LifeStream: Design and prototypical implementation of a monitoring system for dispatch life support

Florian Grassinger, Jakob Doppler, Markus Wagner and Wolfgang Aigner
Institute of Creative\Media/Technologies, St. Pölten University of Applied Sciences, Austria
first.lastname@fhstp.ac.at

***Abstract*—Most laypersons who reanimate for the first time do it inappropriately. Until now the only way to review the ongoing reanimation was verbal feedback by the dispatcher on the phone, who has only limited resources in order to review the reanimation process. To overcome this issue, we designed and implmemented LifeStream, a system using current smartphone technologies in order to measure reanimation parameters: chest compression rate (CCR) and chest compression depth (CCD). The system is based on a server, web client and mobile application, which gathers, processes and transfers the data. The development of algorithms for CCR and CCD detection as well as the evaluation of the system functionality is part of this paper. We conducted a 2-day user test, where we compared the guided standard reanimation process to the application supported process. The results of the tests showed that it is possible to develop an application, which runs for at least ten minutes (crucial time till ambulance arrives) and enhances the whole reanimation cycle for laypersons and dispatchers [1].**

I. Introduction

Our fast aging population results in an increase of out-of-hospital cardiac arrest situations. Often dispatch life support and Cardiopulmonary Resuscitation (CPR) interventions are performed by untrained laypersons and bystanders rather than medical professionals [2]. The fear of making bad decisions often restrains people from helping and saving life's or bridge the critical minutes until the ambulance arrives [3]. Time critical medical emergency situations are situations where a proper execution of all steps in the chain of survival is crucial and therefore every second counts [4]. A CPR often requires immediate reaction and even if the chest compressions are not totally appropriate, the attempt is crucial to save a person's life. Over the past years, cardiopulmonary resuscitation has continuously improved and was further investigated by Roessler et al. [5].

Today's smartphones are equipped with multimodal sensors to measure important context and even vital parameters that can be used to assess the situation during a reanimation. For the development of a functional prototype, which assists laypersons or unexperienced people by performing chest compressions, we used the accelerometer sensor of the smartphone. Additionally, we utilized the network connectivity, as well as maintain an ongoing phone call and perform background tasks such as transmitting real-time data to develop an effective algorithm for chest compression rate and depth detection.

The main goal of this research is to present a prototypical implementation of a system which uses algorithms for chest compression rate (CCR) and chest compression depth (CCD) detection and compares them to existing standards and therefore enhance the overall reanimation process for the dispatcher and the layperson.

The major contribution of this paper is a functional prototype that was tested during user tests and evaluated as well as a straightforward experimental implementation.

II. Related work

There are a number of tools and research work, that deal with the quality of CPR and its enhancement. However, none of them transmits the data in real time to an emergency medical dispatcher (EMD). *PocketCPR* [1] is a mechanical device that enhances the quality of CPR by simple audiovisual feedback in real time, which was already evaluated [6]. The mobile version is called *ZOLL PocketCPR*, [2] which gives real time feedback of an ongoing CPR through the smartphone. It uses smartphone sensors to give the user audio-visual feedback and introduces the user to the whole process of CPR. *CPREzy* [3], is designed for CPR assistance and offers a simple interaction. It has an audible chirp and visual light pacing system with a metronome to guide the CPR. In a study the device was compared with a normal reanimation and the results have shown, that there was no significant difference in compression rate or duty cycles between the techniques [7]. Song et al. [8] describe the usage of an inbuilt accelerometer sensor in smartphones to enhance the quality of a CPR by directly measuring the CCR and CCD. The main difference is that the feedback is restricted to the user and not an EMD. Up to now, like the ones mentioned, have begun to examine how to enhance the quality of CPR. But none of these studies concentrate on direct user feedback and on the dispatch of crucial CPR parameters to the EMD.

