases:

List of functionality

End user:
1. Start the application
2. Set the user personal information
3. View live results
4. View history results
5. Search for a result by date
6. Put in background
7. Quit the application

ALEservice:
8. Start a new period of monitoring
9. Analyze signal
10. Save results on the database

1. Start the application	
Goal	The End-user launches ALE on his device
Derived from	
Primary actor	End-user
Preconditions	ALE installed
Actors	End-User, ALEgui, ALEservice
Success	ALEgui and ALEservice started
Failure	Issues from the device
Trigger	The End-user wants to use ALE
Description	1. End-user clicks on ALE icon 2. ALEgui start 3. ALEgui start ALEservice
Extensions	3.1 ALEservice is already active, ALEgui bind ALEservice

2. Set personal information	
Goal	The End-users setup his personal information
Derived from	
Primary actor	End-user
Preconditions	ALE started
Actors	End-User, ALEgui
Success	Personal information saved
Failure	
Trigger	The End-user enters his personal information
Description	1. End-user clicks on the ALEgui menu 2. End-user clicks on the Setting button from the menu 3. ALEgui displays the settings page 4. End-user enter his personal information 4.1 End-user gives his gender 4.1.1 ALEgui saves data 4.2 End-user gives his age 4.2.1 ALEgui saves data 4.3 End-user gives his weight 4.3.1 ALEgui saves data 4.4 End-user gives his height 4.4.1 ALEgui saves data
Extensions	

3. View live result	
Goal	The End-user sees his result of the day
Derived from	
Primary actor	End-user
Preconditions	ALEgui and ALEservice started
Actors	End-User, ALEgui, ALEservice
Success	ALEgui displays results for the current day (calories for activity, calories for 24h, time per activity levels)
Failure	
Trigger	The End-user wants to see his result of the day
Description	1. End-user clicks on the live result's tab

	2. ALEgui waits for the next update data from ALEservice 3. ALEgui redraws results
Extensions	

4. View history result	
Goal	The End-user looks for history results
Derived from	
Primary actor	End-user
Preconditions	ALEgui started
Actors	End-User, ALEgui
Success	Week results are display
Failure	No results on database
Trigger	The End-user wants to see past results from the week
Description	1. End-user clicks on the history result's tab 2. ALEgui collect data from the database 3. ALEgui displays results for the last 7 days
Extensions	3.1 If they're less than 7 day results, ALEgui blank information for empty days.

5. Search for a result by a date	
Goal	The End-user want to find results at a specific date
Derived from	
Primary actor	End-user
Preconditions	ALEgui started
Actors	End-User, ALEgui
Success	Result at the selected date is display
Failure	No result at this date
Trigger	The End-user want to see a result at a specific date
Description	1. End-user clicks on the history result's tab 2. ALEgui collects data from the database 3. ALEgui displays results for the last 7 days 4. End-user clicks on the search by date button 5. ALEgui displays a calendar 6. End-user selects a date

	7. ALEgui collects data for this date from the database 8. ALEgui displays result
Extensions	8.1 There is no information at this date, ALEgui informs the user with a message

6. Put in background	
Goal	The End-user wants to quit the application but wants to keep ALEservice still active in background
Derived from	
Primary actor	End-user
Preconditions	ALEgui and ALEservice started
Actors	End-User, ALEgui
Success	ALEgui is stopped
Failure	
Trigger	The End-user wants to quit the application but wants the service to be still running
Description	1. End-user presses the button "home" or "back" from his mobile phone 2. ALEgui unbind the connection with ALEservice 3. ALEgui close itself 4. ALEservice keeps running
Extensions	

7. Quit completely the application	
Goal	The End-user wants to quit completely the application
Derived from	
Primary actor	End-user
Preconditions	ALEgui and ALEservice started
Actors	End-User, ALEgui, ALEservice
Success	ALEgui and ALEservice are stopped
Failure	
Trigger	The End-user wants quit completely the application
Description	1. End-user clicks on the menu button 2. End-user clicks on the Quit button

	3. ALEgui sends a message to ALEservice 4. ALEservice stops and closes 5. ALEgui stops and closes
Extensions	

8. Start a new period of monitoring	
Goal	The current monitoring period is done, start a new one
Derived from	
Primary actor	ALEservice
Preconditions	ALEservice started
Actors	ALEgui, ALEservice
Success	New period of monitoring
Failure	
Trigger	
Description	1. ALEservice restarts the time for a new monitoring period 2. ALEservice saves activity data on database 3. ALEservice sends a message to ALEgui 4. ALEgui displays information for the new period
Extensions	4.1 ALEgui is not active, ALEservice does nothing with it and continue

9. Signal analyze	
Goal	Signal from accelerometer is analyzed
Derived from	
Primary actor	ALEservice
Preconditions	ALEservice started
Actors	ALEgui, ALEservice
Success	Activity data sent to ALEgui
Failure	ALEgui is not active
Trigger	
Description	1. ALEservice computes data from the accelerometer 2. ALEservice sends activity data to ALEgui 3. ALEservice updates the database

Extensions	2.1 ALEservice is not active, do nothing with and continue

10. Save results on the database	
Goal	Results of measurements are saved in the database
Derived from	
Primary actor	ALEservice
Preconditions	ALEservice started
Actors	ALEservice
Success	Calories information are saved
Failure	
Trigger	
Description	1. ALEservice computes data from accelerometer 2. ALEservice saves data on database
Extensions	

4. Design

The following section described the application's design made before his implementation. All functional and non-functional requirements are derived from the previous scenario.

4.1 Requirements

Functional

ALE records activities levels continuously and in real-time. It detects, while the system is running, every change of acceleration of user and detects if the acceleration reach fixed threshold. For each activity levels detected, ALE records the time period of that activity. Activity levels are analyzed and classify into categories:

- Sedentary (standing, sitting, sleeping)
- very low (walking slowly)
- low (walking normally)
- moderate (walking fast)
- vigorous (running)

ALE is able to monitor activity levels and estimates calories burned while it is active and results are presented for a period of a day. ALE calculates for these period how many calories the user burned and for each activity level, how much time the user spent for it. In advance, ALE estimates a RMR based on the age, weight, height and gender of the user. This estimation also gives how much calories the user burned if he stayed sited during 24 hours. The estimation of calories is automatically updated every two seconds (see section 5.1.5). ALE restarts the period of monitoring for every day at midnight.

The calories burned are estimated by two equations. First, Resting Metabolic Rate (RMR) is calculated with the equation of Harris-Benedict described in the section 2.1.1.1. RMR is based on the user’s personal information: height, weight, age and gender. Secondly, ALE estimates energy expenditure in kilocalories with Metabolic Equivalent Task (MET) based on a table of activities (see section 2.1.1.2) and adjusted with the RMR (see section 2.1.1.3). Then, ALE computes for each activity level the energy expenditure based on the time the user spent on each of them and computes the resting time as sedentary activity to have the final results in kilocalories. The final results represent a period of 24 hours.

The results' accuracy is enough to detect activity ticks and estimates calories for a 24 hours period. We assume the best-effort to classify the different kind of activity levels because each situation is different: user's weight, user's height, clothes, shoes and where the phone was (pocket, bag, and hand).

Non - functional

- Real-time estimation
- Accurate
- Low battery use
- On one device
- Can be run in background
- Minimally obtrusive with no input from the user
- Security and privacy of data
- Minimal storage capacity

For a minimally obtrusive use, ALE is still "alive", i.e. running, even when the screen is locked and when the user starts another application. The user can quit the application manually at any time.

The battery's life today for a smartphones is not powerful enough to keep ALE alive during 24 hours, hence we are limited by the hardware and assume that it is not possible to use ALE more than 8 hours continuously without having to charge the battery.

Human computer interaction

Three screens are accessible by tabs and represent the main graphical interface of ALE. The first screen represents the live monitoring of the current day with two areas:

- Estimation of energy expenditures in kilocalories
- Chart of activity levels

The estimation of energy expenditure is represented by two values. The first one represents the total of calories computed for all the activity levels and the second one represent the estimation for a period of 24 hours and presents calories estimated with the RMR when the user is at rest and calories estimated for each activity levels.

The chart is a histogram with a color bar for each category. The bar grows depending on the time detected for the activity. By default, the scale is set at 30 minutes max but ALE can update this scale depending times from activities with step of 30 minutes.

The second screen represents history results. The user can see the results for up to 7 days ago without the current day. The user can also select a date to see results like: time spent per activity levels, kilocalories for activity levels and kilocalories for 24h.

The third screen is just a scroll page with instructions and help for the user.

The user can click on the menu button to open a small dialogue window with two options: "Quit" and "Settings".

- The option "Quit" closes completely the application
- The option "Settings" opens a new page to set users personal information like: gender; age; height; weight.

4.2 Functional building blocks

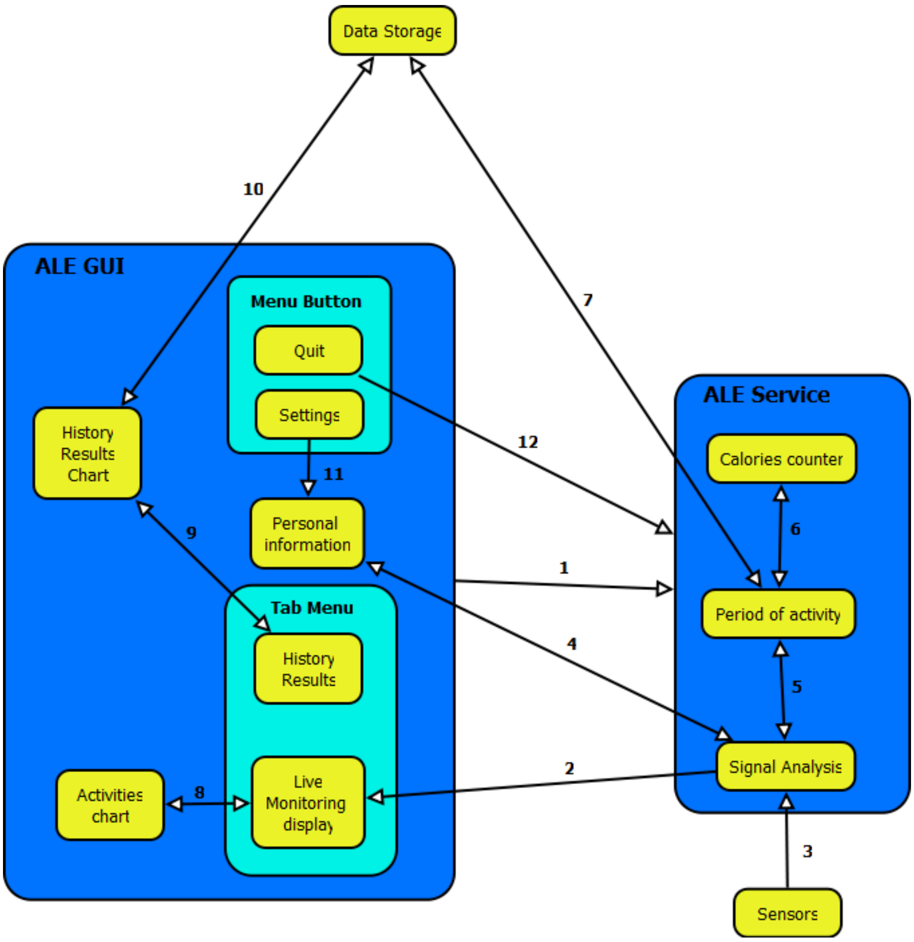


Fig. 13 Functional building

ALE was divided in two main blocks, GUI and Service (Fig. 13) and was relied with two external blocks. The block GUI managed all information displayed on the mobile phone. It managed also menus and interaction with the block ALE Service (1). This one was the block who analyzed and managed signals information from the block Sensors (3). It also managed all information sent to the GUI blocks (2). Below we explain the function of each building block. Numbers in bracket corresponds to arrows from Fig. 13.

Externals blocks

ALE needed these external blocks, and they are provided by the mobile phone.

Sensors

It is a hardware blocks which manage all sensors from the mobile phone. For our case, we used accelerometers and magnetic field sensors.

primitive

`onEvent(3)`, send information from sensors like X, Y, Z data for the accelerometer and magnetic sensor. Information are sent at a frequency defined in the ALE Service block.

Data Storage

Store information like activity results on a database.

primitive

`sendQueryResult(7, 10)`, send result to be stored in a form of from a SQL query.

ALE Service

Principal block that manage and compute information for ALE.

Signal analysis

This block is the core of ALE. It sends activity results to the block GUI after different analyzes, it manages also interaction with the other blocks inside the block ALE Service.

primitive

`sendCompleteActivityResults(2)`, sends results per activity levels to the block Live monitoring display. Results are computed from the block Period of activity.

`getPersonalInfo(4)`, get personal information of the user from the block Personal Information. These information are used for the block Calories counter.

`sendActivityResults(5)`, sends row activity results to the block Period of activity at each event of the block sensors.

`getCompleteActivityResults(5)`, get results from the block Period of activity.

`sendPersonalInfo(5)`, sends personal information of the user to the Block Period of activity.

`onUpdatePersonalInfo(5)`, sends message to others blocks that the user updated his personal information.

Period of activity

It manages the period of monitoring of 24h. At the end of the period (at midnight), it automatically starts a new one. It computes activities results during the whole period and sends complete results (activity level and calories burned) to the block Signal analysis. The block manages also information that are stored on the block Data Storage.

<u>primitive</u>

`sendCompleteActivity(5)`, sends complete activities results to the block Signal Analysis.
`getPersonalInformation(5)`, get personal information from the block Signal Analysis.
`sendActivity(6)`, sends activities results to the block Calories counter.
`getCalories(6)`, get calories from the block Calories counter.
`sendPersonalInfo(6)`, sends personal information of the user.
`onUpdatePersonalInfo(6)`, sends message to the block Calories counter that the user updated his personal information.
`startNewActivity(7)`, creates a new row on the database table for a new 24h period.
`openDataBase(7)`, creates a connexion with the block Data Storage
`createDataBase(7)`, if it is the first time that ALE starts on the device, this primitive will create a new database.
`udpateActivity(7)`, updates activity and calories results on the database.
`getActivityOfTheDay(7)`, check if there is already a row on the table for the current period to continue it or create a new one if it found nothing.

Calories counter

This block computes calories burned for each activity levels and calories burned in a 24h period. It uses personal information from the user and also time spent per activity level.

<u>primitive</u>

`getPersonalInformation(6)`, get personal information.
`sendCaloriesBurned(6)`, send results of the calories computation.

GUI

Manages all GUI of the application and also user interaction with ALE.

<u>primitive</u>

`startService(1)`, when the block ALE Service is not already launched.
`bindService(1)`, when the block ALE Service is yet launched.

Tab Menu

Sub- blocks that manages main GUI of ALE.

Live Monitoring display

Block that displays a chart and information related to activity level from the user. Information are provide by the ALE Service block. The chart comes from the block Activities chart.

primitive

`sendActivityResult(8)`, sends activity result to the block Activities chart

History Results

This block manages the user interface of history results. It displays results from the last 7 days. It manages also the query from users to display a result at a specific date.

primitive

`getPastResult(9)`, asks to the block History Results Chart to send history results from the current date.

`getPastResultOfDay(9)`, asks to the block History Results Chart to send history results from a specific date.

Activity chart

Creates the graphical chart from information send by the block Live Monitoring Display.

primitive

`sendChart(8)`, sends the chart draw from activity results.

History Results Chart

Creates chart of past results.

primitive

`sendCompleteChart(9)`, sends chart for the last 7 days.

`sendChart(9)`, sends chart of result at a specific date.

`openDataBase(10)`, creates a connexion with the database.

`getActivityOfTheDay(10)`, asks the database to send results at a specific date.

`getActivityOfTheWeek(10)`, asks the database to send results for the last 7 days.

Menu Button

Sub-block that manages user interaction with the menu button from the mobile phone

Quit

Option of the menu to quit completely ALE. The GUI and ALE Service will be stopped.

primitive

`closeService(12)`, message to the block Service that ALE stops.

Settings

Option of the menu to manage user's personal information and launch the display of the block Personal information.

primitive

`startPersonalInfo(11)`, message to the block Personal information (weight; gender; height; age)

Personal information

Block that manages personal information of the user and store results.

primitive

`sendPersonalInformation(4)`, sends information of the user.

`onUpdate(4)`, alerts that the user updated his personal information.

5. Implementation

The ALE implementation was made in two phases. First, we started to experiment with different solutions about the activity monitoring. The next section presents our results experimentation. Then we build the complete application around our activity monitoring algorithm, ALE, and validate it (Section 6).

5.1 Algorithm implementation

To detect physical activity with a mobile phone, we needed a phone containing different sensors to monitor body movements of the user. The most important sensor was the accelerometer sensor (called also 3D accelerometer or Gsensor) which was able to detect movements in 3D. With this sensor, we analyzed its row data to interpret walk or run movements of the user. We observed that the best solution for a generic system adapted for every users was to compensate the gravity (removed gravity values from x, y, z) and with different filters to remove noise in the signal. At first we tried the development of a pedometer (i.e. step counter), but at the end we found that a system to estimate activity levels was more accurate. The creation of an accurate pedometer was not really applicable for a mobile phone and without manual configuration and calibration guided by the user. We had to take into account a lot of variables that decrease the accuracy of the system like: weight; length; types of clothes; shoes that the user wears. For the activity recognition part, we used a system to quantify activity levels in a window time and results were classified based on activity level threshold table. This method was easier to create than pedometer and result showed that it was accurate for different users.

5.1.1 Sensor logging

To monitor physical activity, we needed to detect body movements. For that, the accelerometer was used. This kind of sensor could detect proper acceleration:

"An accelerometer measures proper acceleration, which is the acceleration it experiences relative to freefall and is the acceleration felt by people and objects. Put another way, at any point in spacetime the equivalence principle guarantees the existence of a local inertial frame, and an accelerometer measures the acceleration

relative to that frame. Such accelerations are popularly measured in terms of g-force."(Accelerometer Wikipedia 2010)

The acceleration is relative to the earth gravity. For example, an accelerometer let on a table and vertically oriented will return a result of -9.81 m/s, one G. (Fig. 14) This situation doesn't represent acceleration because the element doesn't move and represent only a force. But *"acceleration causes an inertial force that is captured by the force detection mechanism of the accelerometer".* (Instructables 2010)

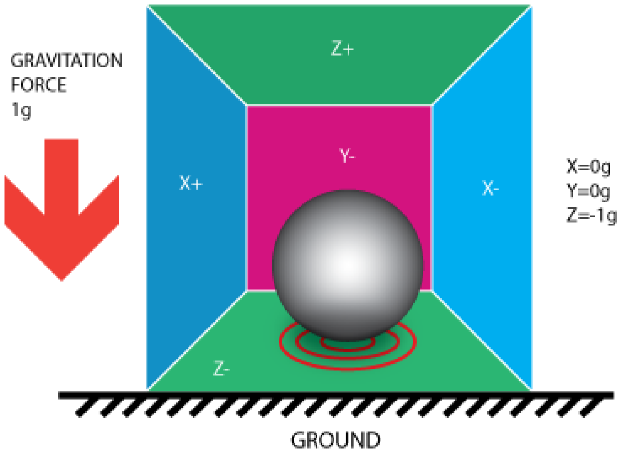


Fig. 14 The ball represents the measured force from an accelerometer let on a table. (Instructables 2010)

To detect an activity in a 3D environment, we needed to use 3 accelerometers which were oriented on orthonormal basis (x, y, z). On movement, the sensor returned 3 results (one result per accelerometer). If the sensor had a lateral acceleration of 1g (on X axis), the result would be:

X = -9.81 m/s (-1g)
Y = 0
Z = 0

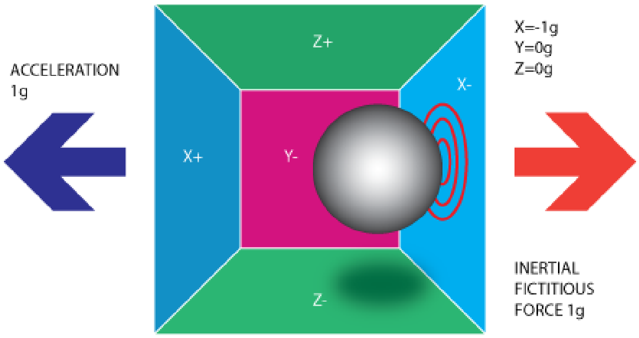


Fig. 15 A lateral acceleration of 1g (Instructables 2010)

On the picture bellow (Fig. 15), the result showed that the accelerometer detected an opposite force of the direction of the movement.

The Android accelerometer was a 3D accelerometer. It returned values (x, y, z) in m/s^2. Every Δt time, the sensor sent data to the device. The frequency was variable and could be parameterized. To analyze a movement in a 3D spatial environment, it was important to compute the three values. For that, we used the dot product (Equation 1). The result represented a vector of acceleration (Va (m/s^2)).

$$Va(m/s^2) = \sqrt{(x*x)+(y*y)+(z*z)}$$

Equation 1 Dot product used for the vector of acceleration

We deployed a small application that monitors row values acquired from the accelerometer and saved in a file. The first result showed that it was important to remove the gravity factor. At rest, the vector of acceleration was always 9.81 m/s. To have a right signal it was not possible to keep the gravity values. We needed to have Va = 0 when no movement was detected. We tried two methodologies to remove gravity. The first one was to subtract 9.81 to the Va. This solution showed that the signal could be negative. For this research, we needed to estimate acceleration on a window time and negative value gave wrong values. The second solution was to remove gravity for each vector (x, y, z) in terms of the orientation before to compute the vector of acceleration. That mean, we removed the gravity factor depending the angle of the vector (x, y, z) with the gravity axis.

