Triathlon Trainer: Defining Multiple Activities in a Single Session Using a Smart Watch

Orkun Krand Old Dominion University, Justin Whitlock Old Dominion University

"https://github.com/JustWhit/Triathlon_Trainer"

Abstract

Athletes often do more than one activity in a single workout, especially if they are training for a triathlon, which requires an athlete to transition from one activity to another quickly. Current tracking applications require the athlete to stop and start for each activity, or they offer a method of manually dividing an activity[1], which can be difficult if the athlete did not keep careful track of what they were doing when. Transitions count against the overall time for a triathlon, so taking the time to stop and start an application during a race is inconvenient and not practical. Using the built-in gyroscope and accelerometer of a smartwatch, it is possible to define start and stop times of individual activity sessions, when looking for either cycling or running and possibly for swimming and other activities, and differentiate between activities and transition periods.

1. Introduction

1.1 The Problem: For today's athletes, there are many resources and wearables created to assist with training. Numerous motion studies have covered activity recognition for running, cycling and swimming[1][2][3][4][6][15] using accelerometers and gyroscopes but current applications still depend on the athlete to notify it of what activity they are performing. For most applications, this is sufficient, since athletes tend to do only one activity at a time. An exception to this assumption is the triathlete.

Triathlons are a combination of three activities: running, cycling, and swimming. The order is not fixed, and there is also a transition period between activities which counts against the athlete's overall time. Using an activity tracker which requires you to define a single activity means you must start and stop each activity as they happen. This could increase an athlete's transition time in an unacceptable way, and it also would not show the transition time itself outside of the three activities.

1.2 The Solution: The purpose of this project is to improve on the functionality of current systems by providing a multiple activity option in the context of a triathlon, which would then define the individual activities automatically, as opposed to manual definitions. This would allow users to record once, it would give them feedback on all activities without having to engage with the interface mid-session or manually split it afterwards, and it would allow them to focus on their performance rather than fiddling with their watch.

Using an Apple Watch series 3, we were able to define start and stop times of individual activities within 13.95 seconds on average. The accuracy of running was better than cycling; between 3.2 and 8.5 seconds for runs and 19.41 seconds for rides. We found that the difference in accuracy was due to the similarity between data collected from the watch while a user is walking with their hand on a bicycle's handlebars, rolling the bike along, and when the user is actually riding a bicycle.

2. Previous Work

Activity recognition using wearable sensors is not a new concept. There have been many papers which use accelerometers or gyroscopes to identify or evaluate swimming[15][14][13], running[12][11][10][7], or cycling[1][3][4][6], and some success using a wrist mounted sensor[3][4][6][8][14][15]. These papers asked if it is possible to recognize the activities using the sensors they deployed. In this paper it is assumed that activities are recognizable, and we are asking if it

---

[1] STRAVA is an athletic tracker which gives a user the ability to edit their workout and manually split it into multiple activities after it's been posted "https://www.strava.com/"

is possible to use the recognition of activities, in the context of athletic training or a multi discipline race, to define those activities from start to finish in a meaningful way.

3. Challenges

One of the issues of tracking a workout is that a user may not be doing an activity continuously, yet it should all be considered one activity. For example, a user might be running, and stop for a minute to take a break or tie a shoe, and then continue running. In this circumstance it may appear that the user is about to change activities, but then they continue with what they were doing. In the context of a triathlon, this should all be considered one activity. The second challenge is how to define the transition. There are some things that can be safely assumed to be a part of the transition period, walking, changing or tying shoes, drinking water, but these may also occur during other activities. The third issue involves misclassifications. Athletes may do any number of things during a race that falls outside of the expectations of the training set, resulting in misclassifications: falling off their bike, tripping over a curb, or jumping over a puddle. These misclassifications should be evaluated and discarded based on the context of the classifications around them. Lastly, how to define the beginning and ending of an activity.

4. WatchOS App

Leveraging Swift 4.0 and various extensions (which we will elaborate on), we were able to create a simple application that will collect accelerometer and gyroscope data from the Apple Watch 3 continuously. The main challenge with this was the WatchOS feature that kills processes working in the background, once the screen is off. Another challenge was sending the data from the Apple Watch to the iPhone, which we then accessed via XCode on a Macbook.

In order to save battery, WatchOS doesn't allow most apps to keep running once the screen is off. We tried a couple different approaches to keep our app running in the background and ended up utilizing the Healthkit extension to start a workout session any time a user starts exercising. Healthkit workout keeps all background processes alive as it needs to keep collecting data even when the screen is off. Initial experiments with the Healthkit showed that if the workout session is not terminated, it will drain the watch's battery rapidly. We found, while testing, that even after collecting 2 hours of data via our app, the Apple Watch was able to keep running for the rest of the day without requiring a recharge. So our app is not harsh on the battery life of the watch, it just requires the workout to be terminated when not being used, which is included in our code.

