A wireless sensor-based system for self-tracking activity levels among manual wheelchair users

Alexandre Grillon[1], Andres Perez-Uribe[1], Hector Satizabal[1], Laurent Gantel[2], David Da Silva Andrade[2], Andres Upegui[2], and Francis Degache[3]

[1] Institute for Information and Communication Technologies IICT, HEIG-VD
[2] Institute on Information Technologies InIT, Hepia Geneva,
[3] University of Health Sciences, HESAV, Lausanne,
University of Applied Sciences and Arts of Western Switzerland (HES-SO)

Abstract. ActiDote —activity as an antidote— is a system that uses wireless sensors to recognize activities of various intensity levels in order to allow self-tracking while providing motivation for manual wheelchair users. In this paper, we describe both the hardware setup and the software pipeline that enable our system to operate. Laboratory tests using multi-modal fusion and machine learning reveal promising results attaining a F1-score classification performance of 0.97 on five different wheelchair-based activities belonging to four intensity levels. Finally, we show that such a low cost system can be used for an easy self-monitoring of physical activity levels among manual wheelchair users.

Key words: self-tracking, wheelchair, handicap, wireless sensors, wearables, machine learning

1 Introduction and Motivations

Physical inactivity has been identified as a major contributor to the exacerbation of physical illnesses. The WHO identified it as the fourth leading risk factor of global mortality after high blood pressure, tobacco use and high blood glucose. Therefore, in recent years, many actions against inactivity have come to the fore. For instance, diverse pedometer devices have been developed to help people reach various physical activity goals, like walking 30 minutes per day or completing 10'000 steps per day. Moreover, many smartphone applications attempt to help people self-track their physical activity and motivate them to continuously exercise. Unfortunately, an equivalent recommendation for people using wheelchairs is missing and there is a clear absence of motivational devices that can support the self-tracking of physical activity among people using manual wheelchairs. The few studies that have dealt with this issue concluded that current commercial physical activity measurement devices are not appropriate for wheelchair users [1], and to make things worse, those users very often adopt sedentary habits as a consequence of their disability. The result is that, according to the Centers for Disease Control and Prevention (cdc.gov), obesity rates for adults with disabilities are 58% higher than for adults without disabilities.

In this paper we present ActiDote —activity as an antidote— a system based on wearable and wheelchair-attached sensors that wirelessly communicate with a smartphone to allow the tracking of the physical activity of people with motor disabilities (e.g., using wheelchairs). Our system integrates machine learning algorithms that analyse the sensor data to identify the intensity of the physical activity being realized, and provides a daily (and weekly) summary of activities as feedback to the user.

This paper is organized as follows: in Section 2, we present the state-of-the-art in the domains of physical tracking and activity recognition among disabled people. In Section 3, we describe the Actidote system, namely the hardware setup, the activity recognition approach and the feedback visualization. Section 4 presents some experimental tests and results, and finally Section 5 presents our conclusions and future work.

2 Related Work

The increasing availability of wearable sensors embedded in smartphones, watches and physical activity trackers has opened the door to a wide number of applications, mainly in health and wellness improvement. Many devices and services help with tracking physical activity, caloric intake, sleep quality, posture, and other factors involved in personal well-being (e.g., the so-called Quantified-Self movement[1]). One typically collects data by means of sensors like GPS, accelerometers, gyroscopes, barometers, heart rate meters, thermometers, microphones, etc. As far as the physical activity monitoring is concerned, recent research and development has allowed to leverage the power of accelerometers for building systems capable of estimating energy expenditure and to achieve mass market penetration (e.g., Jawbone, Fitbit, Nike+, Polar Loop, Garmin vivofit, etc). Unfortunately, it has been found that these general public devices do not provide accurate estimates of the energy expenditure of people using wheelchairs, in particular, during wheelchair propulsion [1, 2]. Even, the SenseWear Armband (SWA), which has been validated as a means to estimate energy expenditure in overweight children, in patients with cancer, and healthy children, it has provided inaccurate measures among the disabled population [3].