In Android API, there were two ways to detect orientations. The first one was to use the orientation sensor. This one returned 3 values (azimuth, pitch, roll) but only when it detected an orientation change. That meant the sensor could not be synchronized with the accelerometer sensor. The Android API references gave also the warning that

the orientation sensor was not a real device, but provided results by calculation. We used the same approaches for the second way. We used the magnetic sensor to compute on demand the orientation. For that, the Android API gave two functions:
getRotationMatrix: *"Computes the inclination matrix **I** as well as the rotation matrix **R** transforming a vector from the device coordinate system to the world's coordinate system which is defined as a direct orthonormal basis".* (Android.com 2010 SensorManager)

This function computed data from the accelerometer and the magnetometer on a matrix (3x3). The result was used for the following function:

getOrientation: "*Computes the device's orientation based on the rotation matrix.*" (Android.com 2010 SensorManager)

With the matrix calculated before, this function returned an array with orientation values like:

- Azimuth, rotation around the Z axis
- Pitch, rotation around the X axis
- Roll, rotation around the Y axis

Then, with acceleration and orientation values, it was possible to calculate the gravity compensation (*gx, gy, gz*) with the following method:

$$gx = G * (\sin(roll) * (\cos(pitch))$$
$$gy = G * (\sin(pitch))$$
$$gz = G * (\cos(pitch) * (\cos(roll))$$
$$X = x - gx$$
$$Y = y - gy$$
$$Z = z - gz$$

x, y, z are values from accelerometer
Pitch and *roll* from results of the method described before.
X, Y, Z are results of acceleration compensated for gravity.

To test this equation, we calculated the Va with different orientation of the device. Each time, the Va should be equal to 0 (when the mobile didn't move).

5.1.2 Sampling frequency and accuracy of sensors

On the Android SDK, it was possible to setup the accuracy of the accelerometer and magnetic sensors independently with the Sensor Manager. This one allowed changing the accuracy within 4 levels (difference was frequency of update):

- Delay fastest
- Delay game
- Delay normal
- Delay UI

We experimented the fastest and the game delay to see which one was the most useful for our application. The fastest delay had an update frequency of about 20 ms but with a high standard deviation. The delay game had an update frequency of about 40 ms but was more consistent with a low standard deviation. After some test of movements we saw that the fastest delay was too noisy to be used with a generic filter. The delay game was less noisy and more consistent but of course we lose precision. Other levels of accuracy gave a very bad precision because much acceleration was not taking into account due to the slow frequency update. We concluded that the level for "Game" was the best compromise for our experimentation, data was enough accurate to detect movement and peaks of accelerometer. Accuracy of sensors had also an impact on batteries. Sensor drains a lot the battery especially with a high frequency, so the delay was also an important factor choice for this issue.

5.1.3 Data interpretation

Signal for accelerometer data was a succession of peaks for body movements, but after a strong analysis, we were able to determine which peaks corresponded to walk movements and which ones were noises. The following section describes our approach to signals analyzing and we also introduce how to accurately interpret the signal.

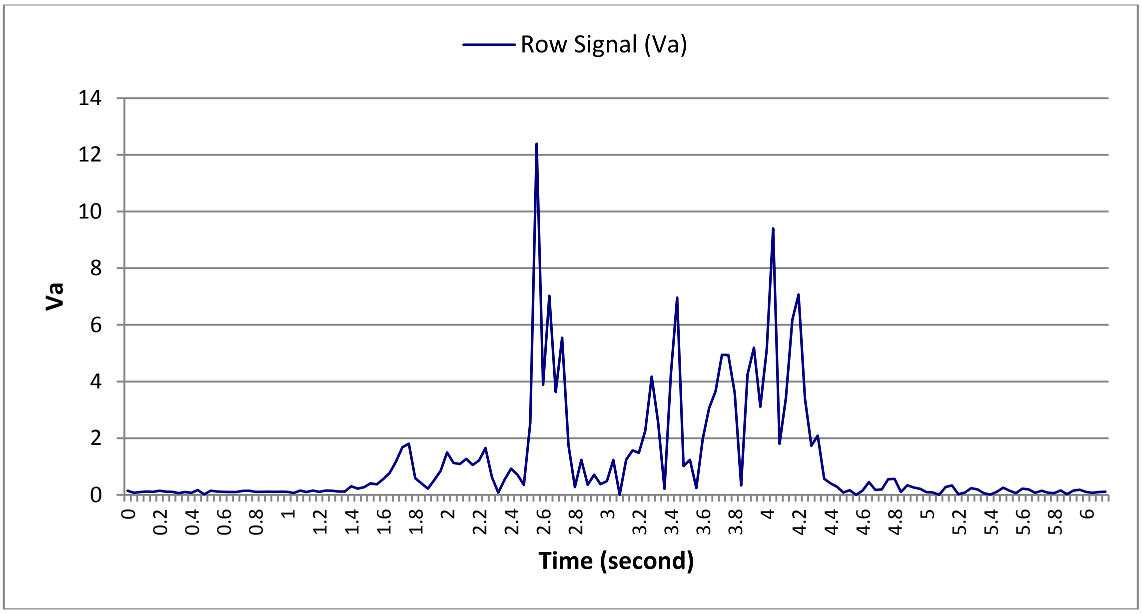


Fig. 16 Row signal of three steps

The Fig. 16 represents the row signal from a simple walk. The walk was a three steps movement, started and finished on the left leg. The mobile phone was on a pocket's jeans on the left side. We found a lot of peaks and also a lot of noise on the signal. The major questions were which peak represented a step and which one was just a rebound or a noise? It was hard to answer these questions but we found groups of peaks (Fig. 17) on the signal. Each group represented a step.

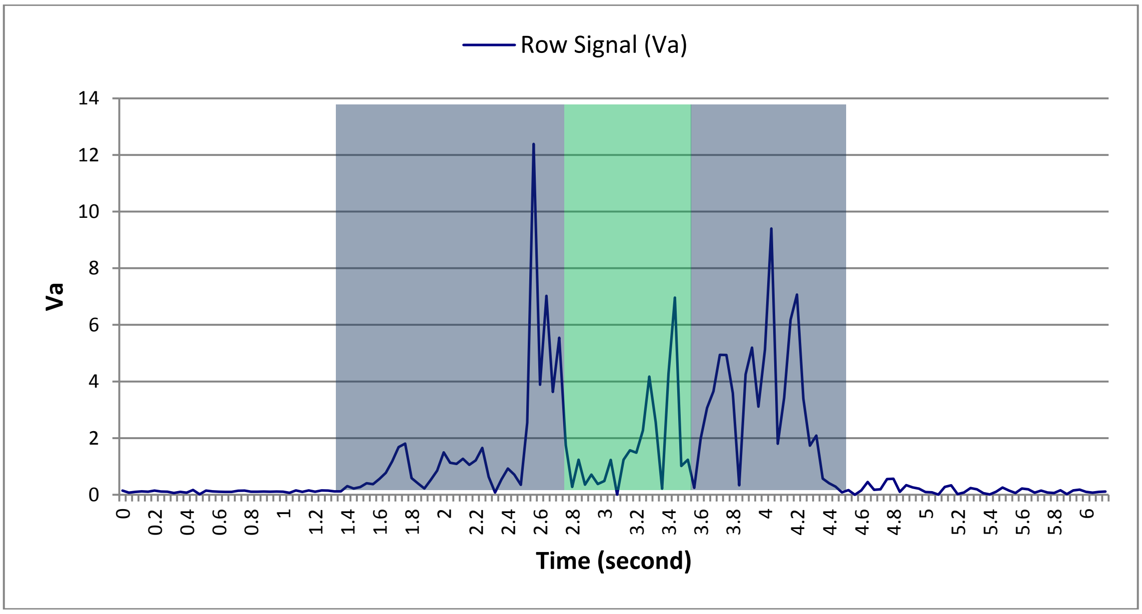


Fig. 17 Groups detected on the row signal (blue: left leg - green: right leg)

The previous figure showed that groups were not similar. To understand that, it was important to decompose the movement.

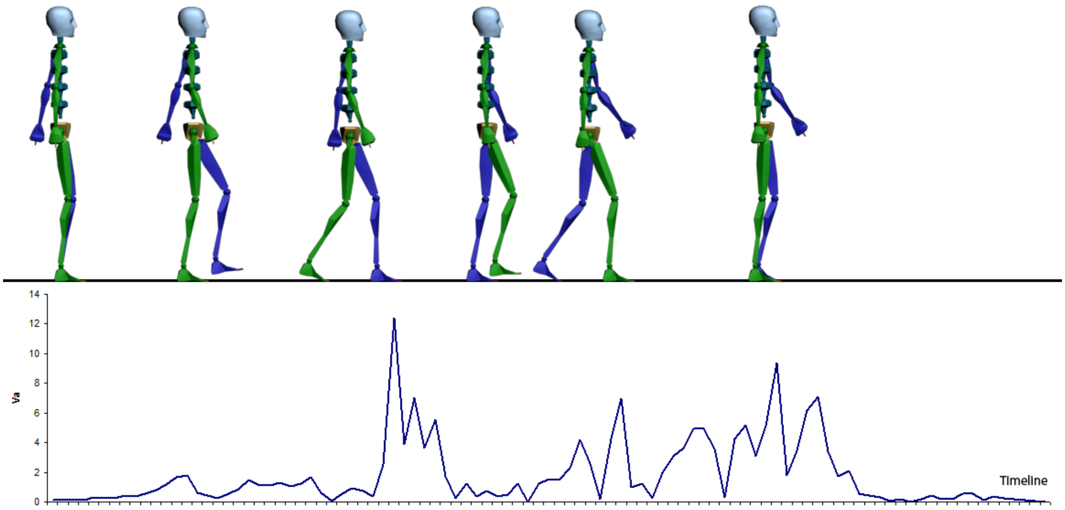


Fig. 18 Decomposition of a walk movement (skeleton model from Autodesk 3dsMax)

In this figure (Fig. 18), we can observe that the first movement was very slow. The user started with both legs on bottom and then he accelerated. The first high peak represented the effect when the foot was at the bottom (1^{st} step) and peaks that following the first one were noise elements because the mobile phone bounced on the pocket. The second step followed directly and the movement accelerated. The peaks, when the right leg touches the bottom, were lower than the left leg for two reasons. Firstly, the movement was made on the opposite side of the mobile phone; secondly, the movement was mixed with the one of the left leg and also with the whole body movement. For the third step, the acceleration was more constant because the movement was complete (both legs, body) and also faster. At the end, we also observed that the mobile phone bounced on the pocket when the user stopped to walk. All this decomposition of the movement was analyzed with an empirical way.

5.1.4 Signal filtering

To create a pedometer or an activity level evaluator, we needed to have a signal without noise and to analyze it with an algorithm. We tried different methods to create filters that would remove noise from signal. This part was very difficult because we had to find good parameters to create a generic filter that would match different users. The reason of the generic filter was that each user walks differently. We used two different filters:

1. The first one was a filter that keeps max values on a sample. This filter was very useful to condensate the signal and also to remove low values.
2. The second one was an average filter that computed for each data points an average on a sample to remove all resting noise of the previous filter.

We started with the "max values filter" because this method kept all max values instead of the average filter that removed peaks intensities.

The max values filter kept only high values on a time window. For each data points of the signal, the filter took 16 followers data points to compose a sample. Then it caught the max values of the whole sample and finally restarted with the next data points. We defined the size of the sample at 16 data points. We tried different values to find the best size of the sample because we wanted a filter that would remove noise without eliminate peaks accelerations. A small sample keeps noisy information and a big one removes the peaks from real acceleration.

Round 1:

```
a = element of the signal (Va)
sample(size 16)  = [a; a+1; … ; a + 15)
aFiltered = MAX(sample)
```

Round 2:

```
a+1 = element of the signal (Va)
sample(size 16)  = [a+1; a+2 … ; a + 16)
a-1Filtered = MAX(sample)
```

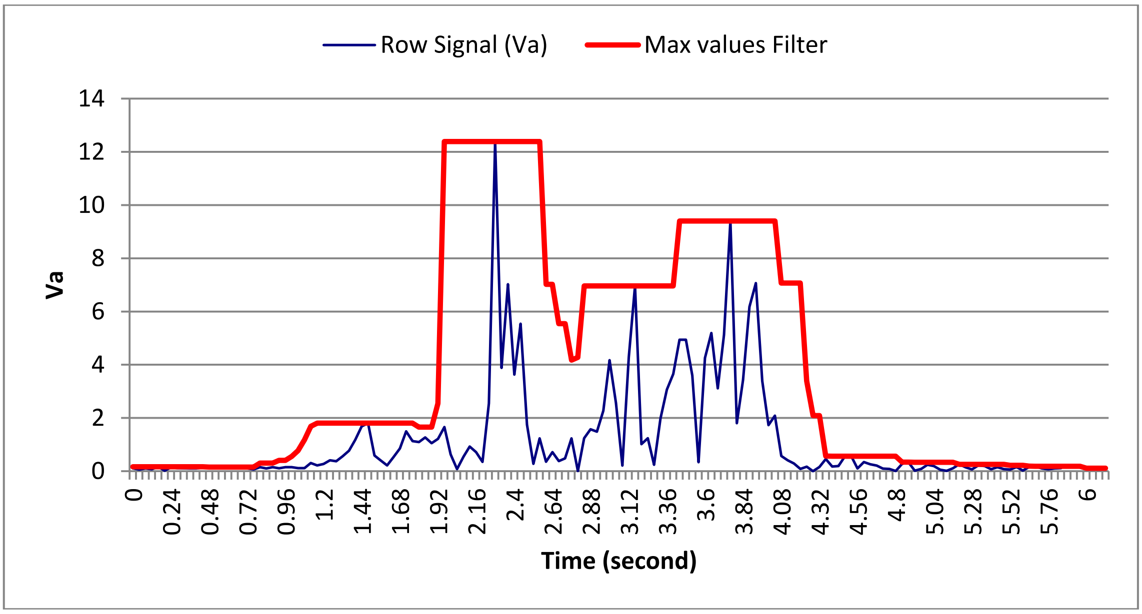


Fig. 19 Result of the Max values filter

In the Fig. 19, the filter removed a lot of information from the row signal but kept values about peaks. For our experimentation, we needed only levels of activity, that's why we used strong filters. We also used this sample's size to condensate on a window time the activity into groups of peaks.

The second filter – "Average filter" – was only used to remove rest of noise and also to have a smother signal (Fig. 20). The filter worked in a similar way than the Max values filter with a moving sample, but the size's sample was different. We divided per two the size's sample of the previous filter (size = 8). We experimented that if the sample's size was smaller, the filter became useless and for the inverse too strong.

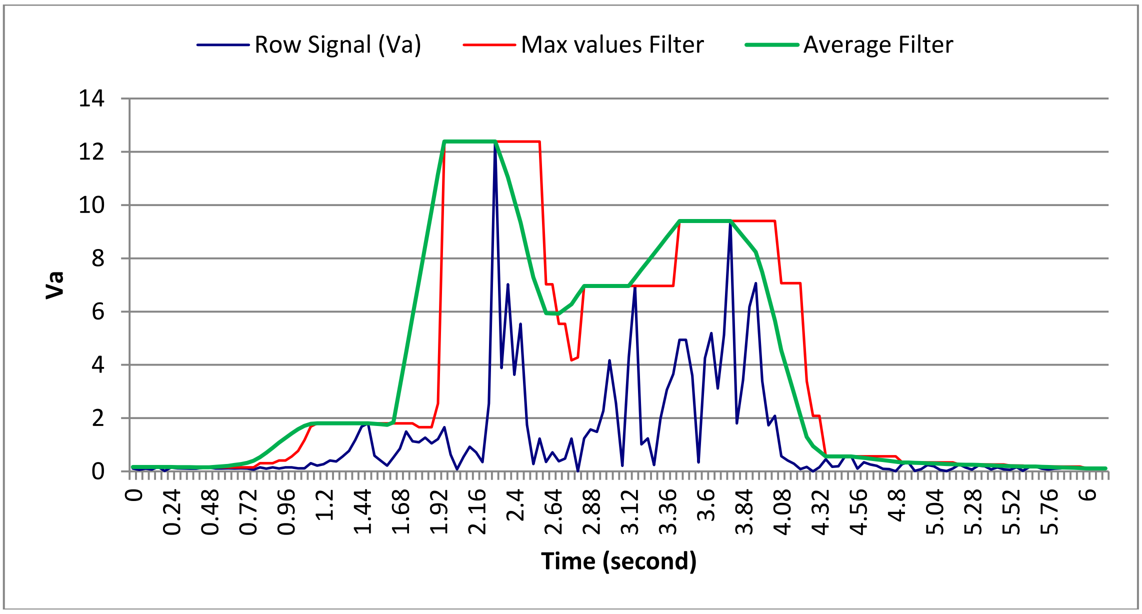


Fig. 20 Result of the Average filter

5.1.5 Threshold determination

Activity levels represented categories depending on the intensity of the body movement. To detect these levels during body movements, we represented them by thresholds. This section will talk about our experimentation on calibrating thresholds of the vector of acceleration to finally classify the acceleration into activity levels. To define these levels, we followed the ones from BodyMedia SenseWear (see Section 2.2.3 and Section 6).

Activity level categories:

- Sedentary (standing, sitting)
- Very Low
- Low
- Moderate
- Vigorous

Thresholds were estimated on a window time. It means that we analyzed the signal during a short time then we compared it with a threshold table to see if the activity during the period corresponded to a level.

To detect the activity level on a period of time (See next section) we used filters described before, and then we computed the median for the full period (Fig. 21). We used median instead of an average because the average was too influenced by peaks from noisy information. The median reflected more activity levels depending on how much the signal was condensate. Nobody used median on the other research presented at the Section 2.2.2.1. We found this solution via experimentation. We observed that if the movement was very vigorous, the signal was higher and more concentrated and gave a higher median. That means that the signal had less lower values than for a low level signal.

After multiple experimentations we saw that many times the median results were to close among them and that's why we added a factor to increase the signal (by empirical experimentation, we defined the factor at 20%). We used this factor to increase the interval among thresholds and also to have a generic application for a lot of different users.

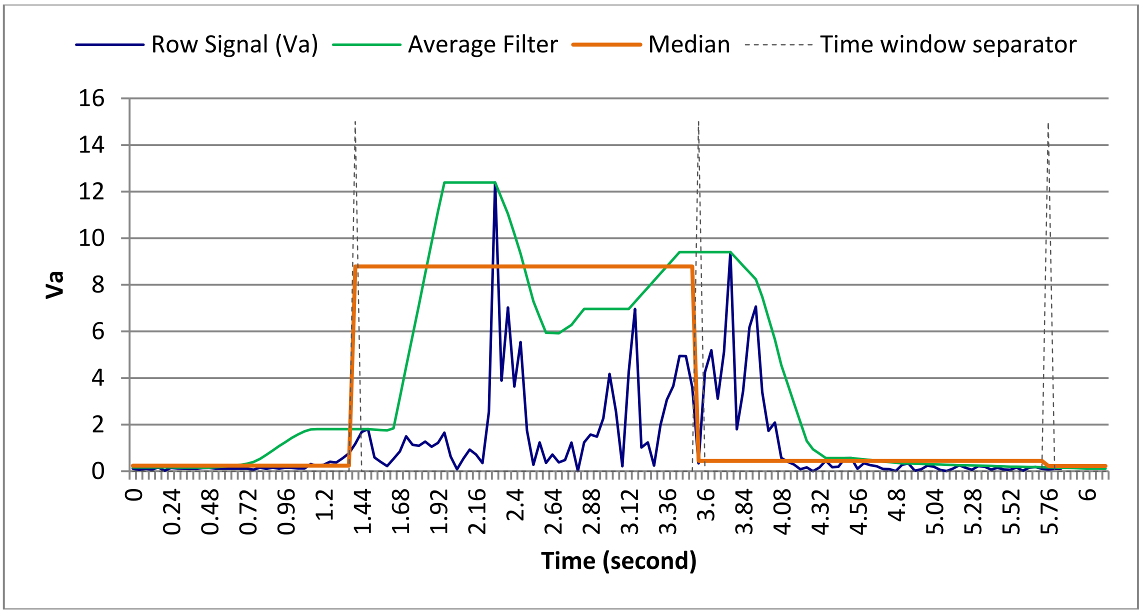


Fig. 21 Threshold in a time window (≈ 2 seconds)

5.1.5.1 ALE service

The Fig. 22 represents the life-cycle of our experimentation of ALE development. There was two periods of window time for monitoring data points (Va). The first one recorded data points during a fixed time, the second caught additional data points to be used for filtering. As we described before (see Section 5.1.4), the "max values filter" used a sample of 16 data points. To analyze completely each data points from the first time window, we needed to add 16 data points. Then, the application started to filter and analyze the signal. At the end, additional data points were copied on the next window time before starting the new turn. Only the first turn didn’t have previous data points on window time.