Our other main challenge was transferring the data from the Apple Watch to the iPhone. Because we were processing the data on the computer, we decided to store our data as csv (comma separated values) files to minimize preprocessing. Because of the number of samples we collected per second and how long a triathlon usually lasts, our data files got very big very fast. The initial transfer method we used: creating a dictionary to store data and then transferring it using the application context of the WatchConnectivity framework then converting the dictionary into a csv file on the iPhone weren't fruitful as sending a dictionary of that size wasn't allowed. So instead, we decided to create the csv file on the Apple Watch and send that using the transferFile method of the same framework. This method proved to be reliable, successful, and fast. We tested with data of various sizes to ensure it wouldn't crash or ignore portions of data.

Our WatchOS application is a very minimalist design with only a button to start and stop the workout because this was more of a proof of concept application than a final product. There is also a timer to show how long it has been since the workout started and labels showing the axis values of the accelerometer and the gyroscope. All the features except the start-stop button have been disabled for testing purposes. Overall, the application would benefit from a UI redesign to make it more user friendly. Same goes for the counterpart iOS application which currently shows a blank screen. In the current version, a user only knows the workout has started or stopped via the text on the button which changes from “Start Recording” to “Stop Recording” and vice versa when clicked.

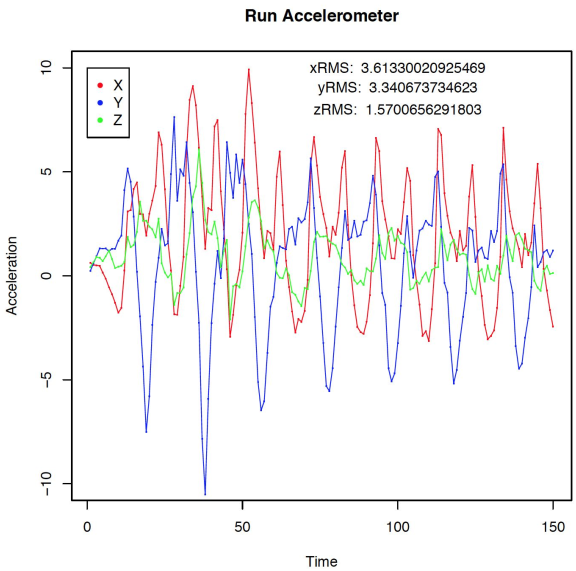

Figure 1. X, Y and Z output from the Apple Watch's accelerometer over 5 seconds, collected at 30 Hertz, during a run.

5. Data Collection

5.1 Experiments: We had three volunteers, two male and one female. Each of the volunteers was in their early forties, and all of them had trained for and completed a triathlon in the past. Each volunteer completed a combined workout, featuring a run and a ride, in any order. Each activity was about 10 minutes long, with a short transition period between the two, for a maximum of 25 minutes. We then had each volunteer record 5 minutes of running and five minutes of cycling in separate sessions for training data. In order to classify when a user is neither running nor cycling, we recorded 5 minutes of what the volunteers would do during a transition period in a triathlon. This consisted of changing their shoes, tying their shoes, doing stretches, walking around to catch their breath, sitting down, and drinking water. In addition to the volunteers, Justin was recorded doing a single workout session which included a ride, a run, and a second ride. Justin did not contribute to the training data.

5.2 Data: The data we collected consisted of the x, y and z axis from both the accelerometer (see **Figure 1**) and gyroscope on the Apple Watch, at a rate of 30 Hertz. Each of the combined sessions was recorded using a GoPro Hero 4 Silver at 1080p - 60 frames per second. We recorded when the user pushed the start button on the watch and their subsequent activities and transitions. These recordings were used to establish ground truth.

6. Data Analysis

6.1 Classification: In order to classify what the user was doing, we used a common sliding window technique[1][3][4][5][7][8][11][14][15], shifting the window by 1 second, or 30 data points. The initial window size used was larger, around 5s, but a 3s window was also tested for comparison. The 3s window gives 66% overlap, while the 5s window gives 80% overlap. The difference in the results, however, was negligible. The accuracy of our SVM for the 3s window was 85.1% on average, and 84.3% for the 5s window. With less than 1 percentage point difference, we saw no reason to spend the extra computational power on the larger window.

From each window of raw data several features were extracted for each axis of both the accelerometer measurements and the gyroscope measurements: root mean square, standard deviation, mean, median, avg distance between peaks, the standard deviation of the distance between peaks, the average amplitude between peaks and the standard deviation of the amplitude between peaks. The indexes of the peaks are extracted using peakutils, a Python library, with a threshold of 0.02 divided by the maximum value in the window, and a max distance of 2 data points. The resulting vectors for each window are then processed with a one vs. all SVM from the SKlearn library in Python using a linear kernel.