III. Background

Using an accelerometer the following physical and technical backgrounds should be considered:

A. Physical considerations of spatio-temporal parameters

The algorithms for CCR and CCD detection are based around the physical concept of acceleration and its first and second order integral velocity and distance. Any change in the velocity of an object results in acceleration. So acceleration

[1] http://www.zoll.com/de/produkte/pocketcpr/, Accessed: April 11, 2016
[2] https://goo.gl/B1pdMR, Accessed: April 11, 2016
[3] http://www.heartworkscpr.com/cprezy-facts.html, Accessed: April 16, 2016

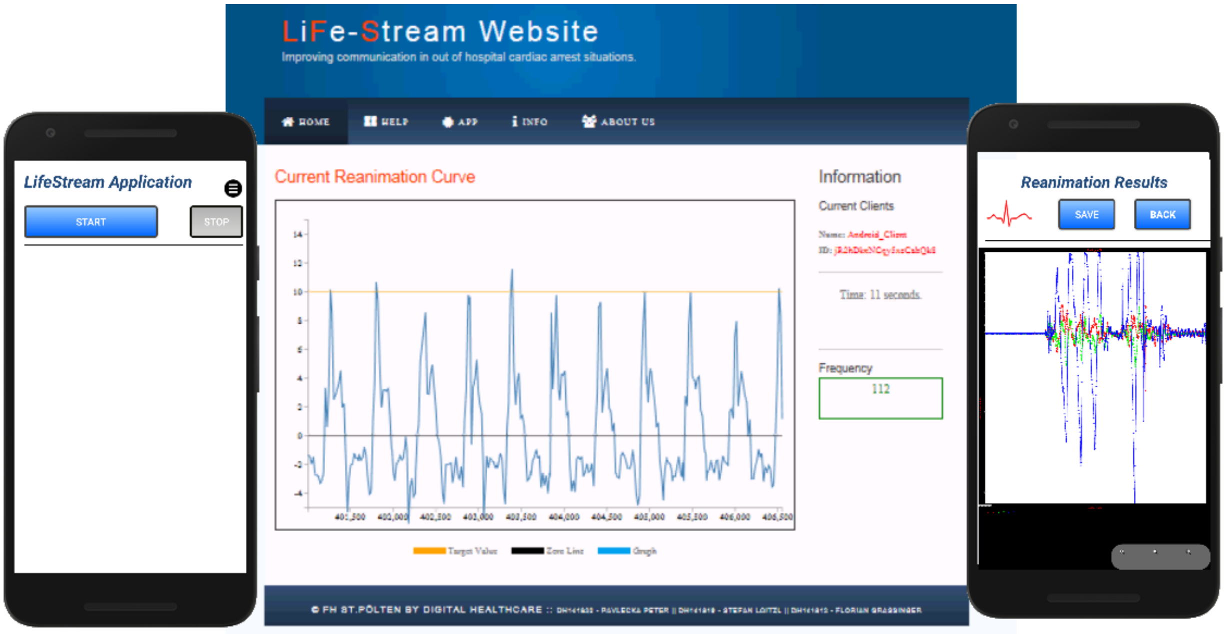

Fig. 1. The website client and the application. On the left side is the starting point of the application and on the right side the results view after the application was finished (mainly for testing purpose and recording).

is related to velocity, or depends on the change of it [9]. The relation of acceleration, displacement and velocity is important, as all of these three quantities are vector quantities (give information about direction) [10]. Using the example of displacement (the directed distance between two points *A* and *B*), it is theoretically possible to determine the final position of the mobile phone, if it is used for CCD detection. The distance would be wrong, as it's only a scalar and counts up the traveled way and not the direct way between two points. Frequency detection is also possible, because after a certain push threshold is exceeded, the push is correct and this counts to the total frequency.

B. Technical considerations

The accelerometer is a powerful mechanical low cost sensor, which is implemented into nearly every smartphone. It offers the possibility to measure the acceleration in a specified direction. The values measured by the smartphone are in m/s^2 and always include the acceleration and deacceleration [11]. In essence the accelerometer measures force that is applied not acceleration. Acceleration just causes an inertial force that is captured by the force detection mechanism of the accelerometer or acceleration is the amount of force needed to move each unit of mass. All calculations take place directly on the smartphone and are processed further to the server, redirecting them to the lifestream website (see 1 for visualization. For the prototype the visualization is restricted to one mobile client. Later, each EMD has his own implementation in the already established call taking system where it shows the visualization of an ongoing CPR.

Fig. 2. Reanimation with phone and LifeStream-App.

IV. Design & Implementation of LifeStream

Based on the input of project members and partners the requirements for the main prototype were formulated: A mobile client with a medical dispatch visualization server to handle clients and a visualization website for visualizing data.

A. Usage scenario

When receiving an emergency call, the EMD advises the caller to open the application (if not open already). Then the application registers at the server endpoint of the medical dispatch center and starts the streaming session. The EMD then instructs over the phone and guides the reanimating layperson through the process. The phone has to be placed between the hands and the victim. Although, many people hesitate (results of user studies) to push directly on the phone, it won't crack in most cases as the hands are laying flat on the phone. Figure 2 shows the placement of the hands.