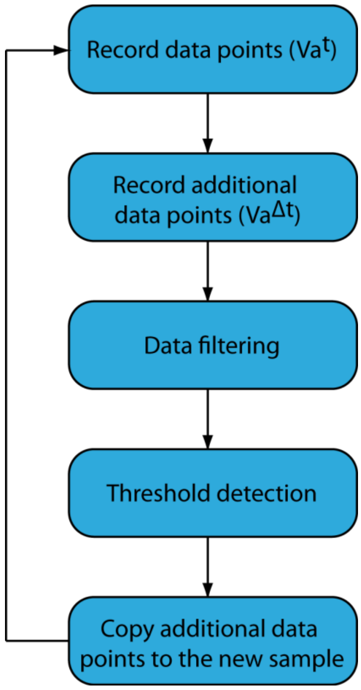


Fig. 22 ALE life cycle of the activity monitor function

The Fig. 23 represents the life cycle described above on a time line. The first loop contains fewer data points than the next one because it started from 0 and there were no previous data points. The time to complete the complete cycle was variable with about ± 100ms.

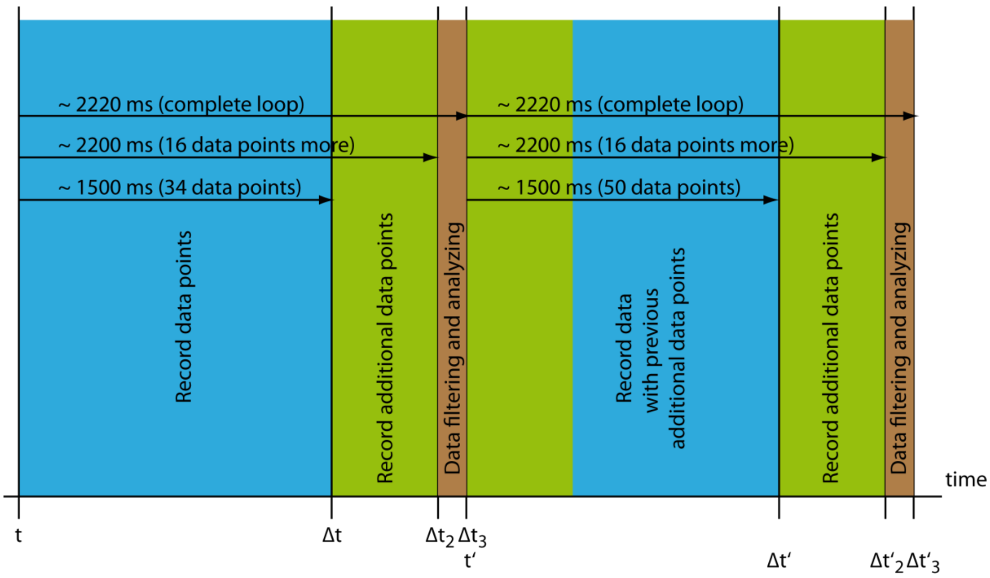


Fig. 23 Life cycle on a time line

We set the window time (Δt) at 1500ms. As other parameters, we made this choice via heuristic reason. If the Δt was too short, the activity level was too variable among window time and if the Δt was too long, the activity level was minimized by low data points. We don't have comparison with other systems. The time for the part “filter and analyze” was generally short with an average of 20ms but sometime, for external reason – the mobile phone used processor resources – this time could be longer (max 70ms). The frequency of elements returned by the sensor was about 40ms. For majority cases, the application didn’t lose a data point while processing the “filter and analyze” part, excepted when the processor was used for other application. For this case, it append that one data point was not recorded.

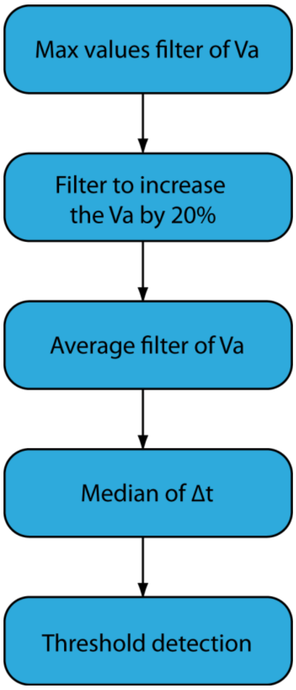


Fig. 24 Life cycle of the « data filtering » part

The “data filtering” part (Fig. 22) was a set of different steps. The Fig. 24 presents these steps. In first, we started to filter the signal with the max values filter then we increased the signal by 20% (explanation above). We passed the second signal – average filter – and finally computed the median for the whole sample. The last step was to detect if the median reached a threshold corresponding to an activity level.

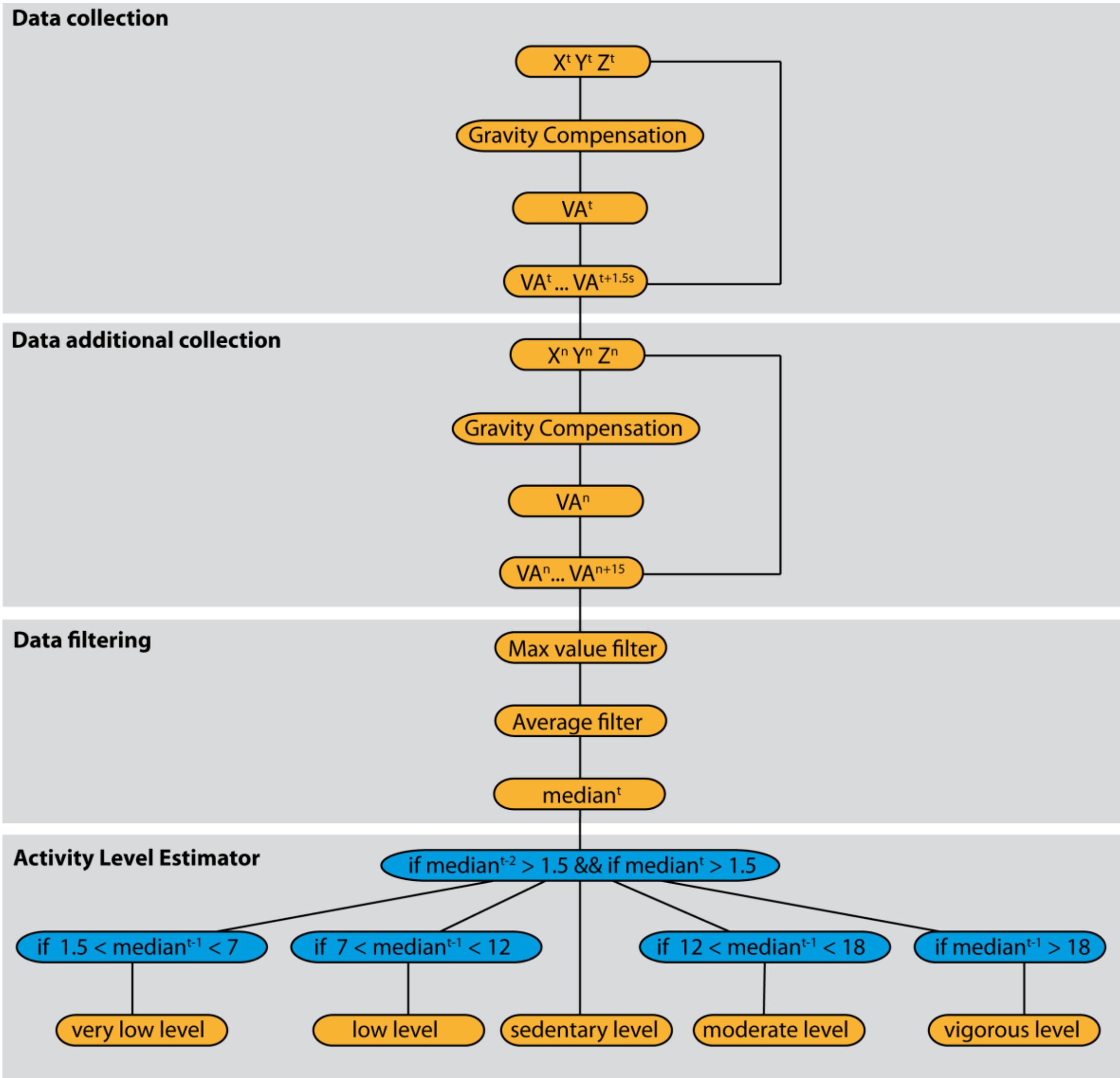


Fig. 25 ALE algorithm

5.1.5.2 User experimentation

To validate thresholds for each activity's level, we started experimentations with 15 users. We asked them to walk minimum 30 steps with different speed of walk corresponding to activity levels and with a pause of 5 seconds in between. We also asked different information about users: weight, height, clothes and kind of shoes (Table 3). These data were to see if one or multiple elements influenced the acceleration vector. Thresholds were determined in first by an empirical way then we did some adjustments after the user test. The first definition of threshold has been defined according to the study of the first user. Then we made the study with 15 others users and we adjusted the definition based on the results.

User N°	Weight	Height	Clothes	Shoes	Gender
1	75	175	Dress	City shoes	Female
2	87	183	Jeans	City shoes	Male
3	85	165	Jeans	City shoes	Female
4	72	174	Pants	Sport shoes	Male
5	52	170	Jeans	Sport shoes	Female
6	82	172	Jeans	House shoes	Men
7	60	162	Jeans	City shoes	Female
8	51	156	Pants	City shoes	Female
9	65	178	Jeans	City shoes	Male
10	77	178	Jeans	City shoes	Male
11	57	161	Jeans	City shoes	Female
12	100	180	Pants	City shoes	Male
13	72	174	Jeans	City shoes	Female
14	92	181	Pants	City shoes	Male
15	59	165	Jeans	City shoes	Female

Table 3 Subject characteristics: Threshold determination

Before presenting the results, we will explain how data were measured and also which information were valid for our threshold validation. As we explained before, we measured the median of the activity signal during 2s. Not all medians were valid because we wanted to remove peaks of movements that were not - in the reality - a true movement. For example, our application will detect a movement when the user put the mobile phone on his pocket. For that, we added the condition that median was valid only if the median before (n-1) or after (n +1) was also valid. We also added the condition that if the median was under the first threshold, we didn't record it.

The Fig. 26 presented the example test result of the user 9. We could observe three groups of movements – as asked – and also the median computed (the green bold line). We saw that the three activities reached each time the corresponding thresholds. The first and the last median measured on each groups showed that it was all the time below the threshold. When the user started to walk, it was from a non movement and then he accelerated. We found the same reason when the user stopped his walk, the signal went down because the user decelerated. On the second group – low level – one median reached the wrong threshold. It was not really an error from the system, but we observed that users could not walk continuously with the same speed.

Thresholds were tested with these values:

- Sedentary: Va median < 1.5
- Very low: 1.5 <= Va median < 7
- Low: 7 <= Va median < 12
- Moderate: 12 <= Va median < 18
- Vigorous: 18 <= Va median

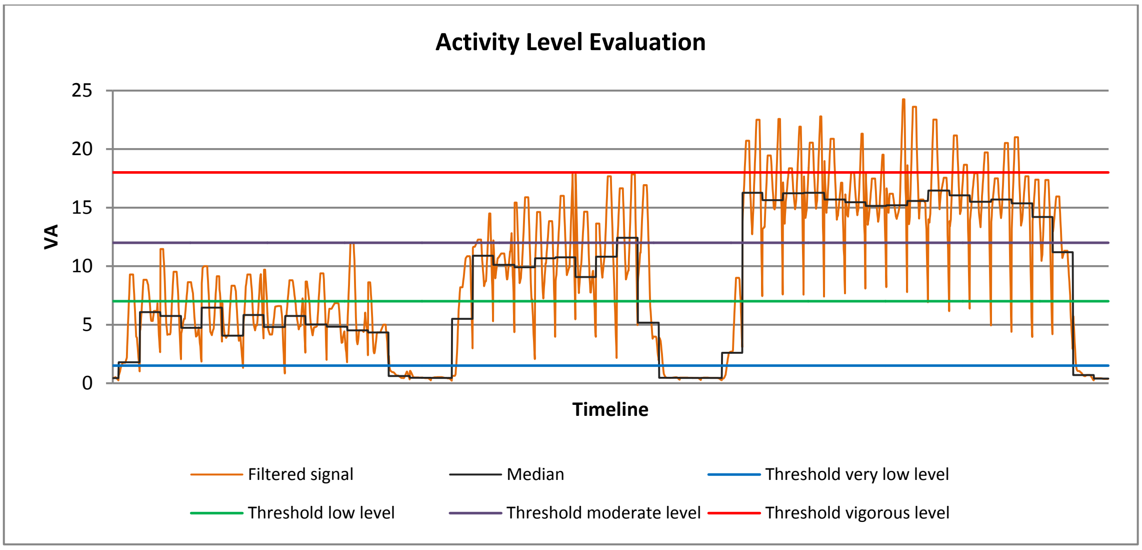


Fig. 26 Result from the user 9

5.1.5.3 Thresholds validation

For each user, we collected the results from a csv file created by the ALE prototype saved on the mobile phone memory card. We computed manually - per activity level - the median, the average, the standard deviation, the minimum value and the maximum values. The following Table 4 presents the results for each user.

User		Very low	Low	Moderate
1	Average	4.04	8.06	14.00
	Standard deviation	0.89	0.78	1.51
	Median	3,75	8,28	13,86
	Min	3.23	6.87	12.49
	Max	6.4	9.07	16.67
2	Average	7.23	10.67	15.47
	Standard deviation	1.06	0.92	1.88
	Median	6,92	10,82	14,55

	Min	5.5	9.39	13.74
	Max	9.36	12.12	18.24
3	Average	4.26	6.58	16.71
	Standard deviation	0.84	0.79	2.33
	Median	4,42	6,60	17,20
	Min	2.8	5.5	13.62
	Max	5.18	7.7	19.25
4	Average	5.55	9.84	17.71
	Standard deviation	0.92	1.55	1.8
	Median	5,65	10,01	17,51
	Min	3.6	7.69	15.85
	Max	6.78	11.53	19.99
5	Average	2.17	6.72	13.37
	Standard deviation	0.36	0.41	1.4
	Median	2,27	6,69	13,77
	Min	1.59	6.29	11.48
	Max	2.73	7.18	14.46
6	Average	6.57	12.05	16.34
	Standard deviation	0.9	2.2	1.69
	Median	6,65	12,53	15,67
	Min	5.33	8.69	15.23
	Max	7.7	14.57	18.81
7	Average	4.07	6.84	11.68
	Standard deviation	0.58	0.51	1.75
	Median	4,14	6,62	12,02
	Min	2.84	6.3	8.67
	Max	4.8	7.66	13.89
8	Average	2.8	5.28	8.74
	Standard deviation	1.2	0.92	1.26
	Median	2,58	4,86	9,26
	Min	1.52	4.58	7.09
	Max	5.4	6.8	10.16
9	Average	5.17	10.57	15.75
	Standard deviation	0.77	0.96	0.43
	Median	4,92	10,71	15,66
	Min	4.05	9.07	15.14
	Max	6.46	12.4	16.46
10	Average	3.26	9.03	14.64
	Standard deviation	0.58	0.32	2.16

	Median	3,20	9,03	14,80
	Min	2.43	8.67	11.88
	Max	4.07	9.38	17.08
11	Average	4.37	8.51	
	Standard deviation	0.83	1.19	
	Median	4,31	8,51	
	Min	3.06	6.58	
	Max	6.44	9.96	
12	Average	1.89	8.83	13.29
	Standard deviation	0.34	0.70	1.75
	Median	1,96	8,91	13,37
	Min	1.37	7.81	11.5
	Max	2.47	9.66	14.99
13	Average	3.19	10.22	18.06
	Standard deviation	0.91	0.16	0.71
	Median	3,36	10,22	18,06
	Min	1.54	10.1	17.56
	Max	4.31	10.33	18.56
14	Average	5.42	11.02	16.25
	Standard deviation	1.03	2.18	3.78
	Median	5,32	11,54	17,77
	Min	4.07	7.34	11.95
	Max	7.24	13.17	19.04
15	Average	2.96	6.14	11.54
	Standard deviation	0.41	0.46	0.02
	Median	3,13	6,03	11,54
	Min	2.29	5.76	11.52
	Max	3.51	6.75	11.55

Table 4 Threshold validation: complete results for user tests

We started - before the threshold's update - to observe which variable influenced the vector of activity. It was impossible to take into account all variables so for the threshold validation we only focused on these variables: weight, height and gender. We saw that we didn't have enough results to analyze influences by clothes or shoes and the sample of users was too short to see if the age was an influence.

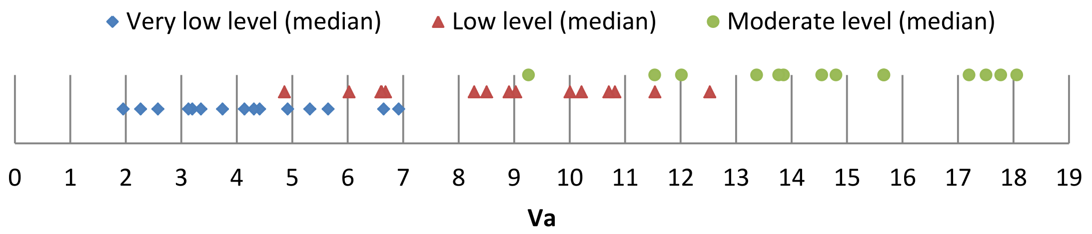


Fig. 27 Chart of median computed

In Fig. 27, each dot represents the value average from the previous table. We separated these dots in groups corresponding to the kind of activity. We found that some dots overlap among thresholds. With the next three tables we tried to find explanation of that.

<u>We remind</u>

- Sedentary: Va median < 1.5
- Very low: 1.5 <= Va median < 7
- Low: 7 <= Va median < 12
- Moderate: 12 <= Va median < 18
- Vigorous: 18 <= Va median

User N°	Weight	Height	Clothes	Shoes	Gender	Median very low level	Median low level	Median moderate level
8	51	156	Pants	City shoes	Female	2,58	**4,86**	**9,26**
5	52	170	Jeans	Sport shoes	Female	2,27	**6,69**	13,77
11	57	161	Jeans	City shoes	Female	4,31	8,51	N/A
15	59	165	Jeans	City shoes	Female	3,13	**6,03**	**11,54**
7	60	162	Jeans	City shoes	Female	4,14	**6,62**	12,02
9	65	178	Jeans	City shoes	Male	4,92	10,71	15,66
4	72	174	Pants	Sport shoes	Male	5,65	10,01	17,51
13	72	174	Jeans	City shoes	Female	3,36	10,22	18,06
1	75	175	Dress	City shoes	Female	3,75	8,28	13,86
10	77	178	Jeans	City shoes	Male	3,20	9,03	14,80
6	82	172	Jeans	House shoes	Male	6,65	**12,53**	15,67
3	85	165	Jeans	City shoes	Female	4,42	**6,60**	17,20
2	87	183	Jeans	City shoes	Male	6,92	10,82	14,55
14	92	181	Pants	City shoes	Male	5,32	11,54	17,77
12	100	180	Pants	City shoes	Male	1,96	8,91	13,37

Table 5 Threshold validation: Results classified by users' weights

This Table 5 presented the results with a classification by weight of users. Values in bold showed results that didn't match with our first threshold estimation. The very low level matches for all users with the threshold of 1.5 to 7 Va. The low level had 5 results under the base threshold of 7 to 12 Va and one higher with the user n°6. Finally, we found 2 values under the moderate threshold (12 to 18 Va). We concluded that the moderate threshold was too high. If we put it at > 6 Va, only 2 values - user n° 2 and 6 - were wrong for the very low threshold and only one for the low threshold (user n° 8).

User N°	Weight	Height	Clothes	Shoes	Gender	Median very low level	Median low level	Median moderate level
8	51	156	Pants	City shoes	Female	2,58	**4,86**	**9,26**
11	57	161	Jeans	City shoes	Female	4,31	8,51	N/A
7	60	162	Jeans	City shoes	Female	4,14	**6,62**	12,02
15	59	165	Jeans	City shoes	Female	3,13	**6,03**	**11,54**
3	85	165	Jeans	City shoes	Female	4,42	**6,60**	17,20
5	52	170	Jeans	Sport shoes	Female	2,27	**6,69**	13,77
6	82	172	Jeans	House shoes	Male	6,65	**12,53**	15,67
4	72	174	Pants	Sport shoes	Male	5,65	10,01	17,51
13	72	174	Jeans	City shoes	Female	3,36	10,22	18,06
1	75	175	Dress	City shoes	Female	3,75	8,28	13,86
9	65	178	Jeans	City shoes	Male	4,92	10,71	15,66
10	77	178	Jeans	City shoes	Male	3,20	9,03	14,80
12	100	180	Pants	City shoes	Male	1,96	8,91	13,37
14	92	181	Pants	City shoes	Male	5,32	11,54	17,77
2	87	183	Jeans	City shoes	Male	6,92	10,82	14,55

Table 6 Threshold validation: Results classified by users' height

The Table 6 above presented the same wrong results than the previous table but we saw that Va was very influenced by height, more than for the weight. On comparison with previous tables, we found that gender was also an influence.