Researchers have thus attempted to estimate energy expenditure of manual wheelchair users using activity-dependent models [5], increasing the sampling frequency and computing features of the raw data [4], and by fusing multiple modalities (e.g., accelerometry and heart rate monitoring) [3]. In [4], Garcia-Masso et al., obtained accurate estimations of energy expenditure on paraplegic persons using four wearable sensors (Actigraph GT3X accelerometers), one on each wrist, one on the waist and one in the chest. Finally, there has been some work attempting to better estimate energy expenditure, but without involving disabled people, is the use of Machine Learning techniques to derive data-driven models that exploit data from accelerometers or multiple sensors [6, 7, 8, 9].

[1] http://quantifiedself.com

3 A data-driven approach to physical tracking

The amount of energy spent by a person during a given activity can be easily assessed by using a mechanistic approach if the right physiological variables are measured. However, this measurements are not easy to perform due to different practical constraints (e.g., sensors are expensive and not portable). Data-driven approaches offer an alternative to the mechanistic analyses which have been traditionally used for modelling complex metabolic phenomena. Data-driven models try to discover the relationships between the variables involved in the analysis from the data, while in mechanistic approaches these relationships are based on prior theoretical knowledge about the phenomenon. Indeed, in a data-driven approach the variables used do not necessarily have a physiological meaning, and in most of the cases they are mere proxies to a quantity that is difficult to measure.

One of the main strengths of a data-driven approach is its robustness. The fact of using the acquired data to build the model provides a better tolerance to sensor imprecision given its non-dependence of sensor calibration or precise sensor positioning and orientation. While mechanistic analysis relies on the understanding of a physical phenomena that requires well defined inputs for obtaining an accurate solution, data-driven approaches accept not to understand the phenomena, making it less dependant to the accuracy of the inputs. Our approach is thus to acquire as much pertinent sensor data as possible for further identification of the required inputs to build the model.

3.1 Hardware Setup

In order to acquire the pertinent data for building our data-driven model, we have designed a set of wireless sensors to be embedded on an ordinary wheelchair. Our main goal is to estimate in the most accurate manner the energy expenditure of the user, specifically focusing on the expenditure due to physical activity. The type of physical activity can be estimated through different motion sensors placed on the wheelchair and on the user body (e.g, chest, wrist). However, activity intensity is impossible to infer from motion sensors only. For instance, displacing a wheelchair on a regular surface will require much less energy than on a sandy surface, even if motion sensors read similar data. Static effort is another example, it can be more physically intensive to stay static on an uphill than a displacing on a downhill. Force sensors on the wheels provide a more accurate measure in order to estimate such physical intensity.

We have thus equipped an ordinary wheelchair with a set of several Bluetooth Low Energy (BLE) sensors in order to build a complete physical activity monitoring system (PAMS). The system is composed of five sensors allowing to collect data regarding the movements and the effort done by the wheelchair user.

We have designed a wireless sensing board to be fixed on the wheel. It contains a gyrometer and accelerometer device used to determine the wheel speed and an amplifying circuitry for connecting the strain gauges of 3 load cells (Figure 1). The wheel has been modified in order to assess the forces applied to it through

measuring with load cells the mechanical deformation between the hand rim and the wheel. We have thus replaced the hand rim separators by three load cells in order to measure the tangential force applied to the wheel. Figure 2 depicts how the load cell has been coupled to the wheel. The load cells are connected to form a Wheatstone bridge topology. The two output signals from the bridge are then amplified, compared and filtered. Finally, the result is read by an embedded microcontroller through an analog-to-digital converter to get the strength value. Figure 3 shows the complete hardware setup.

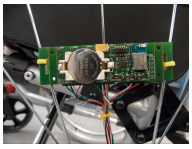

Fig. 1. Wireless sensor board

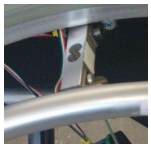

Fig. 2. Load cell coupled to the wheel

Fig. 3. Modified wheel

A second wireless sensing board is directly fixed to the chair. This one is also equipped with accelerometer and gyrometer but without strain gauge interfaces. Its goal is to measure wheelchair inclination for discriminating between upslope, downslope and flat surfaces as well as rotations of the wheelchair.