During the usage scenario definition and the continuous collaboration with the project partners and members, the following four main design considerations were defined:

1) **Simple usability**: The application must be simple and has to gather data, perform calculations, transmit it and stop the whole data acquisition and transmission process.
2) **Restricted functionality**: Frantic laypersons require an application that is protected against unwanted termination. This means the buttons, which are normally used to terminate the application or go back, were disabled. The application is also running in full screen mode and it stays in wakeup state the whole time (10 minutes minimum till ambulance arrives [1]).
3) **Easy configuration**: A simple and non-intrusive menu is used, which allows the change of parameters for the calculation and termination of streaming.
4) **Fast transmission**: As a stable network connection cannot be granted the transmission has to be optimized. Therefore a small and simple data format (JSON [12]) is used, which already contains calculations.

B. Server, Website & Mobile Client

Server: The server is a basic NodeJS server which serves the website and redirects the mobile clients. It allows bidirectional communication, so real-time communication is possible. The server distinguishes between normal clients (web) and mobile clients (Android), who transmit data to it.

Website: The website combines various web technologies. D3.js [4] is used for visualizing the data in a running line graph in real time. The website can be reached over the domain *lifestream.fhstp.ac.at*. At the current prototype state every web client receives the website and while an Android client is connected and streaming data, he can view the reanimation data (seen in Figure 1). On the website the visualization is separated into two major parts. First, the dynamically updating line chart that constantly plots the acquired reanimation data from the smartphone. The color codes used for the frequency, or pushes per minute, are abstractions based on the frequency range:

- Red indicates a very bad frequency (all under 90 or above 130).
- Yellow indicates an average frequency (from 90 to 100 & 120 to 130 pushes).
- Green indicates an optimal frequency (from 100 to 120 pushes).

Second, the information section includes information about the registered clients and the reanimation parameters, e.g., the *frequency*.

Mobile Client: The mobile client is available for Android devices and opens a stream to the server and transmits data of an ongoing reanimation. The data acquisition, calculation and transmission can be started with a simple button press, while the stop functionality is hidden in a small menu above along other configuration options.

[4]https://d3js.org/, Accessed: April 28, 2016

C. Calculation restriction:

Physically and theoretically it should be possible to calculate the traveled distance of the phone by using the accelerometer. If the acceleration is integrated once, the result is the velocity of the object (in this case the smartphone). After a second integration the result is the traveled distance [9], [13]. Despite these equations seem fairly straightforward to implement, they are practically not possible. The natural spread error propagates problematically after each integration as well as the included gravitational force that applies to the phone. A solution to this problem is the usage of a *linear accelerometer*, a sensor fusion of various other sensors that factors out the gravitational force. The main issue with using the above mentioned method is that accelerometers are bad at dead-reckoning (continuous position determination). Accelerometers have some noise which varies from smartphone to smartphone as each has its own manufacturer and device type. The noise can be filtered using various filter types, but normal accelerometers produce raw data, which is not filtered or smoothed. This noise will usually result in a non-zero mean, that is continuously added and accumulates in the resulting velocity signal and later of course in the distance integration. This behavior is called sensor drift, as the integration starts fairly well, but quickly accumulates the errors and the resulting values drift away.

Using the linear accelerometer of the Android system leads to better results, as the gravity is already removed and the resulting values are much smoother. After the gravity is removed and the values are read and filtered with a respective filter, it is advised to calculate the magnitude of the acceleration values before continuing with further calculations [11].

D. Calculation Solution:

By taking all the previous problems and considerations into account, a final functional prototype was developed. The algorithm is a very basic but powerful *peak detection* and *frequency* estimator. After 15 seconds (an adequate update time, based on expert feedback) the frequency on the website is updated based on the average reanimation frequency during this time. The frequency is calculated for these 15 seconds or any other interval, approximated to one minute and then transmitted to the server along with other values (e.g. approximate pressure depth).