User N°	Weight	Height	Clothes	Shoes	Gender	Median very low level	Median low level	Median moderate level
8	51	156	Pants	City shoes	Female	2,58	**4,86**	**9,26**
15	59	165	Jeans	City shoes	Female	3,13	**6,03**	**11,54**
3	85	165	Jeans	City shoes	Female	4,42	**6,60**	17,20
5	52	170	Jeans	Sport shoes	Female	2,27	**6,69**	13,77
7	60	162	Jeans	City shoes	Female	4,14	**6,62**	12,02
1	75	175	Dress	City shoes	Female	3,75	8,28	13,86
11	57	161	Jeans	City shoes	Female	4,31	8,51	N/A
13	72	174	Jeans	City shoes	Female	3,36	10,22	18,06
12	100	180	Pants	City shoes	Male	1,96	8,91	13,37
10	77	178	Jeans	City shoes	Male	3,20	9,03	14,80
4	72	174	Pants	Sport shoes	Male	5,65	10,01	17,51
9	65	178	Jeans	City shoes	Male	4,92	10,71	15,66
2	87	183	Jeans	City shoes	Male	6,92	10,82	14,55
14	92	181	Pants	City shoes	Male	5,32	11,54	17,77
6	82	172	Jeans	House shoes	Male	6,65	**12,53**	15,67

Table 7 Threshold validation: Results classified by gender

This Table 7 was very interesting for our threshold validation because we found that - in general - females had lower values than males. As we explained for the first table, if we changed the Low threshold at 7 to 6, all results for women would be right. But for men, it was not necessary to update this threshold. Same observation with the third threshold Moderate at 12 to 11.

Finally, we decided to do a variable set of thresholds depending on the user's sex:

Male

- Sedentary: Va median < 1.5
- Very low: 1.5 <= Va median < 7
- Low: 7 <= Va median < 12
- Moderate: 12 <= Va median < 18
- Vigorous: 18 <= Va median

Female

- Sedentary: Va median < 1.5
- Very low: 1.5 <= Va median < 6
- Low: 6 <= Va median < 11
- Moderate: 11 <= Va median < 18
- Vigorous: 18 <= Va median

To resume, we mainly validated our threshold with empirical experimentations. We did some assumptions to explain the influences of variables.

Regarding the height, we observed that it was an influence but tried understood the reason. We tried to understand if the fact to have longer legs is a case of influence. When the mobile phone is on the pocket, it follows the rotation of the leg, so the movement can be measured by the angle made by the leg. Of course, if the user is tall, the distance - performed by the foot - from the start to the end of the step will be higher than a smaller user, but angle will be quite the same. On the other side, the movement's speed will be higher with a taller user. As we explained, the distance made by the foot was higher. We can also seek an explanation that the user's height was also an effect on the user's weight.

About the weight, we did the assumption that the acceleration was influenced by the weight as a vector of force.

For the gender's variable we could not explain only with variables weight and length because it was not always the case. For this variable, we didn't have explanation apart with an anatomy prospective and the way we walk.

5.2 ALE implementation

In the section 4.2, we presented functional blocks to describe the main structure of ALE. In this section we present the implementation of this structure. The architecture of ALE doesn't correspond to the design structure, but functionalities are the same. A lot of functionalities described on the Section 4.2 are deployed in the same class.

5.2.1 ALE classes

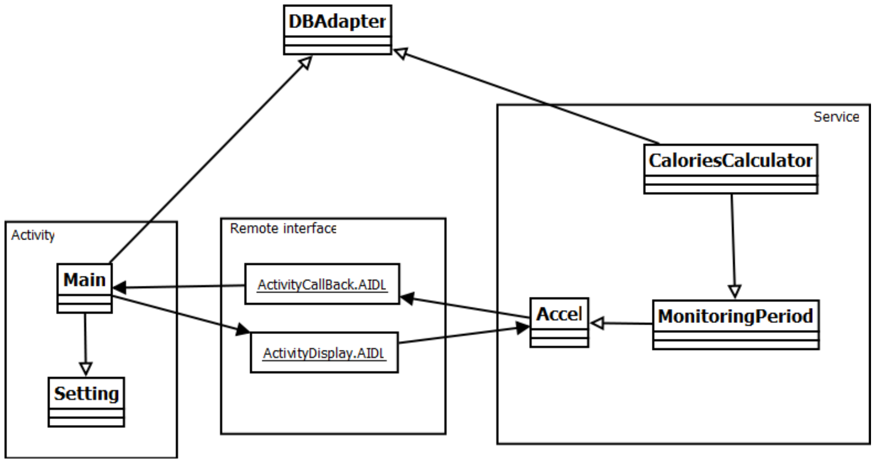


Fig. 28 Classes diagram of ALE

ALE is composed by two principal classes (Fig. 28). One Android Activity (Main.class) and one Android Service (Accel.class). The Android SDK makes this distinction to separate applications that run in the foreground with a GUI (Activity) and applications that run in the background without any GUI (Service). Activity and Service can be independent or bind trough the same package or not. An activity can communicate with different active services. To have a communication between activity and service, the SDK proposes two methods.

1. Create a direct communication on case that activity and service are on the same package,
2. Communicate through a remote interface.

On the Android web page, we found the definition about remote interface:
"On the Android platform, one process can not normally access the memory of another process. So to talk, they need to decompose their objects into primitives that the operating system can understand, and "marshall" the object across that boundary for you." (Android.com 2010 Remote Interface)

The definition above says that it is not possible to have access to the memory of another process. It is right, but for the case of an Activity and a Service on the same package, it is possible to create a communication because both classes work on the same memory.

The Android SDK provides the AIDL tool to generate codes that transform objects into primitives. AIDL mean "Android Interface Definition Language". For ALE, we used the AIDL tool to create two remote interfaces (Fig. 28). The communication between processes works only in one way, that's why we created an interface to sends message from the Activity to the Service and vice-versa.

5.2.2 ALE Service

The main core of ALE runs on a Service and sends activity results to the ALE Activity. The Service is composed by one class (Accel.class) that extends the Android Service and two classes (MonitoringPeriod.class; CaloriesCalculator.class) that manages and computes the activities results.

The class Accel contains a lot of functionalities. It is the first class made for our prototype and we built after other functionalities on it. The class contains the module to manage the remote interfaces for the communication with the Activity and also the complete algorithm (see Section 5.1) that analyzes the signals sent by sensors. The class also manages the transmission of the data to the other classes and through the remote interface to the Activity.

The class MonitoringPeriod can be described as a workflow. It manages time period of 24h and manages all information that are stored on the database. The class also formats the results to be sent to the Activity. All transactions to the database are managed by this class. It creates a new entry for each period of 24h and also for each event sends by the class Accel. The class checks also the actual time and creates automatically a new entry on the database at the end of the period of the day.

The class CaloriesCounter computes on demand calories burned during a 24h period. It is based on the user's RMR and MET. We used same equations described in the section 2.1.1. MET are adjusted with the user's RMR. We also added a method to update the RMR when the user changes his personal information.

The class works in two phases. When the Service starts, the class computes calories estimated for 24h based on the RMR of the user. Then, on demand from the class

MonitoringPeriod, the calories counter updates the calories estimation with the values from activities level. By example (unit time is in hour):

Very low level = 1h
Low level = 0.15h (30 minutes)
Moderate level = 0h
Vigorous level = 0h

For each activity level, the counter computes calories burned with the time spent for it and also with the weight of the user. Each level corresponds to a MET value: (MET values are not adjusted for this example, its only basic values. See the Section 6 for explanation of these values):

Very low level = 2.5 MET
Low level = 4.5 MET
Moderate level = 6 MET
Vigorous level = 9 MET

Then, the counter computes calories with the weight of the user (65kg in this example) with this equation: *MET * time(h) * weight(kg)*

Very low level = 81.3 kcal
Low level = 43.9 kcal
Moderate level = 0 kcal
Vigorous level = 0 kcal

The next step computes calories for the non-activity. Period of non-activity corresponds to 24 hours minus time from other activities.

Sedentary period = 24 - 1 - 0.15 = 22.5h
Sedentary MET = RMR kcal-kg^{-1}-h^{-1}
Sedentary calories = 1462.5 kcal / day.

Finally, the counter adds all calories results.
Total calories: 1462.5 + 81.3 + 43.9 = 1587.7 kcal.

5.2.3 Data storage

The data is stored on a SQLite database provides by the Android SDK.

The database is composed by one table:
`Activity{ id | date, veryLowLevel, lowLevel, moderateLevel, vigorousLevel, calories, totalCalories}`

`id`, primary key and composed of integers number.

`date`, date of the period of activity, text field
`veryLowLevel`, activity time, text field
`lowLevel`, activity time, text field
`moderateLevel`, activity time, text field
`vigorousLevel`, activity time: text field
`calories`, calories burned for activity levels, text field
`totalCalories`, calories burned in 24h, text field

We created a specific class (DBAdapter.class) to manage the database with different functions used by ALE, like finding the results at a specific date, or updating a row with new values of activities. We used text field for a better compatibility with SQLite.

5.2.4 Android Activity

An Activity is considered as an independent application. It is generally composed by a class Main and a XML layout. The layout defines the GUI for the application. For ALE, the main interface is defined into the XML file and call by the class Main.

The class Main manages all information required by the different tabs on the GUI. The user interface is composed by three main screens and a tab menu. This menu manages the visibility of a page. The first page represents live values sent by the module ALE Service (Fig. 29). The second page displays past results for 7 days (Fig. 30) and the last page describes the application and also gives some FAQ for the user.

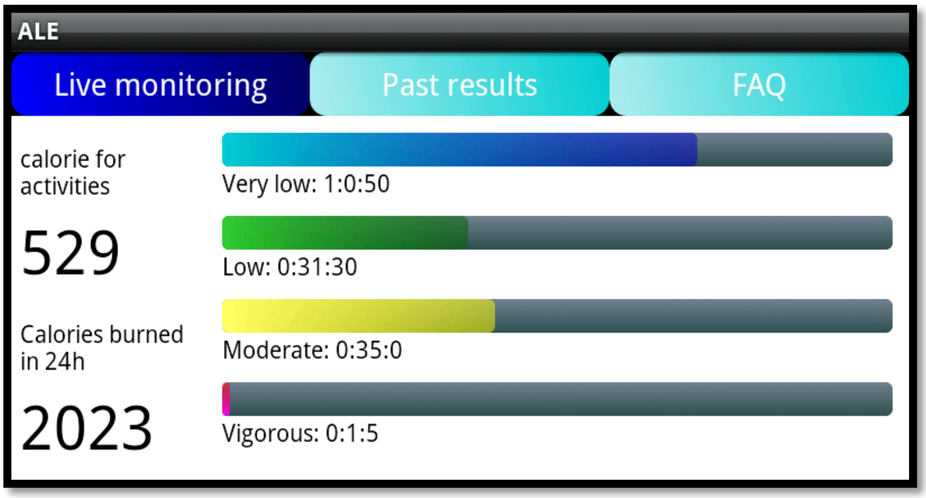


Fig. 29 Main screen (live monitoring) of ALE

Fig. 30 This screen of ALE represent past results

All graphical elements are created and updated inside the class Main. It updates graphical bar (Fig. 29) that represents activity levels at each time Service send a new value. The scale of the bars is by default set for 30 minutes because with a higher scale, the bars displayed were too short when the user was not active during the day. If an activity level is bigger than 30 minutes, the class automatically updates the scale with an increment of 30 minutes.

When the class Main starts, it launches automatically the Service if it is not yet active. In the case of the Service is yet active, the Main class creates a link with the Service.

The class Main displays a menu with option when the user presses the "menu" button on his mobile phone. This menu allows the user to quit the application and also to manage his personal information with the class Setting. This class is independent of the class Main but called by it.

The class Settings extend functionality provided and called PreferenceActivity by the SDK. This component creates automatically a graphical configuration page and manages the data storage. The configurations are defined on a XML file and called by the class Setting. We used this functionality to manage personal information asked to the user: weight, height, age and gender. The data are automatically stored on the application's cache and can be retrieved by another Activity or Service inside the same package.

Fig. 31 are screenshots from the page Setting. On the left, we see information asked to the user. On the right, the user can record his age on the pop-up window. The main GUI and the pop-up window are automatically created by the Android API.

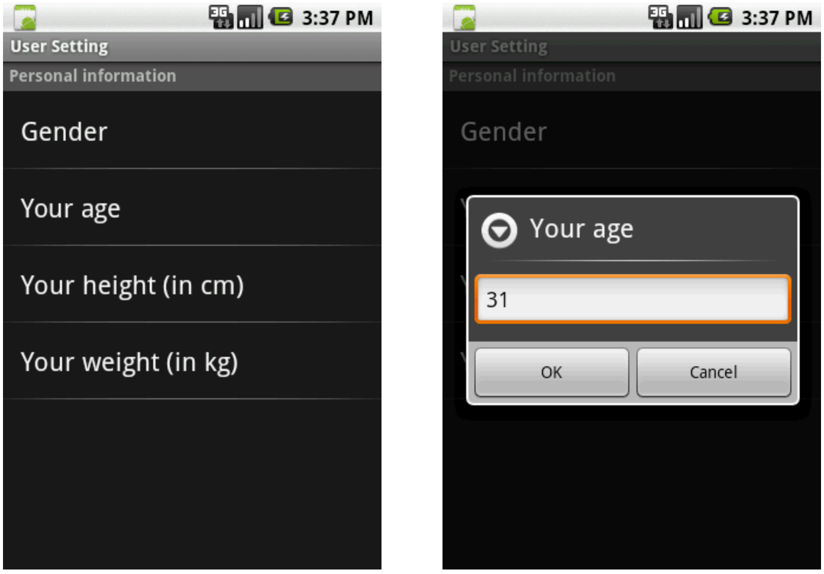


Fig. 31 Screenshot of the setting page

6. Validation of ALE: Comparison with BodyMedia SenseWear

To evaluate the ALE application, we conducted study with different users. To determine how ALE was accurate, we asked users to wear Sensewear (see section 2.2.3) and the mobile phone running ALE on same time. Sensewear was considered the 'gold standard' for ambulatory assessment of energy expenditure (Welk, McClain et al. 2007) and we selected it as the base device to compare with ALE. We conducted this study also to determine for each activity level, the MET corresponding.

Fig. 32 BodyMedia SenseWear device

Procedure

Given the time limit of 2 weeks we have conducted 7 studies with 5 different subjects. Each user was asked to walk. Before the walk, Sensewear and ALE was configured to the user; the weight, height, gender and age were configured. We have made note upon participants cloth (i.e., where ALE was put) and shoes type. SenseWear needed to know if the user was left or right handed. For our experimentation, this was not an important variable because ALE was located near the belt and the side had no impact. We asked the participants to walk outdoors, to have enough space, during a minimum of 15 minutes. We asked participants to walk sometimes slowly and sometimes quickly. Walks were on different places with different grounds. Most of these walks were on road with different elevations.

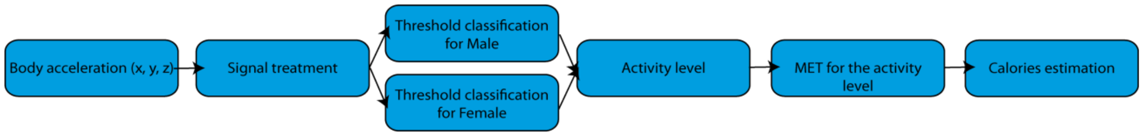


Fig. 33 ALE process

The Fig. 33 explained the complete process of ALE, from the body acceleration to the calories estimation and how from the threshold defined at the Section 5.1.5.3 we estimated MET.

To resume the process

1. The body acceleration is caught by the accelerometer
2. Ale starts the first phases with the signal analysis (filtering and median computing, see Section 5.1.5.1)
3. The result from the previous phase (the Va median) is classified depending thresholds and gender (see Section 5.1.5.3).
4. ALE knows that the result (for the last period of 2 seconds) corresponds to a level of physical activity.
5. An activity level corresponds to a MET value.
6. ALE estimates valorizes burn for the last period of 2 second based on the MET value from the activity level.

MET values on the Table 8 were in the beginning based on activity levels from SenseWear and also based on the table at the section 2.1.1.2. Then, we asked to a user to walk at different speed to calibrate the MET table corresponding to activity levels defined on the implementation. The Table 8 presents MET values defined after calibration for each activity level.

	MET
Sedentary	1
Very low activity	2.5
Low activity	4.5
Moderate activity	6
Intense activity	9

Table 8 MET values per activity level used on ALE

The Table 9 presents activity levels from SenseWear with range of MET.

	MET
Sedentary	1 to 3
Moderate	3 to 6
Vigorous	6 to 9
Very vigorous	9 to more

Table 9 MET values per activity level used on SenseWear

Data Analysis

After the walk, we collected data from both devices and we have prepared it for the comparison. SenseWear returns MET values measured during one minute period, while ALE during a 2 second period. As SenseWear was our baseline device, we computed MET values from ALE to have a MET average for every minute. Then we computed the MET average during the complete walk for both devices. Finally, we measure in percent the difference among results. We measured also the Mean Absolute Percentage Error (Equation 2) (Wikipedia - MAPE 2010) per minutes. These values corresponded to the difference between ALE and Sensewear each minutes, then we computed the average for the complete test's period.

$$\left|\frac{(MET^{ALE} - MET^{SenseWear})}{MET^{SenseWear}}\right|$$

Equation 2 MAPE

Moreover, on a per-minute basis, we computed how much percent ALE overestimated the MET values comparing the value indicated by SenseWear and also how much percent it underestimated the MET. With these estimations, we finally computed the proportion of overestimation and underestimation for the complete period.

SenseWear data

The device exported data on a excel file where we can find different information classified by minute. We collected only the MET values for each minutes. With the device, it was not possible to change the frequency of MET results. MET values were not fixed values like ALE (Table 8), but measured in function of the user's effort. SenseWeare classified activities in four levels (Table 9). We had no information about how the device computed data into MET, however we assumed that the values are accurate as Sensewear is considered the 'gold standard' for ambulatory assessment of energy expenditure (Welk, McClain et al. 2007).

ALE data

Results were log on a file at the end of each loop of 2 seconds. MET values for activity ticks were not variable and corresponded to the Table 8 except the fact that there were adapted to the user, based on the user's RMR (see section 2.1.1.1). We didn't know if SenseWear did the same adjustment based on the RMR like ALE.

Results

Below, we presents results for each tests, at the end, we presents a complete table of all results. For each test, we present the kind of walk, the user's characteristics, a MET average graph, the MET error per minutes and the overall MET error.

User's characteristics mean personal information of the user like weight, height, age and gender. The adjusted MET table (based on the user's RMR, see Section 2.1.1.2) is also shown. On the first graph, we present each time four elements:

- MET from Sensewear (a result per minutes)
- Adjusted MET from ALE (a result per 2 seconds)
- Average MET ALE (average for the whole period of walk)
- Average MET Sensewear (average for the whole period of walk)

Discussion on results granularity

Graphs were useful to analyze if trends from Sensewear and ALE correspond. Most of time, we saw that the curve MET ALE decreased in value, it represented when the user stopped the walk and was standing over the period of less than a minute. Sensewear didn't represent these breaks because values were only returned one MET value every minute. The first graph presents MET results from ALE and SenseWear during the test. It represents also the overall result (Equation 2) for both device and in percent the different of ALE based on SenseWear.

The second graph represents MET values per minute for both devices. With these results we computed the mean absolute percentage error (MAPE) (Equation 2), the over - underestimation percent (Equation 3) and the over - underestimation proportion. We talk about overestimation when ALE MET are over than SenseWear MET and underestimation for the inverse case.

$$\frac{(MET^{ALE} - MET^{SenseWear})}{MET^{SenseWear}}$$

Equation 3 To calculate the over - underestimation. Positive results are overestimation, negative results are underestimation

For each user tests, we present at the end results of overall MAPE, overall overestimation and underestimation with each time the proportion in percent. The overall MAPE is the average of each MAPE computed per minute for the test. For the overall overestimation, we computed the average of each positive value of the Equation 3, and same method for the underestimation but with negative values. For both - over / underestimation - we computed also proportion on the complete test. That mean, how much percent on the whole test, ALE was in an overestimating case and also the inverse. To calculate this proportion, we calculated how much time we had a positive results, so an overestimation, divided by the quantity of results. For example if the test was while 23 minutes, the quantity equal 23. The underestimation was calculated as the percent rest of the overestimation.

6.1 Short term test results

Test 1

Walk during 23 minutes on a street road at different speeds. The road was most of time flat and with low hill. The user stopped walking three times. The user wore SenseWear on the left arm and ALE on the left pocket's Jeans. He wore also standard shoes.