Both sensing boards are equipped with a Bluetooth Low Energy (BLE) module. The module is the BLE113 developed by Bluegiga. It integrates a calibrated antenna at 2.4 GHz and a low consumption microcontroller. The main interest in this all-in-one module is the reduction of the development time on the hardware design and the full implementation of the Bluetooth stack usable with a simple script language named BGScript.

Finally, a commercial smartwatch (i.e., the Moto 360) is fixed on the user's wrist. It is also equipped with motion sensors and is used to detect arm motions in order to improve the model with the detection of other gestures, like auto-propulsing his chair, playing ping-pong or lifting weights. Moreover, this smartwatch is also equipped with a photoplethysmogram sensor used to monitor the users heart rate (HR). HR measurements will permit to enhance the model by directly using physiological data for estimating specific users caloric expenditures needs and capacity.

All the above described collected data from custom sensors and commercial smartwatch is transferred through BLE to an Android handheld device carried by the wheelchair user in order to further perform data analysis.

3.2 Activity recognition

In our project, activity recognition was first envisioned as an intermediate objective, giving that we aim at estimating energy expenditure in the end. However, in this paper we present a system that allows for self-traking of physical activity levels taking already advantage of the activitied being recognized, as in [10]. Indeed, as we will show in the next section, a graphical summary of the amount of time spent on activities of different levels of intensity can be exploited in a straighforward way to motivate more regular physical exercise. The processing chain of the sensor data [11] starts with the sensor-data acquisition: a stream of sensor samples is obtained and the sensor data stream is preprocessed. Typical transformations are calibration, denoising, or sensor level data fusion. Then, the data stream is segmented into sections. A common type of segmentation technique is the sliding window. To characterise these raw data and reduce their dimensionality, features are computed on the identified segments. A classifier, trained at design-time, maps the feature vector into a predefined set of output classes (e.g., activities). A detailed explanation of this activity recognition chain is described in Section 4.

3.3 Feedback visualization

We found important for the end user to have some kind of feedback available. We developed a front-end web interface that shows both the current day summary and a history of the previous week. The interface in its current state is depicted in Figure 4. All statistics are related to the following intensity levels: Sleeping (or None), Light, Moderate, and Vigorous. For now, this interface is in a beta test state. In a future version of this project, we hope that the users will be able to log on to this interface and see these statistics in near real time.

4 Experiments and Results

4.1 Data Collection

In order to recognize different activities, we started by determining the activities that should be detected and learned. This list is formed of activities such as: resting, moving (e.g., at different speeds, on different ground types, given different slopes, self-pushed, pushed by a caretaker), desk-work-like (e.g., desk work, browsing on computer, eating, being on the phone) or even replacements (e.g., in the chair, from the wheelchair to the toilet). Moreover, these activities are grouped in several classes of intensity ranging from sleeping (or None) to