The optimal frequency of 100 pushes per minute should theoretically be achieved by pushing always at least five centimeters into the chest of the victim [2]. As CCD detection with the given sensor and technology is not really possible, the approach with frequency seemed more promising as well as an approximation of the distance based on the z-axis acceleration. As the performed reanimation of the user normally changes over time, especially when the power ceases, the frequency detection is very difficult. The requirement to the algorithm must be to detect hard pushes as well as faint pushes. Therefore, peak detection is implemented. According to previous studies and extensive acceleration data logging and plotting, the following concept was devised. Once the

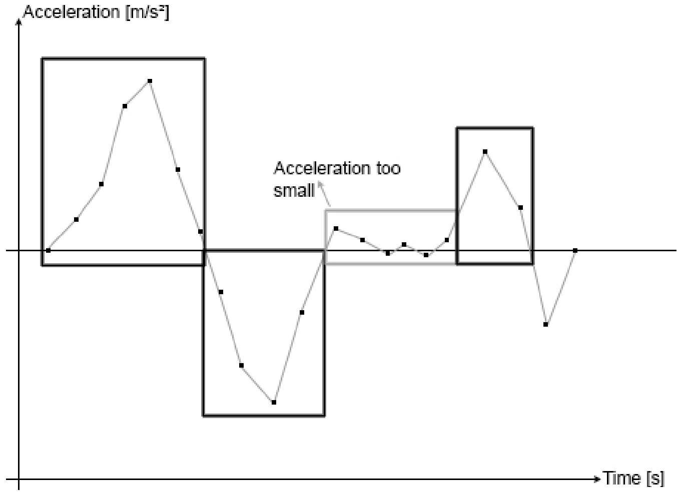

Fig. 3. Peak detection and minimum threshold (along z-axis).

acceleration values or signal traverse the zero line, a change in acceleration happens and a peak can be detected (fig. 3).

Thus, no matter how weak or hard the user pushes, the peak can be detected by its zero-line crossing and change of acceleration with a minimal threshold of applied acceleration. The algorithm is still based on the basic equation and calculation of the magnitude,

V. Evaluation & Validation of prototype

The algorithm was evaluated during a two-day user study, which involved 25 laypersons between 18-50 years (14 lay rescuers and eleven paramedic experienced). They have been selected randomly during an open experiment at St. Pölten University of Applied Sciences, Austria. Each volunteer was instructed before by two professional paramedics. The setup included a reanimation phantom as well as a professional EMD, which was in his actual workplace. The participants were filmed during the whole process and the reanimation phantom also recorded the reanimation process for later comparison to the algorithm. Each participant was not further instructed in the CPR process and they had to reanimate (guided) for full ten minutes. Further randomization happened as some of the laypersons were just reanimating on the phantom without the mobile application. For further insight in the evaluation and test scenario refer to [14] and [15].

A. Results

The outcome of the tests clarified that a guided CPR by using the system is far more efficient for both sides, the EMD and the layperson, rather than a standard phone guided CPR. Some other interesting outcomes as well are:

- Most people hesitate to push on a phone, as it could crack.
- The application detects the frequency very well and is comparable to a professional reanimation phantom that is used for training purposes. However, the accuracy is not as high as a professional sensor, compared to a smartphone accelerometer.
- Displacement can be detected over a short amount of time (movement along the z-axis).
- Displacement detection is not possible with the low cost accelerometer.

VI. Conclusion & Limitations

The performed tests have shown that available smartphone accelerometers along with their embedding systems vary widely and often heavily rely on the hardware and the algorithm used. The accelerometer sensor is often erroneous and creates a non-zero mean that adds up to further calculations. The only solution is filtering and using a linear accelerometer. Often enough the sensor samples slightly slower than the actual sampling frequency as other tasks are more important for the operating system in the background. That means for any calculation it is problematic to rely on fixed time intervals as they are often slightly shorter or longer. The errors are adding up over time and contribute heavily to the whole calculation. A possible solution was to wait an offset time before calculating and estimating depth. At least 100 ms proved to be useful (discarding all before). A remaining problem is that some time stamps are 100 ms or longer and also the fact that numerous important values are lost during the defined pause. During peak detection this can be fatal, as a global maximum could be skipped.

During development it turned out that frequency detection is much easier than continuous position determination, especially in smaller unit ranges (like cm). The theoretic (and physically correct) equations are not feasible for usage in real world applications. Accelerometers measure the acceleration in a body-fixed reference frame, where normally displacement in earth-fixed reference frames is necessary. Therefore, it is not possible to only integrate the accelerometer twice and find the displacement, except it is rotated into the earth fixed frame before the integration takes place. The project showed, that with the given premises of only using the low cost accelerometer sensor in smartphones, it is not possible to make a sturdy point about the displacement. Nevertheless, it is possible to make a point about the current reanimation frequency very well by using the developed peak detection algorithm. Even a position determination could be possible by using the peak detection and the currently viewed values during the peak detection (a so-called window of values) for the integration. As the values are always restricted to a certain amount and interval, a double integration of those values would contain less errors that could add up over time.