	MET	ADAPTED MET		Subject data	
Sedentary	1	0.95		**Weight (kg)**	65
Very low activity	2.5	2.38		**Height (cm)**	178
Low activity	4.5	4.28		**Age**	31
Moderate activity	6	5.71		**Gender**	Male
Intense activity	9	8.56		**RMR kcal/day**	1640.19
				RMR kcal-kg^{-1}-h^{-1}	1.05

Table 10 MET values and user characteristics for the test 1

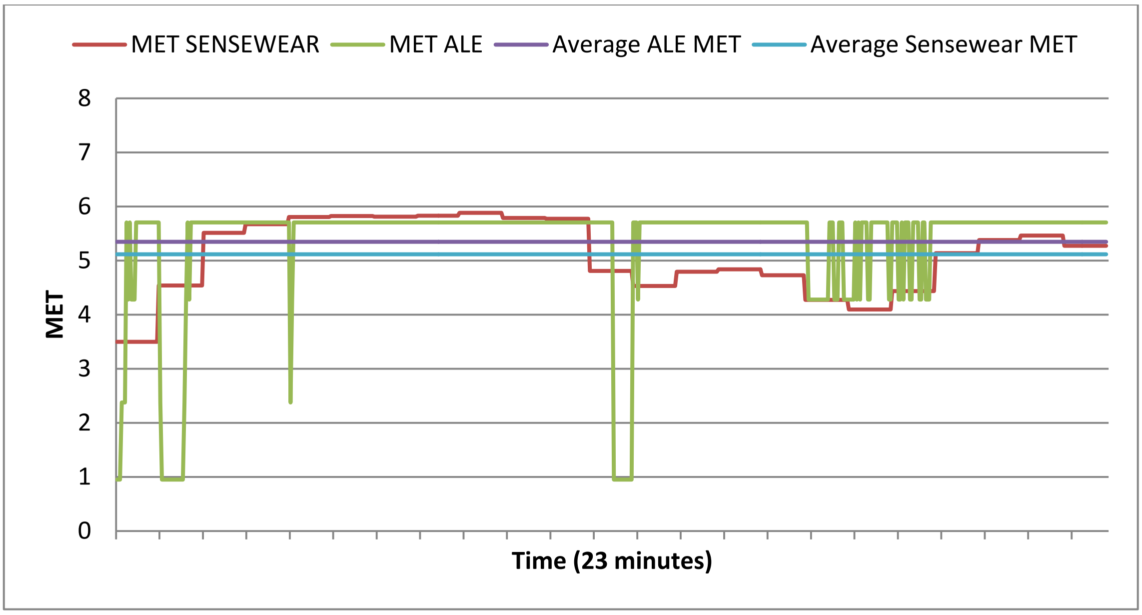


Fig. 34 Representation of ALE and SenseWear data for test 1

	Sensewear	ALE	Difference	MAPE
Average MET	5.12	5.35	0.23	4.47%

Table 11 Difference in percent for the overall results of ALE and SenseWear for the test 1

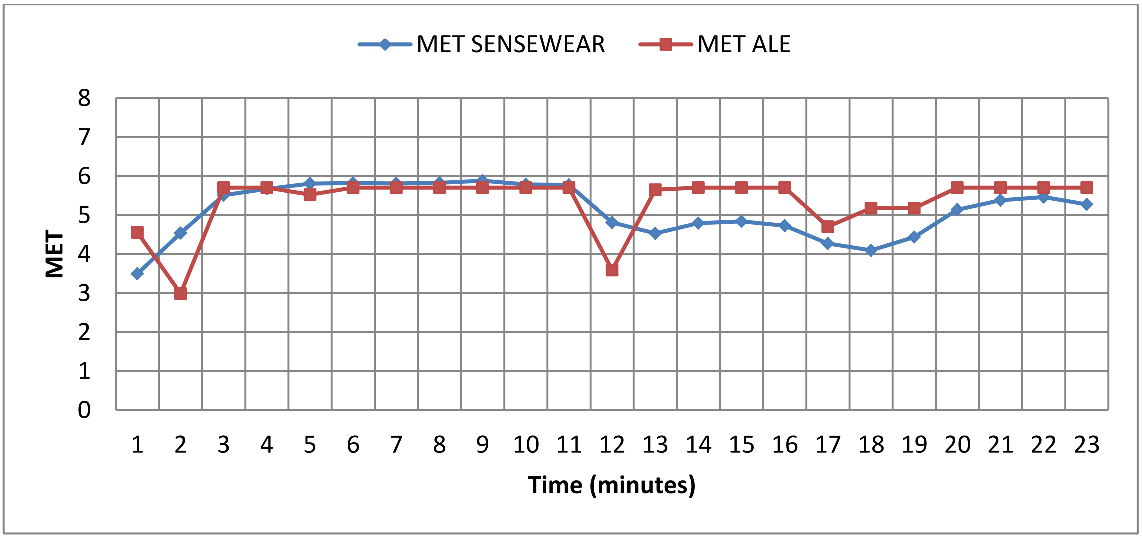


Fig. 35 Results from both device per minutes for the test 1

Results

Overall MAPE	11.98%	Overall Proportion
Overall overestimation	14.26%	60.87%
Overall underestimation	-8.44%	39.13%

Table 12 Results for the estimation error of ALE based on SenseWear for the test 1

Comments

For the given user test, ALE underestimates MET twice (Fig. 34, Fig. 35), at the minute 2 and 12. At these times, the user stopped walking while less a minute. It means that ALE was more sensible for this case than SenseWear. This illustrates our point of difference in results granularity. The overall underestimation on Table 12 represents this case.

On the second part of the graph (Fig. 34, Fig. 35) - from the minute 13 - we noted that ALE overestimates MET. During the test, we asked to the user to reduce his speed and after about 7 minutes to walk faster. MET were estimated from thresholds and for this case, the walk activity was the edge of being classified as low/moderate level and ALE classified it as a moderate level (see 5.1.5.3).

If we compare the MAPE results from the Table 11 and the Table 12 were different. On the Table 11, most errors were compensated by the over / underestimation trends during the whole test and didn't represent the real error per minutes while Table 12 represented more detailed minutes-by-minutes results. As MET value was used to calculate how much calories the user burned, it was possible to estimate that with the MET average and see the difference in the final result of the test. So for this case, we found that ALE overestimate of 4.47% the total calories burned during the test.

Test 2

Walk during 23 minutes on a street road, with low difference of elevation. (Same walk as the test 1). The user wore SenseWear on the right arm and ALE on the right pocket's pant. She wore also standard shoes.

	MET	ADAPTED MET		Subject data	
Sedentary	1	0.95		**Weight**	52
Very low activity	2.5	2.38		**Height**	156
Low activity	4.5	4.29		**Age**	28
Moderate activity	6	5.72		**Gender**	Female
Intense activity	9	8.57		**RMR kcal**	1309.68
				RMR kcal-kg^{-1}-h^{-1}	1.05

Table 13 MET values and user characteristics for the test 2

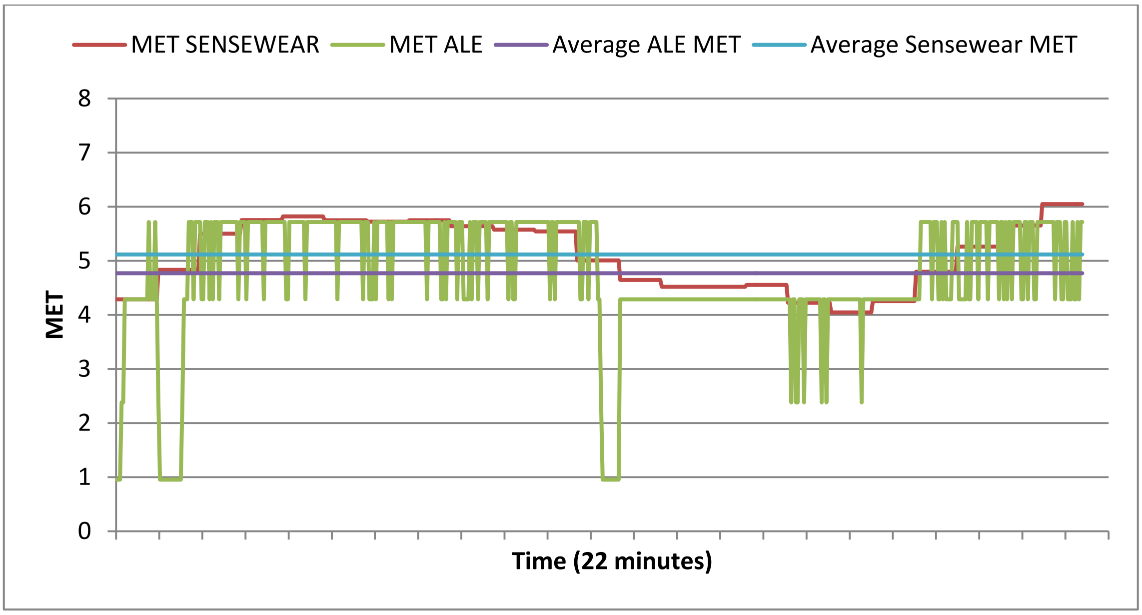


Fig. 36 Representation of ALE and SenseWear data for the test 2

	Sensewear	ALE	Difference	MAPE
Average MET	5.12	4.77	-0.35	-6.77%

Table 14 Difference in percent for the overall results of ALE and SenseWear for the test 2

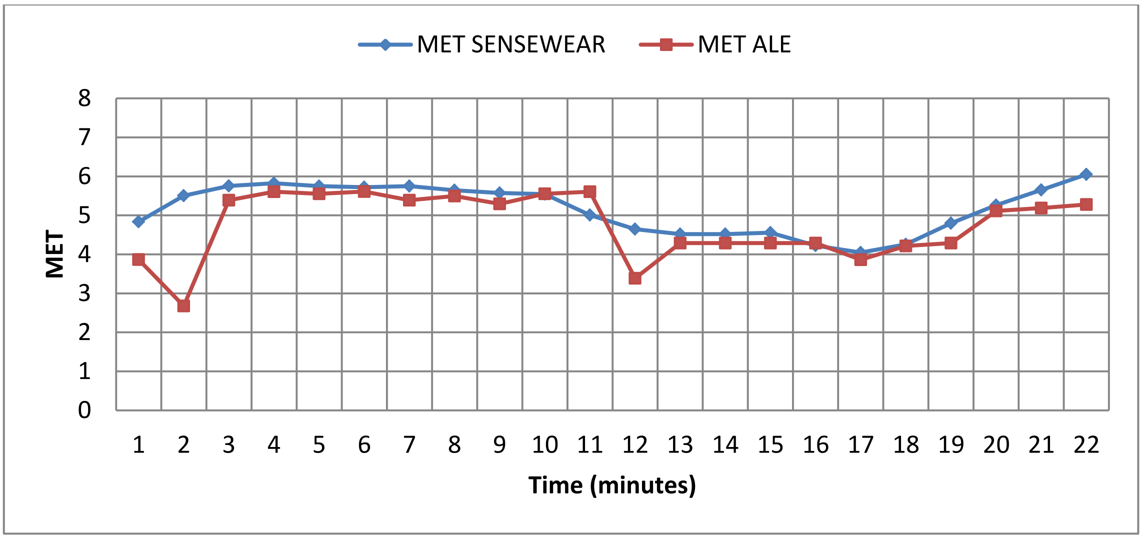


Fig. 37 Results from both device per minutes for the test 2

Results

Overall MAPE	8.98%	**Overall Proportion**
Overall overestimation	4.56%	13.64%
Overall underestimation	-9.68%	86.36%

Table 15 Results for the estimation error of ALE based on SenseWear for the test 2

Comments

Like the previous test, the user stopped walking twice while less a minute (Fig. 36, Fig. 37), at minutes 2 and 12. User from the given test walked on same time and same speed as the user of the test 1.

For this test, results (Table 14, Table 15) show in general an underestimation. If we remove cases of stop walking describe above, the trends of ALE follows Sensewear but always with a lower values.

On the previous test, we explained why ALE underestimate MET when the user decrease his walk speed. We found the same situation for the given test but this time, the walk activity was on the right thresholds. The user for the test 2 was the smallest subject of all. We explained on the section 5.1.5.3 that the variable height had influence on the vector of acceleration. We thought that it was the reason of these lower results. Weight had also influence on this case. On following test 4 and 5, we will explain more about these influences.

Test 3

A 15 minutes walk on street road with a long and high hill. The user walked down and up this hill and then walk slowly on a parking. The user wore SenseWear on the left arm and ALE on the left pocket's Jeans. He wore standard shoes.

	MET	MET ADAPTED		Subject data	
Sedentary	1	0.95		**Weight**	65
Very low activity	2.5	2.38		**Height**	178
Low activity	4.5	4.28		**Age**	31
Moderate activity	6	5.70		**Gender**	Male
Intense activity	9	8.56		**RMR kcal/day**	1640.19
				RMR kcal-kg^{-1}-h^{-1}	1.05

Table 16 MET values and user characteristics for the test 3

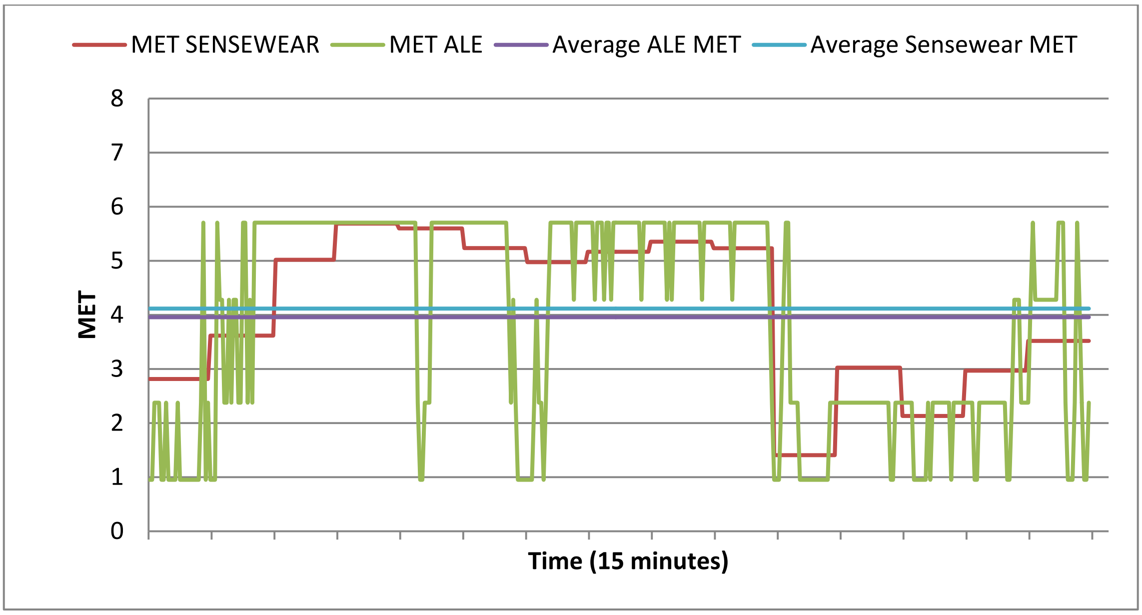


Fig. 38 Representation of ALE and SenseWear data for test 3

	Sensewear	ALE	Difference	MAPE
Average MET	4.12	3.96	-0.16	-3.85%

Table 17 Difference in percent for the overall results of ALE and SenseWear for the test 3

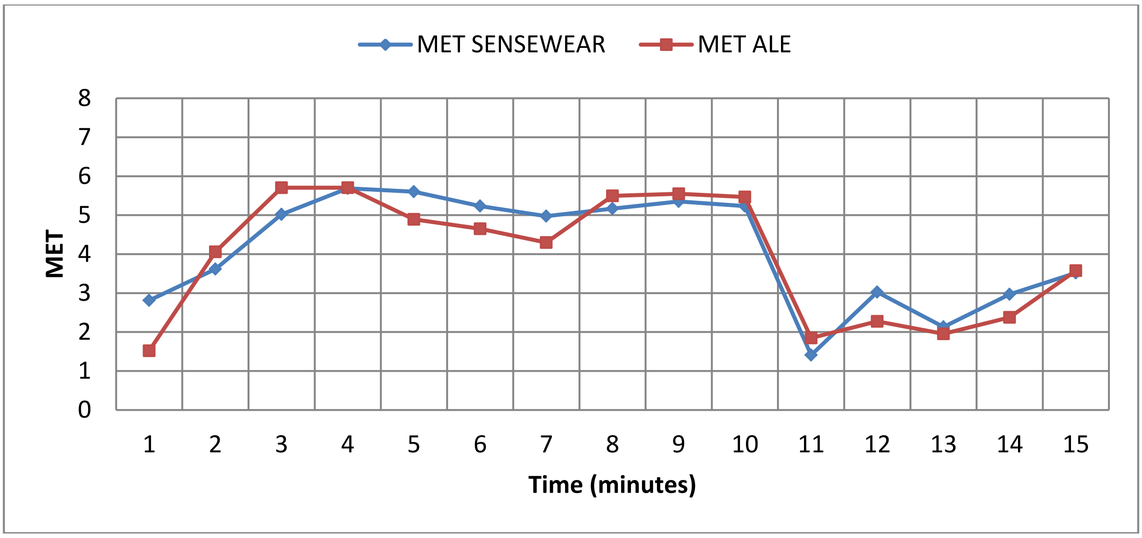


Fig. 39 Results from both device per minutes for the test 3

Results

Overall MAPE	14.02%	**Overall Proportion**
Overall overestimation	9.23%	53.33%
Overall underestimation	-21.36%	46.67%

Table 18 Results for the estimation error of ALE based on SenseWear for the test 3

Comments

For the given test, the user stopped walking twice, at the minute 5 and 7 (Fig. 38, Fig. 39) before the slow walking part (this part started at the minute 11). Before the first stop, ALE overestimates the MET when the user walked down the hill but we observed also that ALE overestimates when the user walked up the hill (from the minute 8). The user walked with disjointed movement during these to this case and that explained why ALE overestimates.

On the Table 17, we found an MET underestimation of 3.85%. We explained that with the two stops but also with the slow walking part. For this case, signals were often too low and were under the first threshold.

Finally, we found that the overall MAPE equal 14% (Table 18) and represented special case explained above. The proportion of over / underestimation were quit similar and explains that the very low results on the Table 17.

Test 4

A 24 minutes walk on street road with a long and high hill. The user walked down, then she walked on a flat road and finally went back and climbs the hill. In the middle of the test, the user waited on place while 5 minutes. The user wore SenseWear on the right arm and ALE on the right pocket's Jeans. She wore standard shoes.

	MET	MET ADAPTED		Subject data	
Sedentary	1	1.12		**Weight**	73
Very low activity	2.5	2.79		**Height**	175
Low activity	4.5	5.02		**Age**	23
Moderate activity	6	6.70		**Gender**	Female
Intense activity	9	10.05		**RMR kcal**	1568.99
				RMR kcal-kg^{-1}-h^{-1}	0.9

Table 19 MET values and user characteristics for the test 4

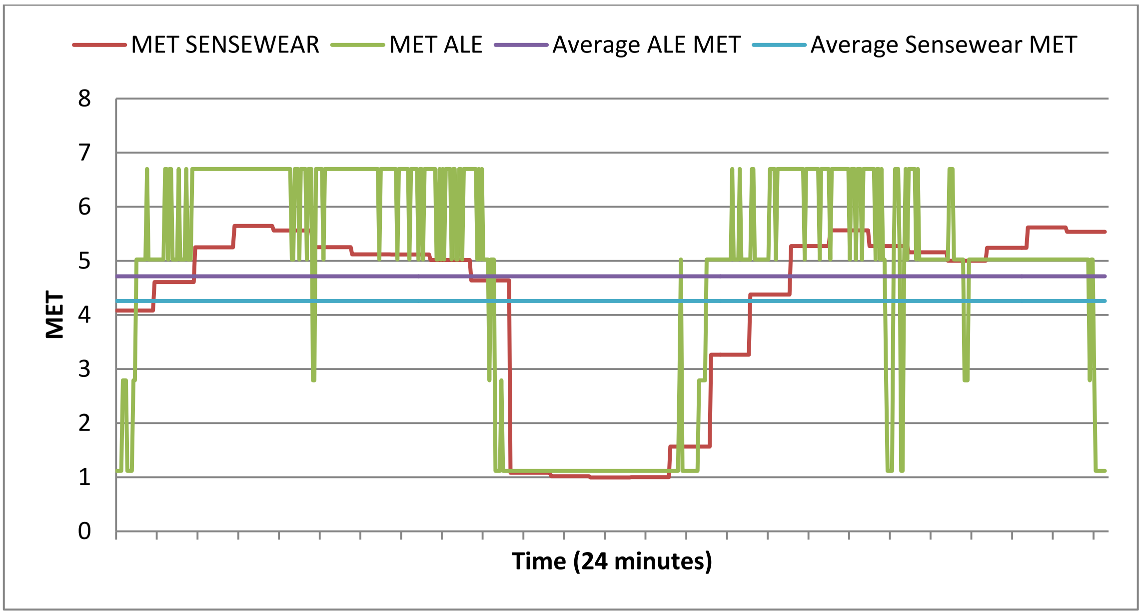


Fig. 40 Representation of ALE and SenseWear data for test 4

	Sensewear	ALE	Difference	MAPE
Average MET	4.26	4.5	0.24	5.66%

Table 20 Difference in percent for the overall results of ALE and SenseWear for the test 4

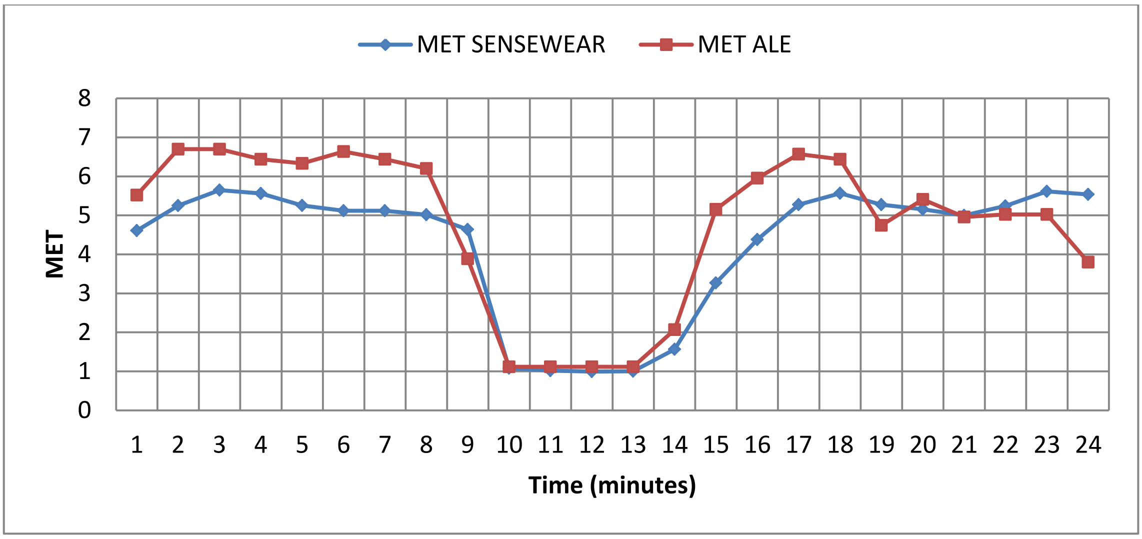


Fig. 41 Results from both device per minutes for the test 4

Results

Overall MAPE	19.27%	**Overall Proportion**
Overall overestimation	16.99%	75.00%
Overall underestimation	-15.50%	25.00%

Table 21 Results for the estimation error of ALE based on SenseWear for the test 4

Comments

On (Fig. 40, Fig. 41) we distinguished multiple parts: down the hill, walk on flat street, wait on place, walk on flat street and up the hill. We observed on the first and the third part an overestimation. Explanations were same as tests above but for the given test we had also issues from weight and height. Signals were very near the highest threshold. Weight and height for this case increase the signal and gave a strong evaluation of the walk. The overall MAPE (Table 21) reflected issues of the overestimation. Finally, we found on that the average error (Table 20) was not too high and calories computed after was near to SenseWear with 131 kcal for SenseWear and 125 for ALE.