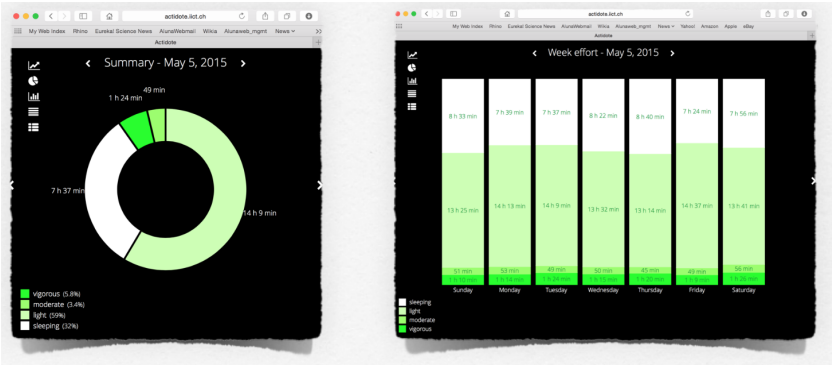

Fig. 4. The web interface for the end user feedback

vigorous, as already shown in Section 3.3. This is done to pave the way for the energy expenditure predicting model that we will develop in a future phase of this project. For the machine learning to take place, the first step is to collect data corresponding to the goal that needs to be achieved. In this case, it means to use the sensors we described to collect the data of the activities we mentioned above. This operation is performed by an Android handheld device. An application allows to configure the different devices regarding the sensors period and data resolution, and to gather the collected samples in several log files to be analyzed by the activity recognition processing chain. The data captures followed a protocol that encompassed five activities: resting, desk work, self-pushing at medium speed on a standard flat indoor floor, self-pushing on an ascending slope, and external pushing by a caretaker. Therefore, each activity defined in the list had an intensity level: none, light, moderate, and vigorous, respectively. The protocol requires no predefined order for doing the activities, but defines that each activity has to be performed for five minutes straight. This is a precaution to make sure that the model has enough examples for each activity and can therefore learn all of them correctly. This protocol is obviously only designed for laboratory purposes and is used to train a model to recognize such activities. The goal is eventually to be able to classify captures free from any protocol, such as captures that any person with Spinal Cord Injury would do in a normal day. To ensure that our model performs well, we chose to train it on a specific data capture, and then to test it on another data capture and the other way around. This way, we can be sure that our predictive model generalizes well on the activities themselves and not on a spurious setup depending on some day-dependent variables. Of course, the more data we have the better the model becomes, but two different captures is a good first step. Furthermore, we grouped the obtained data following an early fusion pattern. We downsampled the Moto 360 accelerometer data by decimation from 50 Hz to 15.63 Hz and upsampled the Moto 360 heart sensor from 1 Hz to 15.63 Hz by repetition. Therefore, at the end of the data collection step, we had two fully labelled datasets of roughly 25

minutes with 5 minutes by activity and a 15.63 Hz sampling rate (therefore with a sample for each sensor every 64 ms).

4.2 Feature extraction

To perform the feature extraction, we chose to roll two windows of respectively 5000 and 1000 ms over the data, in order to have two order of magnitudes of what we can consider a context for every sample. We computed the median, standard deviation and energy of the signal of the windows leading to each sample (included). We extracted these values for the three axes-accelerometer on the wrist, the three axes-accelerometer on the chair, the three axes-gyroscope on the wheel and the three load cells on the pushing ring. Unfortunately, the heart rate sensor from the Moto 360 on the wrist proved to be too noisy and we had to drop it from our analysis. We were therefore left with two captures of respectively 22471 samples (24 minutes) and 23439 samples (25 minutes), with the 72 same features. As a lot of these features are probably redundant and correlated, the best choice for an off-the-shelf model is random forest. This is a model introduced by Breiman in 2001 [12] that bags (short for bootstrap aggregates) classification and regression trees (CART) and randomly samples a subset of the features at each split. In doing so, the model averages many noisy but approximately unbiased models, and hence reduce the variance. For classification, a committee of trees each cast a vote for the predicted class. As trees are invariant under linear modification of input, this also speeds up the activity recognition chain since no further preprocessing is required.

4.3 Analysis and Results

As stated before, we wanted to make sure that our model performed well by training it on a specific data capture and testing it on the other one. We also had to do it the other way around to make sure that the model performs equivalently, otherwise it would indicate an inherent problem with the training step. On another topic, we decided that in order to assess the degree of activity, it made no sense to predict very short activities. Indeed, the end user will have no interest in knowing every activity he did at a very high resolution (64 ms); he will be more interested in a more global feedback about his day. This is why we decided that a good refinement for the predictions was to roll a modal filter of 1 minute length over them. In doing so, we make sure that a very short change of activity will be erased. This can both improve or worsen the predictions depending on the conjuncture, but it will always enhance the feedback we can give to the end user.

Experiment 1: All features The first experiment we tried was to feed into our model all of the features we extracted. For both ways of training and predicting (train on first capture and predict on second capture, and the other way around), we repeated the training ten times and we computed both the normalized confusion matrices and the weighted F1-scores of the raw resulting classification and

of the refined predictions as described in Section 4.3 with the modal filter. In doing so, we expect the results to be consistent independently from the order in which we train or predict the data captures. We used the ensemble package of scikit-learn [13] for building our random forest classifier:

```
RandomForestClassifier(n_estimators=1000, criterion='entropy',
                       oob_score=True, max_features='sqrt')
```

The results of this first experiment, are described in Table 1. Each row in the table represents a different order for training and predicting, each column contains results on the training set (by out-of-bag score), on the testing set with raw predictions and on the testing set with refined predictions.