Acknowledgments

The authors would like to thank Stefan Loitzl and Peter Pavlecka who have been former project members, contributed during development and lead the testing part. We also would like to give a special thanks to our project partners from Nortuf Niederösterreich especially Heinz Novosad and Raphael Van Tuldar for their support. This work was supported by the Austrian Science Fund (FWF) via the “VisOnFire” project (P27975-NBL).

REFERENCES

[1] M. Ljunggren, M. Castrn, and K. Bohm, “Abstract 231: The Effect of Dispatcher Call Processing Interval and Ambulance Response Interval for Survival From Cardiac Arrest,” *Circulation*, vol. 128, no. Suppl 22, pp. A231–A231, Jan. 2016.

[2] R. O. Cummins, J. P. Ornato, W. H. Thies, and P. E. Pepe, “Improving survival from sudden cardiac arrest: the ”chain of survival” concept. a statement for health professionals from the advanced cardiac life support subcommittee and the emergency cardiac care committee, american heart association.” *Circulation*, vol. 83, no. 5, pp. 1832–1847, 1991.

[3] L. Wik, J. Kramer-Johansen, H. Myklebust, H. Sreb, L. Svensson, B. Fellows, and P. A. Steen, “Quality of cardiopulmonary resuscitation during out-of-hospital cardiac arrest,” *JAMA*, vol. 293, no. 3, pp. 299–304, Jan. 2005.

[4] C. Ro, “The ”chain of survival” concept: how it can save lives.” *Heart disease and stroke : a journal for primary care physicians*, vol. 1, no. 1, pp. 43–45, 1991.

[5] B. Roessler, R. Fleischhackl, H. Losert, C. Wandaller, J. Arrich, M. Mittlboeck, H. Domanovits, and K. Hoerauf, “Cardiopulmonary resuscitation and the 2005 universal algorithm: Has the quality of CPR improved?” *Wiener klinische Wochenschrift*, vol. 121, no. 1-2, pp. 41–46, Jan. 2009.

[6] S. Kirkbright, J. Finn, H. Tohira, A. Bremner, I. Jacobs, and A. Celenza, “Audiovisual feedback device use by health care professionals during CPR: A systematic review and meta-analysis of randomised and non-randomised trials,” *Resuscitation*, vol. 85, no. 4, pp. 460–471, Apr. 2014.

[7] G. D. Perkins, C. Augr, H. Rogers, M. Allan, and D. R. Thickett, “CPREzy: an evaluation during simulated cardiac arrest on a hospital bed,” *Resuscitation*, vol. 64, no. 1, pp. 103–108.

[8] Y. Song, Y. Chee, J. Oh, and S. Lee, “Development of android based chest compression feedback application using the accelerometer in smartphone,” in *Proceedings of the International Conference on Biomedical Engineering and Systems*, pp. 130–1.

[9] M. Mounts, “Physics hypertextbook,” vol. 44, pp. 1946–1947, 01 1970.

[10] J. Rybach, *Physik fr Bachelors: mit 92 durchgerechneten Beispielen, 176 Testfragen mit Antworten sowie 93 bungsaufgaben mit kommentierten Musterlsungen*, 2nd ed. Mnchen: Fachbuchverl. Leipzig im Carl-Hanser-Verl, 2010.

[11] G. Milette and A. Stroud, *Professional Android sensor programming*, ser. Wrox Programmer to programmer. Indianapolis, Ind: Wiley, 2012.

[12] N. Nurseitov, M. Paulson, R. Reynolds, and C. Izurieta, “Comparison of JSON and XML data interchange formats: a case study.” *Caine*, vol. 2009, pp. 157–162, 2009.

[13] P. Becker, “Erfassung und Verarbeitung von Sensordaten,” Hochschule Bonn-Rhein-Sieg, 2013.

[14] P. Pavelcka, “LiFeStream - Validation of a mobile application for visual assessment of cardiopulmonary resuscitation (CPR) acceleration data compared with a life-support manikin sensory,” Masterthesis, FH St. Pölten, St. Pölten, 2016.

[15] S. Loitzl, “Validating a sensor-based mobile health service with real-time data acquisition for improving MPDS communication for lay-rescuer in out-of-hospital cardiac arrest based on a randomized simulation trial,” Masterthesis, FH St. Pölten, St. Pölten, 2016.