Test 5

Same walk as the test 4. The user wore SenseWear on the left arm and ALE on the right pocket's Jeans. He wore standard shoes.

	MET	MET ADAPTED		Subject data	
Sedentary	1	1.05		**Weight**	87
Very low activity	2.5	2.62		**Height**	183
Low activity	4.5	4.71		**Age**	27
Moderate activity	6	6.28		**Gender**	Male
Intense activity	9	9.42		**RMR kcal**	1994.73
				RMR kcal-kg^{-1}-h^{-1}	0.96

Table 22 MET values and user characteristics for the test 5

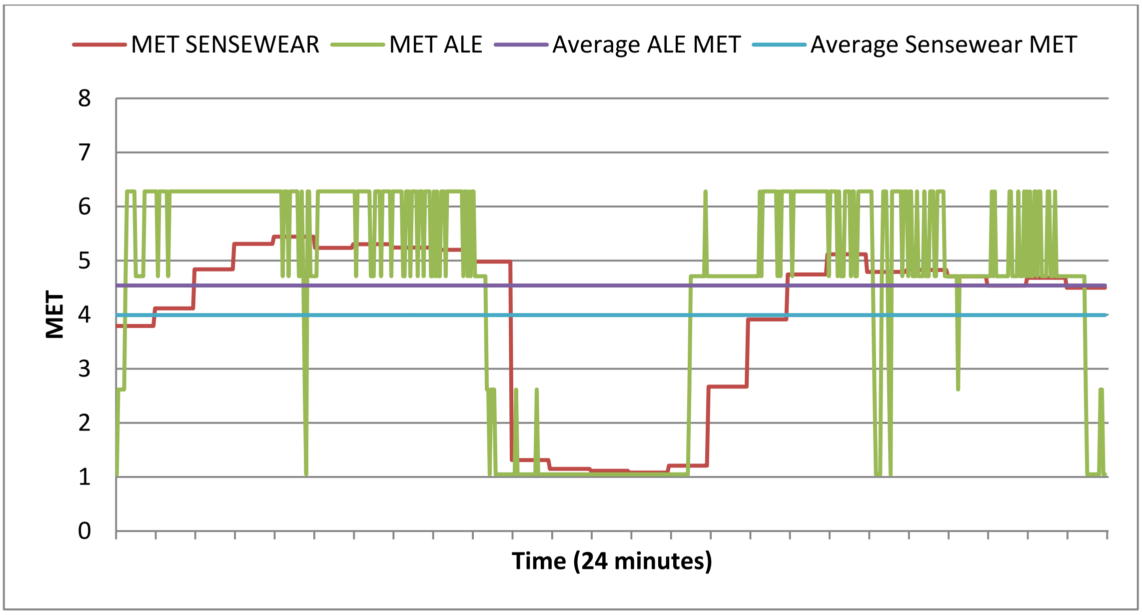


Fig. 42 Representation of ALE and SenseWear data for test 5

	Sensewear	ALE	Difference	MAPE
Average MET	4.0	4.54	0.56	13.70%

Table 23 Difference in percent for the overall results of ALE and SenseWear for the test 5

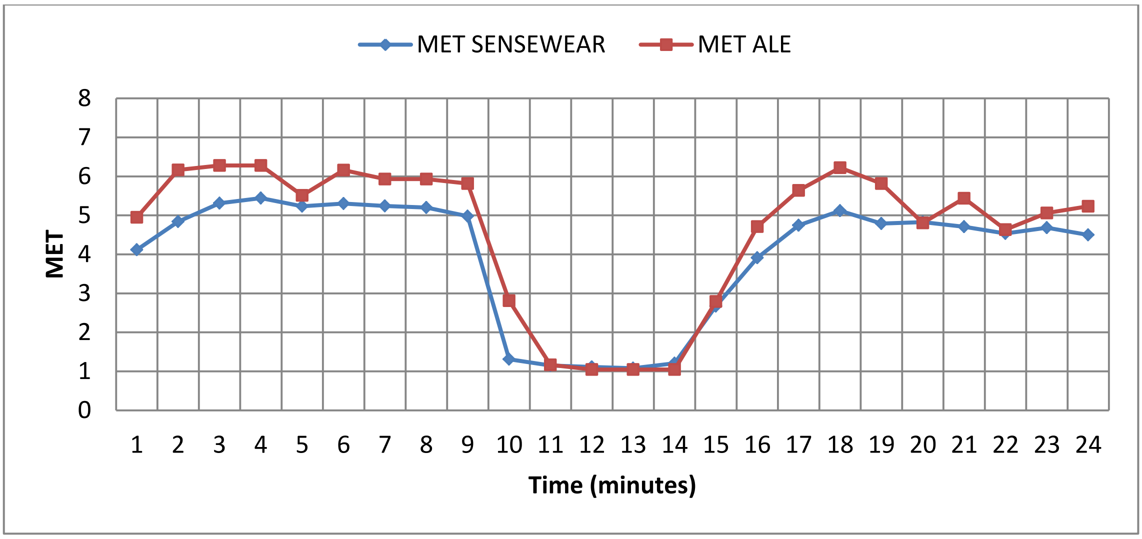


Fig. 43 Results from both device per minutes for the test 5

Results

Overall MAPE	17.27%	**Overall Proportion**
Overall overestimation	19.58%	83.33%
Overall underestimation	-5.72%	16.67%

Table 24 Results for the estimation error of ALE based on SenseWear for the test 5

Comments

For the given test we observed same parts as the previous one (Fig. 42, Fig. 43). Again, ALE overestimated MET. We assumed also that weight and height of the user influenced a lot results found in the Table 23 because a lot of time signals from accelerometer were near the highest threshold. We observed that overall MAPE (Table 24) were lower than test 4 and was near to others tests.

Test 6

A 42 minutes walk on a road forest with small hills. The road was not flat with a lot of hole and stone. At different moment, the user stopped walking because he took his dog for a walk. He wore SenseWear on the left arm and ALE on the left pocket's Jeans.

	MET	MET ADAPTED		Subject data	
Sedentary	1	0.93511494		**Weight**	68
Very low activity	2.5	2.33778735		**Height**	184
Low activity	4.5	4.20801724		**Age**	26
Moderate activity	6	5.61068965		**Gender**	Male
Intense activity	9	8.41603447		**RMR kcal/day**	1745.24
				RMR kcal-kg^{-1}-h^{-1}	1.07

Table 25 MET values and user characteristics for the test 6

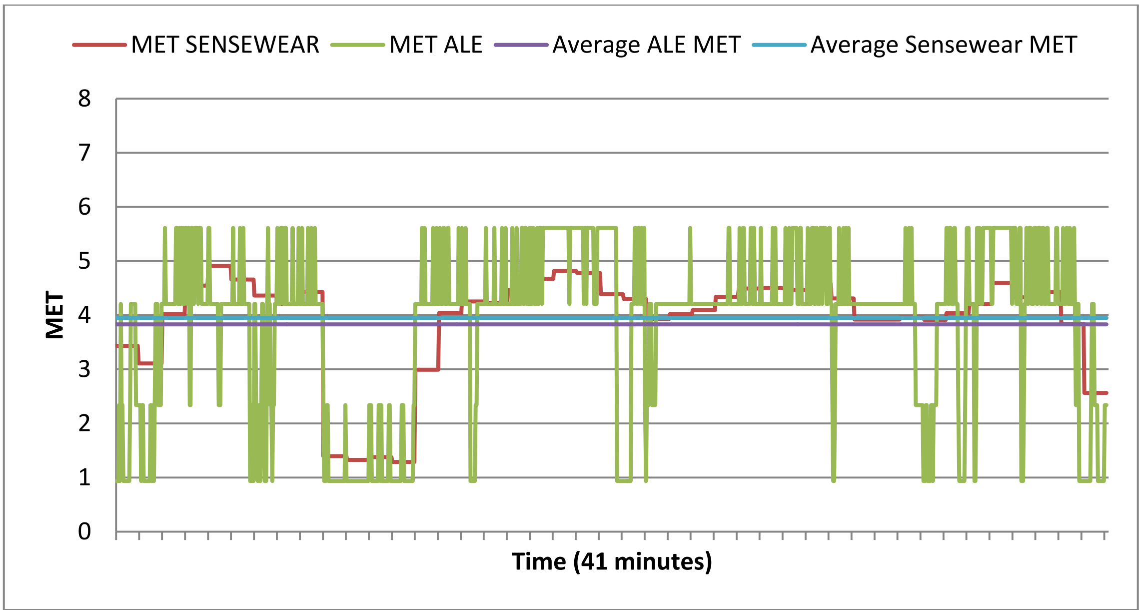


Fig. 44 Representation of ALE and SenseWear data for test 6

	Sensewear	ALE	Difference	MAPE
Average MET	3.95	3.83	-0.12	-6.99%

Table 26 Difference in percent for the overall results of ALE and SenseWear for the test 6

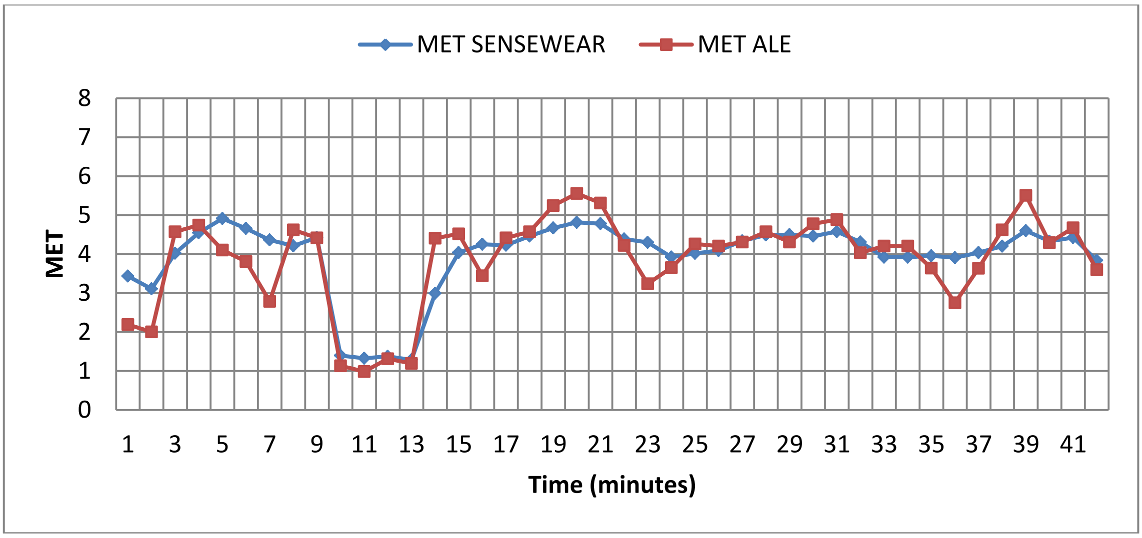


Fig. 45 Difference in percent for the overall results of ALE and SenseWear for the test 6

Results

Overall MAPE	12.52%	**Overall Proportion**
Overall overestimation	10.36%	47.62%
Overall underestimation	-14.48%	52.38%

Table 27 Results for the estimation error of ALE based on SenseWear for the test 6

Comments

On the Fig. 44 we observed a lot of stops from the user. These stops gave errors on the Fig. 45 and increased the underestimation on the Table 27. The given test was very interesting because the ground was never tested before and we did not know if ALE was able to compensate disjointed movements. Finally, we found an underestimation of 7% for the MET average.

Test 7

Same walk as test 6 on same condition. He wore SenseWear on the left arm and ALE on the left pocket's Jeans.

	MET	MET ADAPTED		Subject data	
Sedentary	1	0.95		**Weight**	65
Very low activity	2.5	2.38		**Height**	178
Low activity	4.5	4.28		**Age**	31
Moderate activity	6	5.70		**Gender**	Male
Intense activity	9	8.56		**RMR kcal/day**	1640.19
				RMR kcal-kg^{-1}-h^{-1}	1.05

Table 28 MET values and user characteristics for the test 7

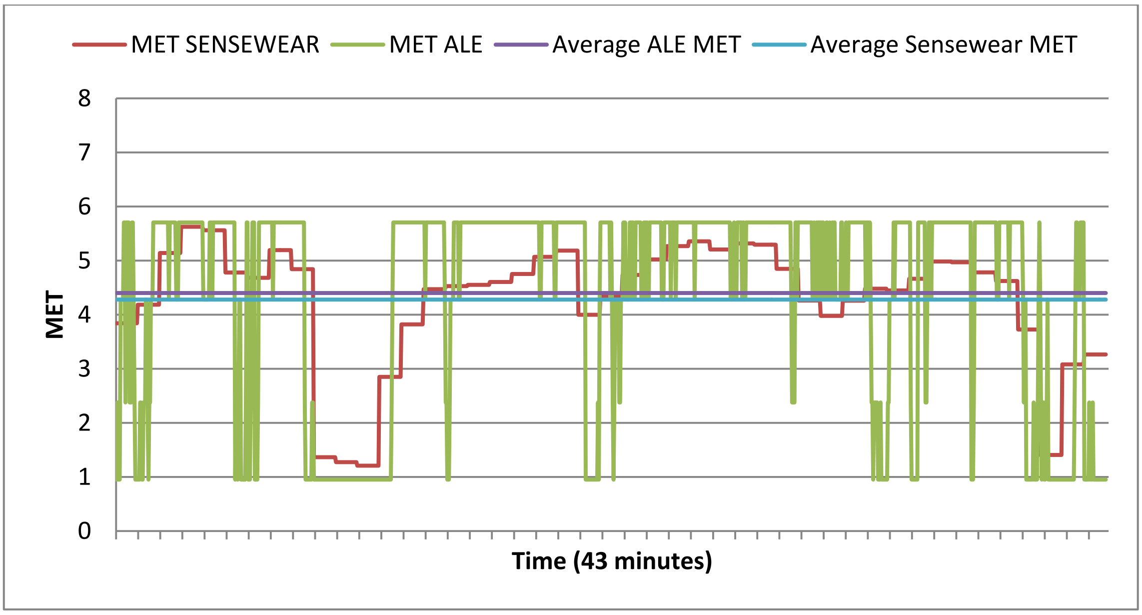


Fig. 46 Representation of ALE and SenseWear data for test 7

	Sensewear	ALE	Difference	MAPE
Average MET	4.28	4.40	0.12	2.86%

Table 29 Difference in percent for the overall results of ALE and SenseWear for the test 7

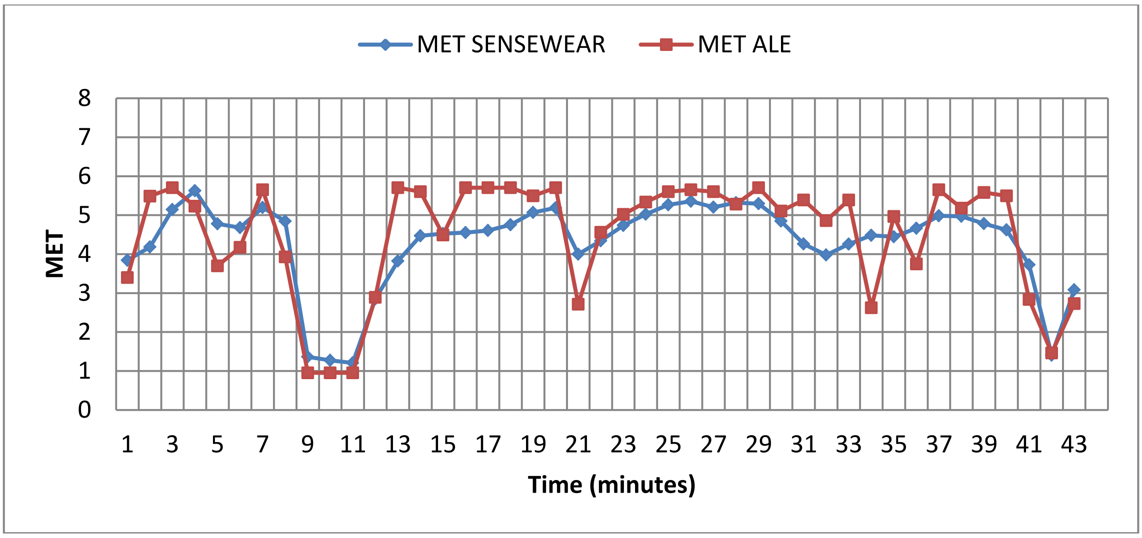


Fig. 47 Results from both device per minutes for the test 7

Results

Overall MAPE	15.97%	**Overall Proportion**
Overall overestimation	14.60%	65.12%
Overall underestimation	-18.52%	34.88%

Table 30 Results for the estimation error of ALE based on SenseWear for the test 7

Comments

The test 7 had similar results as the test 6 (it was the same walk on the same moment). But for this case, we observed ALE overestimated more MET (Fig. 47, Table 30) but had a standard MAPE. The overestimation was often cause by signals near a high threshold. Results of MET average on the Table 29 were close to SenseWear.

Test N°	Period (minute)	Average MET SenseWear	Average MET ALE	Difference	MAPE	Average Overestimation error	Average Underestimation error	Proportion in percent of overestimation	Proportion in percent of underestimation
1	22	5.12	5.35	4.47%	11.98%	14.26%	-8.44%	60.87%	39.13%
2	23	5.12	4.77	6.77%	8.98%	4.56%	-9.68%	13.64%	86.36%
3	15	4.12	3.96	3.85%	14.02%	9.23%	-21.36%	53.33%	46.67%
4	25	4.26	4.72	10.75%	19.27%	16.99%	-15.50%	75.00%	25.00%
5	25	3.99	4.54	13.70%	17.27%	19.58%	-5.72%	83.33%	16.67%
6	42	3.95	3.83	3.01%	12.52%	10.36%	-14.48%	47.62%	52.38%
7	44	4.28	4.40	2.86%	15.97%	14.60%	-18.52%	65.12%	34.88%
Average		4.40	4.51	6.49%	14.29%	12.80%	-13.38%	56.99%	43.01%

Table 31 Results for all short duration tests

The Table 31 resumed all results from previous tests. With the MAPE average, we assumed that ALE was accurate at 14% for walking activities. It was not possible with these results to know if ALE over / underestimate because the trends proportion was very similar with an overestimation proportion of 57% and an underestimation proportion of 43% and also that it depended on the user.

6.2 Long term test results

Because it would be a real case study of ALE long term estimation on how ALE was accurate and also determined the trends of over / underestimating, we conducted a long test along three days for a total of about 30 hours. The test was separate in three parts: 6 hours, 15 hours and 9 hours. We asked to the user to live normally and to disconnect both devices when he went sleeping.