Training Set	Prediction Set	Train	Test (raw)	Test (refined)
Capture 1	Capture 2	0.9998 ± 0.00	0.7754 ± 0.01	0.7678 ± 0.03
Capture 2	Capture 1	0.9999 ± 0.00	0.9314 ± 0.00	0.9874 ± 0.00

Table 1. F1-scores on training, raw predictions and refined predictions for both training orders

As we see in the results on Table 1, both ways of training and predicting are not equivalent. Indeed, the score on the test set when trained of the first capture drops dramatically compared to the same score when trained on the second one. This indicates a disparity between the two captures. This is a recurring problem in machine learning: when not enough data is available, the noise can sometimes be mistaken as signal. The more data we have, the more the noise gets averaged among captures and the easier the signal can be recognized. In our case, we analyzed this problem as being caused by proxy features. Proxy features are features that seem to be discriminant in a given dataset but are not in reality. We included an example for illustration: In Figure 5, we see that if we look at the first capture, this variable —the energy of the chair accelerometer signal in the x-axis over the last 5000 ms— seems excellent to separate activity classes *Still* and *Work*. On the other hand, we see in the second capture that this variable has actually no importance and does not separate these two activities at all. In this case, we can deduce that the proxy variables used did not accurately represent both activities. A classifier, even a human one, trained on the first capture will obviously make mistakes when predicting the second capture. Conversely, a classifier trained on the second capture will select another variable to separate these activities and will probably separate correctly these activities. This example explains the dispaired results shown in Table 1.

This problem is caused by lack of data and should disappear as we collect more data, but in the meantime, a good way of overcoming that issue is to manually select features that we know from expert knowledge are relevant for characterizing the activities we defined.

Experiment 2: Manually selected and engineered features In this experiment, we crafted and hand-picked features that we know are important for

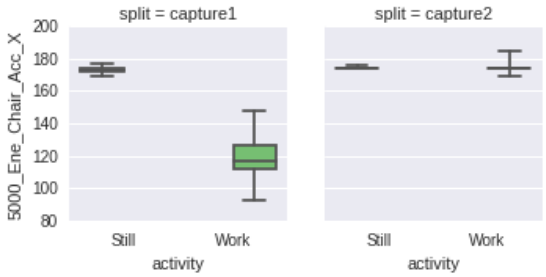

Fig. 5. A "proxy" feature in first (left) and second (right) capture, in activities Still and Work

the resolution of this problem. We therefore ensured that our model only get features that are relevant in every case. Indeed, every feature is certified by expert knowledge to be of influence in the activity we hope to recognize. These features were namely: the sum of the standard deviation of the three load cells over the last 5000 ms, the sum of the standard deviation three axess of the wrist over the last 5000 ms, the median orientation of the chair in the longitudinal plane over the last 5000 ms, the standard deviation of the wheel angular speed in the z-axis (longitudinal ground speed) over the last 5000 ms, the median wrist acceleration in the x-axis (longitudinal acceleration) over the last 5000 ms, and the median wrist acceleration magnitude (norm of the three axis vector) over the last 1000 and 5000 ms. We build models and estimate their performance as it was explained in Section 4.3. These results are presented in Table 2. The model is now working as intended, both training orders are equivalent in terms of results which is the sign of a correct training step. Some more visual results have been computed, they are available in Figures 6, 7, 8 and 9. Figure 6 depicts the confusion matrices of ground truth against raw (left) and refined (right) predictions with the first capture used as training and second capture used as testing, while Figure 7 shows the same matrices but with the reverse training order.