We presented results of this test with three figures (Fig. 48, Fig. 49, Fig. 50) that represented the three parts and also comments into figures that represented some activities of the user, like driving, shopping and working. These graphs represented MET per minute for both devices. After figures, we presented results (Table 32) and our comments about this test.

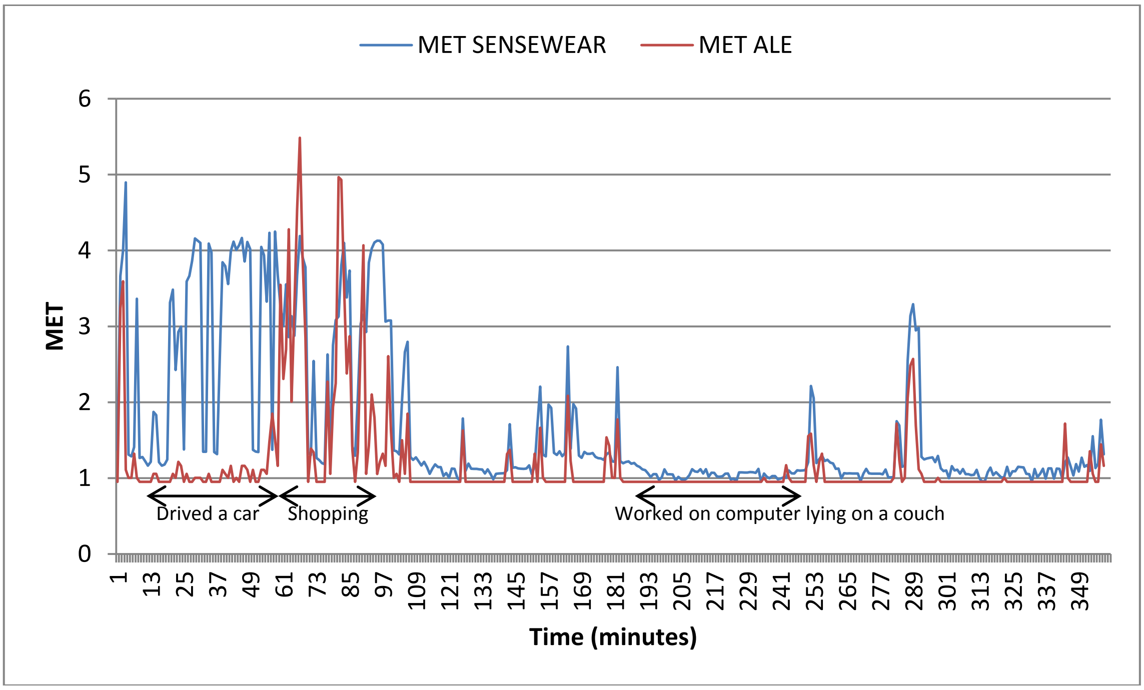


Fig. 48 MET per minute for both devices for the part 1 (6h00) of the long test

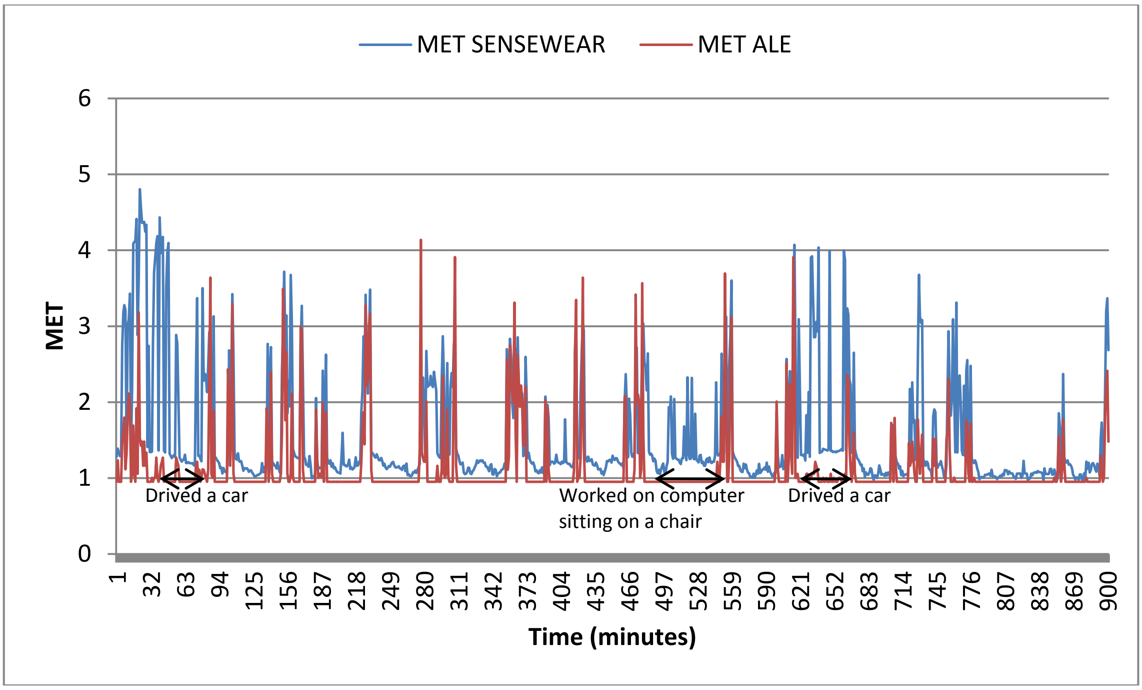


Fig. 49 MET per minute for both devices for the part 2 (15h00) of the long test

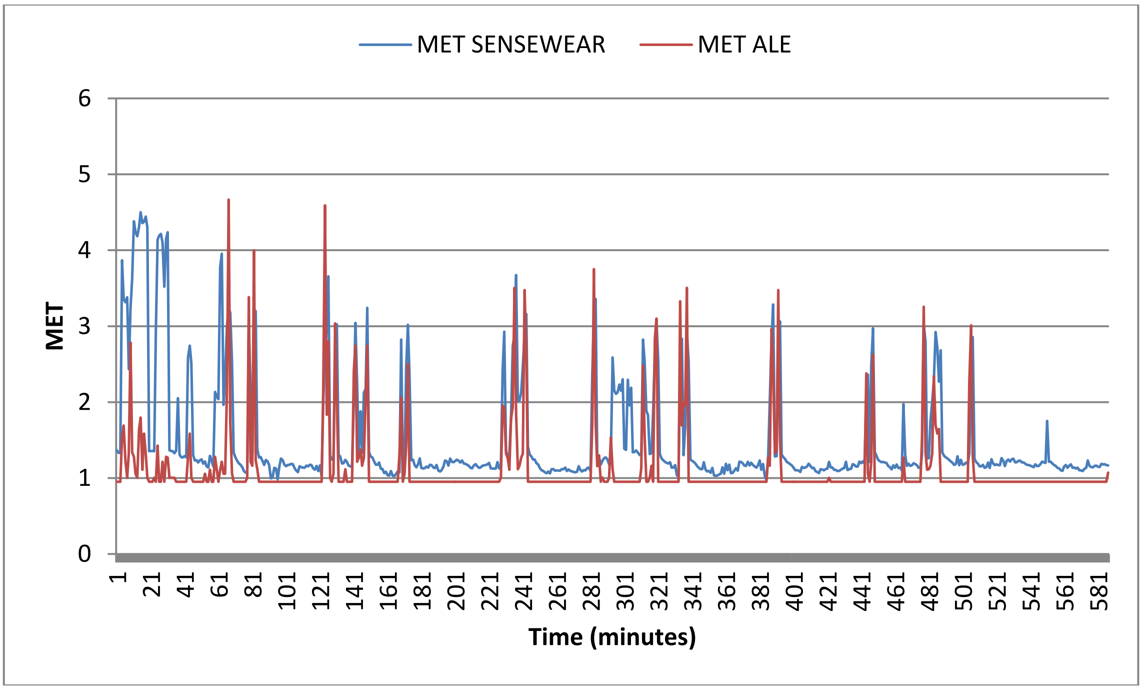


Fig. 50 MET per minute for both devices for the part 3 (9h40) of the long test

Part N°	Period (minute)	Average MET SenseWear	Average MET ALE	Difference	MAPE	Average Overestimation error	Average Underestimation error	Proportion in percent of overestimation	Proportion in percent of underestimation
1	6h00	1.69	1.19	-29.78%	23.18%	21.65%	-23.27%	5.87%	94.13%
2	15h00	1.56	1.13	-27.54%	23.68%	22.16%	-23.74%	3.88%	96.12%
3	9h40	1.53	1.13	-26.37%	23.36%	19.78%	-23.55%	5.11%	94.89%
Average		1.60	1.15	-27.90%	23.41%	21.20%	-23.52%	4.95%	95.05%

Table 32 Complete results for the long test separate into three parts

Comments

On the Table 32 we observed that ALE had an absolute error per minute (MAPE) higher than previous tests on the Table 31 and a trends to underestimate MET with a difference of 28% for a proportion of 95%. That mean that ALE was not able to detect all kind of activities like SenseWear but based on previous results with 7 users, we assume that ALE was enough accurate for walking activities. On Fig. 48 and Fig. 49 we represented different activities made by the user during the day. By example on the Fig. 48, we saw that in the beginning of the test, the user was driving. These information were provided by the user at the end of the test. SenseWear detected during this activity a MET form 1.2 to 4, but on the other hand ALE detected only MET from 1 to 1.2. ALE was not design to detect this kind of activities and we didn't know how SenseWear was able to do that. Other example on the Fig. 49, we noted a part of the day when the user worked on a chair. Same situation, SenseWear detected a higher MET than ALE. Again ALE was not able to detect that because, except arm, the user didn't move. On the Fig. 48, the shopping part was detected by ALE and results near than ones from SenseWear. Again on the Fig. 48, we observed a long part when the user was lying on a couch and worked on computer. Here, SenseWear detected a MET about 1.2 and ALE a standard MET (1).

All these examples explained why SenseWear had higher results for MET. As we tested walk activities, we knew that these parts were accurate but for others activities, ALE had no possibilities to detect the energy expenditure like SenseWear based on galvanic skin response and temperature. Finally, we assume that the average of 23% error per minute was a very good score if we take into account case explained above. At the end, ALE underestimated calories by 23% and with think it stayed closer representing the reality.

7. Discussion and future work

Our experimentation about activity level evaluation is a part of an ongoing European research project called TRAINUTRI. For this project, in this thesis, we tried to experiments the possibility to have activity recognition with the help of a mobile phone. Our application ALE is still a prototype and some functionalities are missing but some part can also be reuses for the project.

7.1 About ALE experimentation

ALE is still a prototype. It is not possible yet to share it on the Android Market. We have to experiment more and conduct more user studies. In the previous section we made a comparison with another device which is noted very accurate, but we didn't conduct this study during enough time and with enough subjects. Our results could not prove the real percent of ALE accuracy. It is why we talk about this evaluation as an application's validation. To determine the accuracy, with respect to available resources, we defined that we need to test our application during minimum 2 consecutive weeks on minimum 15 users. For our research, we didn't the possibility to conduct this kind of study because we didn't have enough devices, SenseWear, and mobile phones. Moreover, this kind of study takes a lot of time that we didn't had.

During our experimentation we observe that batteries of the mobile phone will be issues for the usability of ALE. In fact, we measure the maximum time to use ALE continuously on a mobile phone. We did the test with two mobiles phone: a HTC Desire and a HTC G1. For this test, we deactivate Wifi and 3G connection. For both mobile phones, the battery was full when we started the study. We measure that the HTC G1 was active during 7h30 min and the HTC Desire during 8h30 min. On the Android OS there is an application that determined which application drains the batteries. For both cases, it indicated ALE being responsible for about 70% of battery drain. We searched if it was possible to change our design or implementation to improve the battery life. Unfortunately, it is mainly dependent of sensors. Without the use of it, ALE does not influence much the battery, except for the sleep mode. The only possibility offered by the API was to decrease the frequency of sensors but for our experimentation, it was not possible as we explained. For the sleep mode, the Android OS have a protection for the user side, not for the developer side. In fact, when the user locks his screen, normally, it automatically turns off but the system will also turn off all sensors. To keep alive these sensors, we did not have the choice to keep the screen on. This protection is to alert users of drain battery when sensors are used, but we needed to use it, to assume our application is running. Given this results of this test were not really a surprise but a veritable issue for our user test. Our potential users must to think all time to charge their mobile phone during the day.

About the activity level estimation, a big part of our research was based on empirical experimentation instead scientific approaches. Our threshold validation was made by a user study of 15 subjects that we defend enough for prototype feasibility but not for a scientific validation. Again, 15 subjects were not enough for our conclusion. We needed to have more subjects but also special conditions for the user test. In first, we think that it will be very interesting to conduct a parallel study on a treadmill to measure movement on the same speed for each user. Secondly, we think that it will be also necessary to ask users to wear same clothes and shoes to remove these variables on our evaluation. Not for forget these variables, but to have better results to observe the influence of weight and height variables on ALE results.

In the beginning of our experimentation, we wanted to add the functionality to send activity results on a web server and also the possibility to share these information with other users e.g. active peers. This part is actually missing but will be experimented following this research. We want to experiment with this functionality to see how the fact to share results with other users will improve their motivation for activity. Test results from (Sunny, Katherine et al. 2006; Anderson, Maitland et al. 2007) prove that the data sharing can be a source of motivation for users. We found also a lot of web pages and mobile phone application that manage daily activity of the user with the peer sharing functionality. We wish to do same for ALE.

The GUI of ALE was made in function of our needs. We didn't conduct study on human-computer interaction, to measure usability of our GUI. We think it is a very important part for every software development, because our vision of the application is not often the same as for users. For the future part of our research, we want to conduct a real user study with appropriate approaches for this study. In first, we plan to interviews with different users and ask them to interact with our system but with a paper prototype (mock-ups). Secondly, built the GUI depending results for the previous test and conduct a new one with new subjects. This approach could be very useful for the European project because target subjects are senior people and we have to analyze how these kinds of users interact with computer systems.

7.2 TRAINUTRI project

TRAINUTRI (training and nutrition) is a European project based on the concept of ambient assisted living. The main goal is to help people (seniors between 50 and 65 years) to develop healthy habits with the help of Information and Communication Technology (ICT). The project is composed of three parts:

- A peer sharing networks, based on a web platform
- Evaluation of eating habits and sensible nutrition
- Physical activity recognition and evaluation

For the evaluation of eating habits and also physical activity recognition, mobile phone and computer systems will be used. The main idea is to create a complete platform that combines these 3 parts. Our experimentation of ALE is a part of the physical activity recognition.

The following scenario of Jeanne (Section 3) combined the activity recognition part and the social network part. It was an example of a future work based on the TRAINUTRI project. In this scenario we called the application TRAINI because we remove the nutrition part.

User scenario

Robert works in a corporation for import/export as manager. Jeanne, his wife, goes into retirement this month. She was a teacher at primary school. Jeanne and Robert are 62 years old. They live in Geneva centre in a small flat. They don't need more space because their children have left and live separately in their own flats, also situated in Geneva.

Jeanne is happy to have now a lot of free time, but she doesn't know what to do with his journey. Usually she goes to work and after, she takes a rest at home with Robert. Both are not very active, and stay usually at home. Now, the days for Jeanne are very long. She wants to do some activities, like hobbies. A friend of Jeanne, Sylvie, is retired since two years, and she is very active. Sylvie talks to Jeanne and propose her a new system called TRAINI based on a mobile phone. Jeanne is a bit afraid because she doesn't know how to use this kind of technology. She knows how to use a mobile phone (just to make a call), and how to navigate to Internet, but I many cases she ask for help her children.

Sylvie shows her how to use this system and its options. She helps Jeanne to install, create an account and configure the system. Jeanne is interested about this software and think, if Sylvie can do that, I can also do that! But Jeanne wants to start slowly and don't want to learn too much. So Sylvie teaches Jeanne how to use basic functionalities. The first one is how to calculate the percentage of activities per days and shows to Jeanne that the mobile phone is able to calculate the number of steps and detect when she seats, when she is on a car and when she is really walking. Jeanne doesn't understand how it is possible, but she finally she doesn't care, it is fun!

While a week passed by, Jeanne goes to her mobile phone every day to watch how many steps she does per day. She discovers by herself how to create some challenge directly on the mobile phone. She decides to fix a goal per day and try to reach this goal. She is very motivated and every day, goes out to walk. She thinks that it is fun and the system very motivating because, if she reaches her goal of the day, the system congratulates her with nice flower box on screen. Even she doesn't reach the goal, the system gives her some motivation hints for the next day.

Jeanne wants to use the system amore and asks again Sylvie what more is possible to do with this system. Sylvie shows her how to consult a list of possible outdoor or cultural activities happening in her local community. Jeanne enters some parameter like the kind of activities she wants to do and her interest. Automatically the system proposes to Jeanne a list of different activities, classified by categories, from training activities to cultural activities. Jeanne sees that there is a new exposition to a museum and decide to go today. She also finds a lot of information, like the schedule, the price and some previews of the exposition.

After some weeks, Jeanne sees that it is possible to find some known friends in the system and also the possibility to have a connection with. She asks for a connection with her friend Sylvie. After a confirmation from Sylvie, she sees all goals from Sylvie and also her percentage of activities. Jeanne thinks that she needs to do more herself because the percent of activities from Sylvie is higher. She also sees the possibility to share an agenda and also to recommend some activities. The system proposes also some group activities with people that Jeanne doesn't know and she decides to try one of these group activities with Sylvie.

Now, Jeanne is very active and does a lot of activities every week. Her husband has never tried this system so Jeanne decides to teach to Robert. At first, Robert is not interested. For him, this kind of system is for young people. He doesn't need that! But Jeanne tries again and Robert decides to try some outdoor activities. He meets some new friends that have same hobbies. After some weeks, Robert is in a hurry to go on retirement to have more time to do a lot of activities with Jeanne and new friends he made thanks for the TRAINI.

8. Conclusions

This section presents our conclusions and answers to the hypothesis defined in the section 1. Then we give an overall conclusion about the complete research and finally a personal conclusion from the author.

In the section 1, we defined three criteria as hypothesis research. We answered to the first and to the second criteria with scientific research that we found. The third hypothesis was answered by our complete experimentation with the development of ALE.

1. *"If we deploy ALE on mobile phone, it improves usability by being the least obtrusive to the users, by the way of an all-in-one device like a smartphone and without any external device. It is possible to find external devices that detect physical activities, but most of them should be connected on a system to compute results of the day."*

2. "*If we deploy ALE on mobile phone, it will improve "compliance" that user will use it. Actually, most people wear a mobile phone during the day and the fact to have at proximity their mobile will improve the use of ALE. The trend of research shows that users wore more time a mobile phone.*"

In the section 2.3.3, we showed that a lot of people owned a cell phone. The trends showed that this kind of device were a lot of time at proximity and often directly on the user pocket. Actually, we cannot prove those users will more use ALE based on the fact that the mobile phone is always on a pocket during the day. In the section 2.3.4, we explained that it was a good opportunity for our experimentation because the user had not to wear two devices during a long period of time to measure the physical activity level.

3. *"With the off-the-shelf mobile platforms we can develop activity level estimator at least as accurate as a dedicated system. It is important that ALE system is as accurate as external device built especially for this application."*

We developed an activity level estimator deployed on a mobile phone. We tested and compared our system with SenseWear from BodyMedia which was considered very accurate by professionals. Our results showed that an overall error per minute of 23.41% during a complete day and an error per minute of 14.29% for walking activities. For this kind of activities, ALE showed at the end an overall difference with SenseWear of 6.49% for the calories estimation with an overestimation max at 13.7% and an underestimation max at 6.77%. About the complete day, the difference was higher with an underestimation of 27.9%. We explained on the section 6 why we conclude that ALE was accurate. The device SenseWear detected activities when the user didn't move, like worked sit on a chair. ALE was designed to detect body movement and was not able to detect this kind of activities.

Finally, we conclude that ALE is accurate despite the average 23.41% of error per minutes because we achieved it only with body movement detection. We knew that Sensewear used other sensors, like galvanic skin response, temperature and it was impossible to do same with a mobile phone. We assume that user tests were not enough complete to validate totally our systems but showed a trend that ALE was on the right direction to be later a very accurate system. About the estimation of calories burned by the user, we had finally an underestimation of 27.9% and we think that it better to underestimate calories than overestimate for the user, such the user won't eat too much.

As personal conclusion, I think it was a very good experience and especially the part to analyze body movement. I learned during my study how to create a system with a direct interaction with the user but only with minimal interface. For this project, I had to learn not only how to interact with the user but also with his body. I discovered how a human walk and how analyzed this movement with a signal. After many tests, I am able to detect on a graph walking part and also intensities!

9. References

Ainsworth, B. E. (2002). "The Compendium of Physical Activities Tracking Guide." University of South Carolina. Retrieved 27.09.2010, from http://prevention.sph.sc.edu/tools/compendium.htm.

Ainsworth, B. E., W. L. Haskell, et al. (2000). "Compendium of physical activities: an update of activity codes and MET intensities." Medicine & Science in Sports & Exercise **32**(9): S498.

Akay, M., M. Sekine, et al. (2003). "Unconstrained monitoring of body motion during walking." IEEE Engineering in Medicine and Biology Magazine **22**(3): 104-109.