Training Set	Prediction Set	Train	Test (raw)	Test (refined)
Capture 1	Capture 2	0.9995 ± 0.00	0.8923 ± 0.00	0.9550 ± 0.00
Capture 2	Capture 1	0.9985 ± 0.00	0.8844 ± 0.00	0.9933 ± 0.00

Table 2. F1-scores on training, raw predictions and refined predictions for both training orders with manually selected features

We see that the results are consistent across both orders of training and prediction, which is the sign that the model learned correct representations of the defined activities. We also highlighted the effect, positive in this case, of the modal filter in Figures 8 and 9. These figures present the raw (left) and refined (right) predictions on the second capture (when the model was trained on the

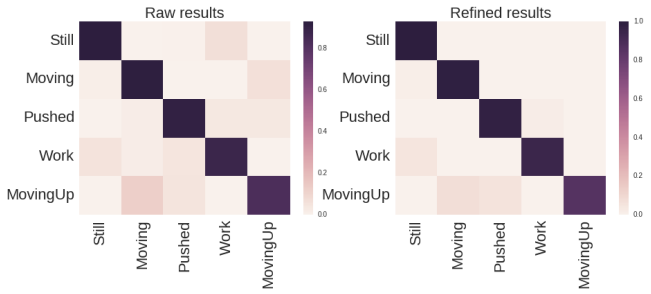

Fig. 6. Confusion matrices of raw (left) and refined (right) predictions using first capture as train test and second capture as test set

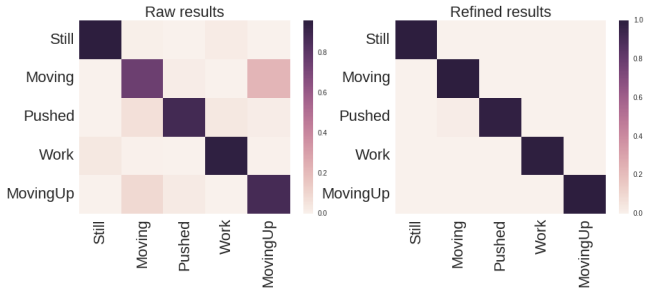

Fig. 7. Confusion matrices of raw (left) and refined (right) predictions using second capture as train test and first capture as test set

first one) in Fig. 8 and on the first capture (when the model was trained on the second one) in Fig. 9. We also see that this filter has a lag effect causing the detection of a start of a new activity to be slightly delayed comparing to its real start.

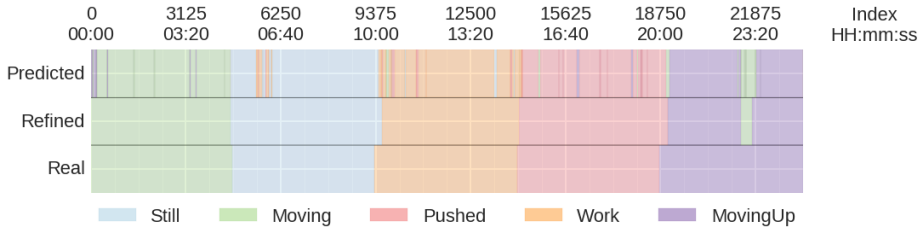

Fig. 8. Timeline of ground truth (bottom) against raw (top) and refined (middle) predictions using first capture as train test and second capture as test set

We can therefore conclude that the system in its basic version can recognize the defined activities with a F1-score of roughly 0.9, which is already an excellent score for an activity recognition task. In its refined version, with a 1 minute

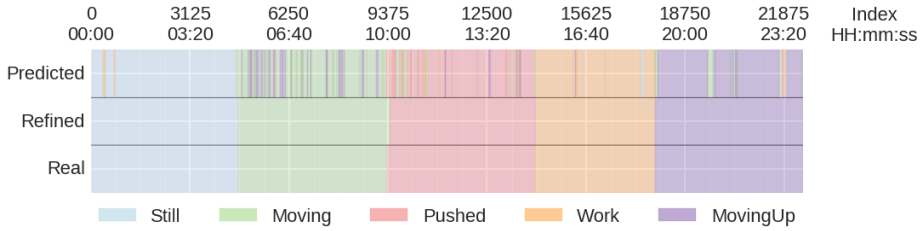

Fig. 9. Timeline of ground truth (bottom) against raw (top) and refined (middle) predictions using second capture as train test and first capture as test set

modal filter applied to it, the results seem to climb around 0.97. Of course, this is an even better score, but its contribution does not only encompasses the score enhancement, but also -and more importantly- gives a much clearer feedback to the user. Knowing the system is able to correctly classify such activities, it is therefore also able to separate activities with different intensities, since the activities are grouped into intensity levels.