Anderson, I., J. Maitland, et al. (2007). "Shakra: Tracking and Sharing Daily Activity Levels with Unaugmented Mobile Phones." Mobile Networks and Applications **12**(2): 185.

Android.com. (2010). "Designing a Remote Interface Using AIDL." Retrieved 10.11.2010, from http://developer.android.com/guide/developing/tools/aidl.html.

Android.com. (2010). "SensorManager." Retrieved 10.11.2010, from http://developer.android.com/reference/android/hardware/SensorManager.html.

Apple Inc. (2010). "Apple - iPhone iOS4." Retrieved 22.11.2010, from http://www.apple.com/iphone/ios4/.

Bagi, L. (2009). "pedometer." Retrieved 09.12.2010, from http://code.google.com/p/pedometer/.

Bagi, L. (2010). "android-pedometer." Retrieved 09.12.2010, from https://github.com/bagilevi/android-pedometer.

Barralon, P. (2005). "Classification et fusion de données actimétriques pour la télévigilance médicale: élaboration et validation expérimentale chez la personne jeune et âgée."

Barralon, P., N. Vuillerme, et al. (2006). Walk detection with a kinematic sensor: Frequency and wavelet comparison. Proceedings of the 28th IEEE EMBS Annual International Conference, New York City, USA.

Bouchard, C., S. Blair, et al. (2007). Physical activity and health, Human Kinetics Publishers.

Byrne, N. M., A. P. Hills, et al. (2005). "Metabolic equivalent: one size does not fit all." Journal of Applied Physiology **99**(3): 1112.

caloriesperhour. (2007). "Calculating BMR and RMR." from http://www.caloriesperhour.com/tutorial_BMR.php.

Connelly, K. H., A. M. Faber, et al. (2006). "Mobile applications that empower people to monitor their personal health." e & i Elektrotechnik und Informationstechnik **123**(4): 124.

Crouter, S., P. Schneider, et al. (2003). "Validity of 10 electronic pedometers for measuring steps, distance, and energy cost." Medicine & Science in Sports & Exercise **35**(8): 1455.

Division of Nutrition & Physical Activity (1999). Promoting Physical Activity: A Guide for Community Action. IL: Human Kinetics. U.S. Department of Health & Human Services, Public Health Service, Center for Disease Control & Prevention, National Center for Chronic Disease Prevention & Health Promotion and D. o. N. P. Activity. Champaign.

Europa Summaries of EU legislation. (2008). "The demographic future of Europe - from challenge to opportunity." Retrieved 28.07.2010, from http://europa.eu/legislation_summaries/employment_and_social_policy/situation_in_europe/c10160_en.htm.

Fitbit. (2010). "Fitbit." Retrieved 08.11.2010, from http://www.fitbit.com/.

Foerster, F. and J. Fahrenberg (2000). "Motion pattern and posture: correctly assessed by calibrated accelerometers." Behavior Research Methods Instruments and Computers **32**(3): 450-457.

Fogg, B. J. (2002). Persuasive Technology: Using Computers to Change What We Think and Do (Interactive Technologies), Morgan Kaufmann.

Hatano, Y. (1993). "Use of the pedometer for promoting daily walking exercise." International Council for Health, Physical Education, and Recreation **29**(4): 4-8.

Health-calc. (2006). "Total energy expenditure." Retrieved 29.09.2010, from http://www.health-calc.com/diet/energy-expenditure-advanced.

Instructables. (2010). "Accelerometer & Gyro Tutorial." Retrieved 18.10.2010, from http://www.instructables.com/id/Accelerometer-Gyro-Tutorial/.

Jakicic, J. "Measuring Energy Expenditure and Physical Activity: Review of Literature." Retrieved 04.11.2010, from http://pmbcii.psy.cmu.edu/core_c/physical_activity.html.

Jakicic, J., M. Marcus, et al. (2004). "Evaluation of the SenseWear Pro Armband (TM) to Assess Energy Expenditure during Exercise." Medicine & Science in Sports & Exercise **36**(5): 897.

Jequier, E. and Y. Schutz (1983). "Long-term measurements of energy expenditure in humans using a respiration chamber." American Journal of Clinical Nutrition **38**(6): 989.

Mathie, M. J., B. G. Celler, et al. (2004). "Classification of basic daily movements using a triaxial accelerometer." Medical and Biological Engineering and Computing **42**(5): 679-687.

Microsoft. (2010). "Window Phone home." Retrieved 22.11.2010, from http://www.microsoft.com/windowsphone/en-us/default.aspx.

Montoye, H., H. Kemper, et al. (1996). Measuring physical activity and energy expenditure, Human Kinetics Champaign, IL.

Mosby's Medical Dictionary (2009). indirect calorimetry. Mosby's Medical Dictionary, 8th edition.

Office fédéral de la statistique. (2010). "Ménages et population - Infrastructure téléphonique." 23.11.2010, from http://www.bfs.admin.ch/bfs/portal/fr/index/themen/16/04/key/approche_globale.indicator.30101.301.html?open=1,2#2.

Open Handset Alliance. (2010). "Open Handset Alliance." Retrieved 22.11.2020, from http://www.openhandsetalliance.com/index.html.

Patel, S., J. Kientz, et al. (2006). "Farther than you may think: An empirical investigation of the proximity of users to their mobile phones." UbiComp 2006: Ubiquitous Computing: 123-140.

Prochaska, J. O., C. C. DiClemente, et al. (1992). "In search of how people change. Applications to addictive behaviors." The American psychologist **47**(9): 1114.

Purcell, K., R. Entner, et al. (2010). "The Rise of Apps Culture." Retrieved 23.11.2010, from http://pewinternet.org/Reports/2010/The-Rise-of-Apps-Culture.aspx.

Research in Motion Limited. (2010). "RIM Company." Retrieved 22.11.2010, from http://www.rim.com/.

Schneider, P., S. Crouter, et al. (2004). "Pedometer measures of free-living physical activity: comparison of 13 models." Medicine & Science in Sports & Exercise **36**(2): 331.

Segen, J. (2006). Concise dictionary of modern medicine, McGraw-Hill.

SenseWear. (2010). "SenseWear." Retrieved 08.11.2010, from http://sensewear.bodymedia.com/.

Smith, A. (2010). "Americans and their gadgets." Retrieved 23.11.2010, from http://www.pewinternet.org/Reports/2010/Gadgets.aspx.

St-Onge, M., D. Mignault, et al. (2007). "Evaluation of a portable device to measure daily energy expenditure in free-living adults." American Journal of Clinical Nutrition **85**(3): 742.

Sunny, C., E. Katherine, et al. (2006). Design requirements for technologies that encourage physical activity. Proceedings of the SIGCHI conference on Human Factors in computing systems. Montreal, Quebec, Canada, ACM.

Symbian Foundation. (2010). "The Symbian Foundation Community Home." Retrieved 22.11.2010, from http://www.symbian.org/.

Tudor-Locke, C. (2002). "Taking Steps toward Increased Physical Activity: Using Pedometers To Measure and Motivate." President: 10.

Tudor-Locke, C., D. R. Bassett, et al. (2004). "A preliminary study of one year of pedometer self-monitoring." Annals of Behavioral Medicine **28**(3): 158-162.

Tudor-Locke, C. and J. D. R. Bassett (2004). "How many steps/day are enough?: Preliminary pedometer indices for public health." Sports Medicine **34**(1): 1-8.

Wang, N., E. Ambikairajah, et al. (2007). Accelerometry based classification of walking patterns using time-frequency analysis. Proceedings of the 29th Annual International Conference of the IEEE EMBS, Cité Internationale, Lyon, France.

Welk, G. J., J. J. McClain, et al. (2007). "Field Validation of the MTI Actigraph and BodyMedia Armband Monitor Using the IDEEA Monitor[ast]." Obesity **15**(4): 918-928.

Wikipedia - MAPE. (2010). "Mean absolute percentage error." Retrieved 24.11.2010, from http://en.wikipedia.org/wiki/Mean_absolute_percentage_error.

Wikipedia. (2010). "Accelerometer." Retrieved 18.10.2010, from http://en.wikipedia.org/wiki/Accelerometer.

Appendixes

Appendix A - User test results

Results for each user test. On the table, MET from ALE and Sensewear, Mean absolute percentage error (MAPE) and Mean percentage error (MPE).

Test user 1

Time (minutes)	MET SENSEWEAR	MET ALE	MAPE	MPE
1	3.498981953	4.554350411	30.16%	30.16%
2	4.540688038	2.989200728	34.17%	-34.17%
3	5.512909889	5.706655936	3.51%	3.51%
4	5.674608231	5.706655936	0.56%	0.56%
5	5.807585716	5.523750297	4.89%	-4.89%
6	5.825932026	5.706655936	2.05%	-2.05%
7	5.812787056	5.706655936	1.83%	-1.83%
8	5.829179287	5.706655936	2.10%	-2.10%
9	5.883692741	5.706655936	3.01%	-3.01%
10	5.788203239	5.706655936	1.41%	-1.41%
11	5.773475647	5.706655936	1.16%	-1.16%
12	4.810780048	3.593079664	25.31%	-25.31%
13	4.53063488	5.653816529	24.79%	24.79%
14	4.795581341	5.706655936	19.00%	19.00%
15	4.83974123	5.706655936	17.91%	17.91%
16	4.729931831	5.706655936	20.65%	20.65%
17	4.274610043	4.702707207	10.01%	10.01%
18	4.095354557	5.178261868	26.44%	26.44%
19	4.437164307	5.178261868	16.70%	16.70%
20	5.137434959	5.706655936	11.08%	11.08%
21	5.380709648	5.706655936	6.06%	6.06%
22	5.462630749	5.706655936	4.47%	4.47%
23	5.273062706	5.706655936	8.22%	8.22%
		Average MAPE	11.98%	**Proportion**
		Average overestimation	14.26%	60.87%
		Average underestimation	-8.44%	39.13%

Test user 2

Time (minutes)	MET SENSEWEAR	MET ALE	MAPE	MPE
1	4.833536625	3.866593366	20.00%	-20.00%
2	5.5029006	2.675462708	51.38%	-51.38%
3	5.754027367	5.387575591	6.37%	-6.37%
4	5.822590828	5.607476636	3.69%	-3.69%
5	5.750141144	5.552501374	3.44%	-3.44%
6	5.722838879	5.611548877	1.94%	-1.94%
7	5.749624252	5.387575591	6.30%	-6.30%
8	5.64215517	5.497526113	2.56%	-2.56%
9	5.575249195	5.293914035	5.05%	-5.05%
10	5.543589592	5.552501374	0.16%	0.16%
11	5.007956028	5.607476636	11.97%	11.97%
12	4.646980762	3.388104982	27.09%	-27.09%
13	4.520570278	4.288070368	5.14%	-5.14%
14	4.519931316	4.288070368	5.13%	-5.13%
15	4.55479908	4.288070368	5.86%	-5.86%
16	4.22210598	4.288070368	1.56%	1.56%
17	4.046005726	3.864557246	4.48%	-4.48%
18	4.257139206	4.21477002	1.00%	-1.00%
19	4.798994064	4.288070368	10.65%	-10.65%
20	5.260782242	5.112699285	2.81%	-2.81%
21	5.651985645	5.188035754	8.21%	-8.21%
22	6.050519466	5.277625069	12.77%	-12.77%
		Average MAPE	8.98%	**Proportion**
		Average overestimation	4.56%	13.64%
		Average underestimation	-9.68%	86.36%

Test user 3

Time (minutes)	MET SENSEWEAR	MET ALE	MAPE	MPE
1	2.813647747	1.518116804	46.04%	-46.04%
2	3.615628481	4.059198717	12.27%	12.27%
3	5.019596577	5.706655936	13.69%	13.69%
4	5.687575817	5.706655936	0.34%	0.34%
5	5.60101223	4.891419374	12.67%	-12.67%
6	5.234192371	4.6498678	11.16%	-11.16%
7	4.976121902	4.298282516	13.62%	-13.62%
8	5.168169022	5.495298309	6.33%	6.33%
9	5.353350639	5.548137716	3.64%	3.64%
10	5.232304096	5.468878606	4.52%	4.52%
11	1.407258034	1.849379239	31.42%	31.42%
12	3.024103642	2.275868736	24.74%	-24.74%
13	2.130362511	1.955058052	8.23%	-8.23%
14	2.969447851	2.377773307	19.93%	-19.93%
15	3.517714739	3.575466528	1.64%	1.64%
		Average MAPE	14.02%	**Proportion**
		Average overestimation	9.23%	53.33%
		Average underestimation	-21.36%	46.67%

Test user 4

Time (minutes)	MET SENSEWEAR	MET ALE	MAPE	MPE
1	4.608208179	5.521173919	19.81%	16.54%
2	5.251231194	6.699851497	27.59%	21.62%
3	5.646455288	6.699851497	18.66%	15.72%
4	5.560116291	6.442164901	15.86%	13.69%
5	5.253722668	6.334795486	20.58%	17.07%
6	5.120531082	6.635429848	29.58%	22.83%
7	5.117636681	6.442164901	25.88%	20.56%
8	5.015133381	6.203566201	23.70%	19.16%
9	4.637789249	3.886772823	16.19%	-19.32%
10	1.080942512	1.116641916	3.30%	3.20%
11	1.019237995	1.116641916	9.56%	8.72%
12	0.994113564	1.116641916	12.33%	10.97%
13	1.000736356	1.116641916	11.58%	10.38%
14	1.56575346	2.0678554	32.07%	24.28%
15	3.26573658	5.153731921	57.81%	36.63%
16	4.378562927	5.955423553	36.01%	26.48%
17	5.274942398	6.571008199	24.57%	19.72%
18	5.564236164	6.442164901	15.78%	13.63%
19	5.273506641	4.745728144	10.01%	-11.12%
20	5.157559395	5.411418517	4.92%	4.69%
21	5.000399113	4.960466974	0.80%	-0.81%
22	5.240641594	5.024888623	4.12%	-4.29%
23	5.616485119	5.024888623	10.53%	-11.77%
24	5.537241459	3.800877291	31.36%	-45.68%
		Average MAPE	19.27%	**Proportion**
		Average overestimation	16.99%	75.00%
		Average underestimation	-15.50%	25.00%

Test user 5

Time (minutes)	MET SENSEWEAR	MET ALE	MAPE	MPE
1	4.115631104	4.951971522	20.32%	20.32%
2	4.839316368	6.16424278	27.38%	27.38%
3	5.310350418	6.280549247	18.27%	18.27%
4	5.44306469	6.280549247	15.39%	15.39%
5	5.235547543	5.514172702	5.32%	5.32%
6	5.303361416	6.159769454	16.15%	16.15%
7	5.240242004	5.931629845	13.19%	13.19%
8	5.201133728	5.931629845	14.04%	14.04%
9	4.980983257	5.815323377	16.75%	16.75%
10	1.30994606	2.810739632	114.57%	114.57%
11	1.14974153	1.167538001	1.55%	1.55%
12	1.112897992	1.046758208	5.94%	-5.94%
13	1.081523299	1.046758208	3.21%	-3.21%
14	1.207290173	1.046758208	13.30%	-13.30%
15	2.670848131	2.791355221	4.51%	4.51%
16	3.912192345	4.710411935	20.40%	20.40%
17	4.744679928	5.640863676	18.89%	18.89%
18	5.117501736	6.222396013	21.59%	21.59%
19	4.792237282	5.815323377	21.35%	21.35%
20	4.828104973	4.807333992	0.43%	-0.43%
21	4.709759712	5.439404259	15.49%	15.49%
22	4.537115097	4.63287429	2.11%	2.11%
23	4.682047367	5.059331338	8.06%	8.06%
24	4.497824192	5.233791039	16.36%	16.36%
		Average MAPE	17.27%	**Proportion**
		Average overestimation	19.58%	83.33%
		Average underestimation	-5.72%	16.67%

User test 6

Time (minutes)	MET SENSEWEAR	MET ALE	MAPE	MPE
1	3.434569836	2.193923516	36.12%	-36.12%
2	3.107766867	2.003817731	35.52%	-35.52%
3	4.021060467	4.571673046	13.69%	13.69%
4	4.546656609	4.744842479	4.36%	4.36%
5	4.911518574	4.104115575	16.44%	-16.44%
6	4.657839775	3.809727538	18.21%	-18.21%
7	4.363938332	2.78802788	36.11%	-36.11%
8	4.211478233	4.623623876	9.79%	9.79%
9	4.426214695	4.415820556	0.23%	-0.23%
10	1.394783258	1.135496714	18.59%	-18.59%
11	1.325941205	0.987065771	25.56%	-25.56%
12	1.375855446	1.312757514	4.59%	-4.59%
13	1.286023259	1.194869092	7.09%	-7.09%
14	2.990875483	4.408399009	47.39%	47.39%
15	4.040113449	4.519722216	11.87%	11.87%
16	4.254291534	3.446071728	19.00%	-19.00%
17	4.229239941	4.415820556	4.41%	4.41%
18	4.460990429	4.571673046	2.48%	2.48%
19	4.670148373	5.247033837	12.35%	12.35%
20	4.816396713	5.558738817	15.41%	15.41%
21	4.781062126	5.310116988	11.07%	11.07%
22	4.386910439	4.225334179	3.68%	-3.68%
23	4.301353455	3.238268408	24.72%	-24.72%
24	3.923071384	3.653875048	6.86%	-6.86%
25	4.020167351	4.259968066	5.96%	5.96%
26	4.09445858	4.208017235	2.77%	2.77%
27	4.339653969	4.311918896	0.64%	-0.64%
28	4.492292404	4.571673046	1.77%	1.77%
29	4.498588562	4.308208122	4.23%	-4.23%
30	4.462958813	4.779476366	7.09%	7.09%
31	4.57801199	4.883378026	6.67%	6.67%
32	4.310431004	4.034847802	6.39%	-6.39%
33	3.919087648	4.208017235	7.37%	7.37%
34	3.9187181	4.208017235	7.38%	7.38%
35	3.957479239	3.640268878	8.02%	-8.02%
36	3.909229279	2.751395885	29.62%	-29.62%
37	4.038267136	3.636558105	9.95%	-9.95%

38	4.205454826	4.623623876	9.94%	9.94%
39	4.600668907	5.506787987	19.70%	19.70%
40	4.329372406	4.294601952	0.80%	-0.80%
41	4.42635107	4.675574706	5.63%	5.63%
42	3.838340998	3.601924218	6.16%	-6.16%
		Average MAPE	12.52%	**Proportion**
		Average overestimation	10.36%	47.62%
		Average underestimation	-14.48%	52.38%

User test 7

Time (minutes)	MET SENSEWEAR	MET ALE	MAPE	MPE
1	3.842601061	3.399335172	11.54%	-11.54%
2	4.184263229	5.487169169	31.14%	31.14%
3	5.141989708	5.706655936	10.98%	10.98%
4	5.627134323	5.231101275	7.04%	-7.04%
5	4.779447556	3.698758477	22.61%	-22.61%
6	4.680783749	4.174313139	10.82%	-10.82%
7	5.195515156	5.653816529	8.82%	8.82%
8	4.842628002	3.92772924	18.89%	-18.89%
9	1.363016009	0.951109323	30.22%	-30.22%
10	1.273757339	0.951109323	25.33%	-25.33%
11	1.208735228	0.951109323	21.31%	-21.31%
12	2.849389076	2.888554239	1.37%	1.37%
13	3.820782185	5.706655936	49.36%	49.36%
14	4.471114159	5.600977123	25.27%	25.27%
15	4.527900219	4.491349579	0.81%	-0.81%
16	4.551903248	5.706655936	25.37%	25.37%
17	4.605970383	5.706655936	23.90%	23.90%
18	4.753071785	5.706655936	20.06%	20.06%
19	5.0700984	5.495298309	8.39%	8.39%
20	5.186614513	5.706655936	10.03%	10.03%
21	3.997573853	2.712422883	32.15%	-32.15%
22	4.341001034	4.561802122	5.09%	5.09%
23	4.736684322	5.019743648	5.98%	5.98%
24	5.022885799	5.336780088	6.25%	6.25%
25	5.266694546	5.600977123	6.35%	6.35%
26	5.355753899	5.651784245	5.53%	5.53%

27	5.204791546	5.600977123	7.61%	7.61%
28	5.319404602	5.283940682	0.67%	-0.67%
29	5.294438839	5.706655936	7.79%	7.79%
30	4.847275257	5.107809326	5.37%	5.37%
31	4.26283741	5.389619495	26.43%	26.43%
32	3.97680068	4.861225427	22.24%	22.24%
33	4.258555412	5.389619495	26.56%	26.56%
34	4.481510639	2.624357205	41.44%	-41.44%
35	4.447526455	4.966904241	11.68%	11.68%
36	4.664525986	3.749565599	19.62%	-19.62%
37	4.984632492	5.651784245	13.38%	13.38%
38	4.970356941	5.178261868	4.18%	4.18%
39	4.782707214	5.583363987	16.74%	16.74%
40	4.623749733	5.495298309	18.85%	18.85%
41	3.726695538	2.835714832	23.91%	-23.91%
42	1.404418588	1.461890255	4.09%	4.09%
43	3.081089497	2.730036019	11.39%	-11.39%
		Average MAPE	15.97%	**Proportion**
		Average overestimation	14.60%	65.12%
		Average underestimation	-18.52%	34.88%