On a broader subject, selecting the features to feed into our model should not be a mandatory step given a sufficient volume of labelled data. Indeed, the noise contained in each individual capture should theoretically be averaged out.

5 Conclusions and Future Work

In this paper, we proved that our ActiDote —activity as an antidote— system in its current state can accurately detect and recognize activities of different intensity levels. Indeed, a weighted F1-score of 0.97 has been attained on two different data captures. This means that without seeing any examples of one data capture, the model was able to accurately predict its content, based only on the other data capture examples. In order to have a robust model that can adapt to different wheelchairs and different users, it is mandatory to train it on a lot more data. This is why one of the next step is to capture labelled data for a lot of people with the current setup.

The system as described is available at lower costs compared to similar setups, and gives the end user a very accurate feedback. Therefore, not only does it provide a means for self-tracking physical activity and self-motivation, but it can also be used for potential continuous monitoring of patients during rehabilitation. Last but not least, the development of self-tracking devices and continuous monitoring should indirectly contribute to reduce health costs.

We are currently working on improving the system accuracy by developing new hardware and software. From the hardware side, we are currently developing our own strength sensors based on strain gauges. The main improvement will consist on measuring the force applied on a radial axis in addition to the tangential axis already measured. Other sensing elements are also being evaluated like pressure sensors on the chair surface in order to track posture and activities like

switching chairs (transfer on public bathrooms, etc.) or adjusting seat position (pressure relief, etc.).

As to the software side, we do not only aim at making this system near real-time, but also capable of recognizing more activities. The next steps encompass extrapolating an energy expenditure measurement, in order for the user (or caregivers) to have a quantitative estimation as feedback.

Acknowledgements

This project has been supported by the HES-SO grant IA-INTERDISC13-04. We would like to thank all the members of the ACTIDOTE consortium for fruitful discussions and collaboration.

References

1. Hiremath, S.V. and Ding, D., Evaluation of activity monitors in manual wheelchair users with paraplegia, J. Spinal Cord Med. 2011, 34(1), 110-117
2. Conger, S.A. , Physical Activity Assessment in Wheelchair Users, PhD diss., University of Tennessee, Knoxville, 2011
3. Tanhoffer, R.A. et al., Comparison of methods to assess energy expenditure and physical activity in people with spinal cord injury, The journal of Spinal Cord Medicine, 2012, 35(1), 35-45
4. Garcia-Masso, X., et al., "Validation of the use of Actigraph GT3X accelerometers to estimate energy expenditure in full time manual wheelchair users with spinal cord injury", Spinal Cord (2013) 51, 898903
5. Gendle, S.C. et al., Wheelchair-mounted accelerometers for measurement of physical activity, Disability and Rehabilitation: Assistive technology, 2012, 7(2), 139-148
6. Vyas, Nisarg, et al. "Machine learning and sensor fusion for estimating continuous energy expenditure." AI Magazine 33.2 (2012): 55.
7. Staudenmayer, John, et al. "An artificial neural network to estimate physical activity energy expenditure and identify physical activity type from an accelerometer." Journal of Applied Physiology 107.4 (2009): 1300-1307.
8. Ruch, Nicole, et al. "Neural network versus activity-specific prediction equations for energy expenditure estimation in children." Journal of Applied Physiology 115.9 (2013): 1229-1236.
9. Pande, A. et al., Accurate energy expenditure estimation using smartphone sensors., 8th Intl. Conference on Body Area Networks, BodyNets13, ICST, Brussels, pp. 8-14
10. Ding, Dan, et al. "Detection of wheelchair user activities using wearable sensors." Universal Access in Human-Computer Interaction. Context Diversity. Springer Berlin Heidelberg, 2011. 145-152.
11. A. Bulling et al., A Tutorial on Human Activity Recognition Using Body-Worn Inertial Sensors, ACM Computing Surveys, Vol. 46, No. 3, Article 33, January 2014
12. Random forests L Breiman Machine learning, 2001
13. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.