6.2 Detection Algorithm: After the SVM model had been trained, the SVM was applied to each activity from the volunteers. The resulting list of predictions for each window was combined with the beginning and ending timestamps for those windows. This gives a second by second timeline for the workout. This timeline is then processed to determine the number of activities in the workout, what type of activities occurred, and define the individual start and stop times of those activities using the decision tree in **Figure 2**, which uses a set of counters to evaluate each windows classification in relation to what comes before the window on the timeline. The counters are weighted differently, for example: The Tcounter, which counts consecutive appearances of transition classifications, must reach at least 10 before it begins to weigh on the decision making, while other counters

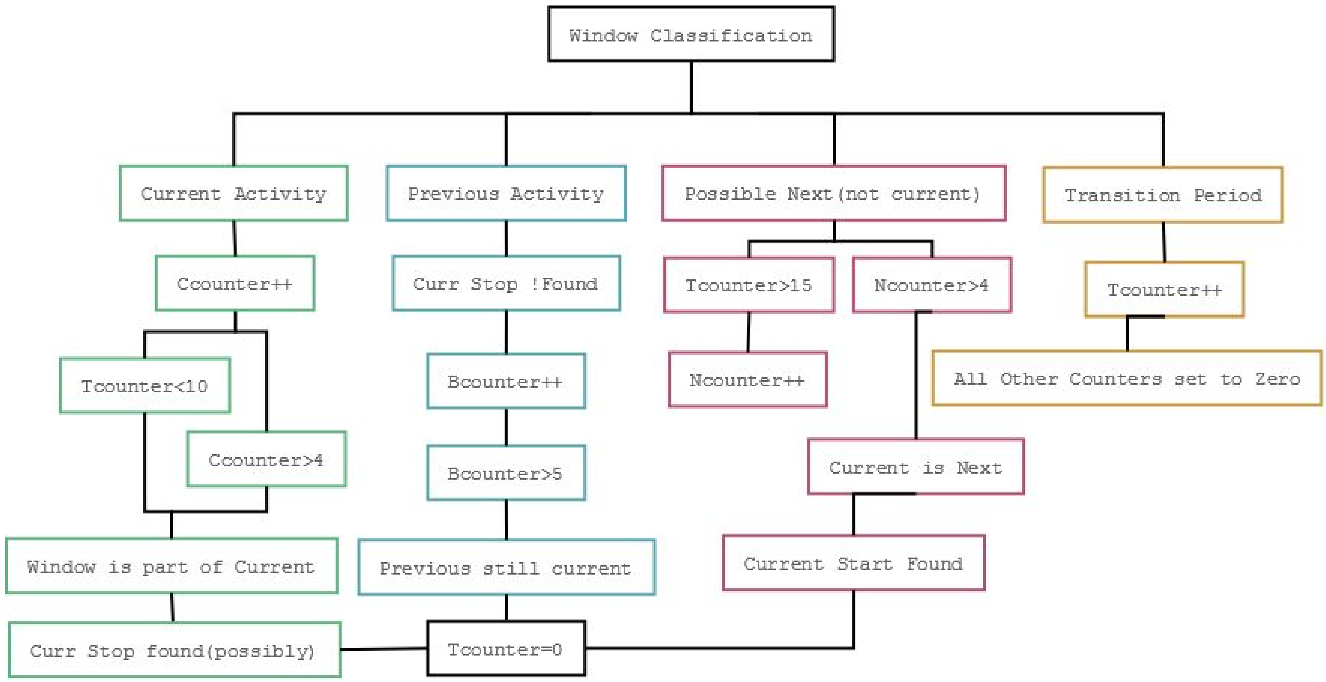

Figure 2: Decision Tree for Activity Definition using a set of counters: Ccounter for Current Activity, Tcounter for Transition Detection, Bcounter for step back in Activity detection, Ncounter for possible Next Activity.

come into play at 4 or 5. Appearances of transition however, reset all counters to 0. The Bcounter, or back counter, became necessary due to variances in the collected data. While an athlete may be doing a run workout, that doesn't necessarily mean they are running the entire time. The path that the volunteers used for data collection of their combined workouts had other runners and walkers and sometimes whole families with pets utilizing the walkway. This required the volunteers to slow down, walk, or navigate around these obstacles. Due to the limitations of our training data, which was collected in an empty parking lot, some of these instances resulted in misclassifications by the SVM. Under the right circumstances, these misclassifications could be interpreted as a new activity. The Bcounter allows the algorithm to step back to the old activity if the data indicating a new activity seems weak.

7. Results

7.1 Volunteers: The SVM accuracy for each volunteer's combined workout averaged 84%. This was calculated by SKlearn's builtin score function. It uses the one vs. all prediction of the SVM and compares it to the ground truth taken from the GoPro footage. The training data from all three volunteers was used for each classification.

For each volunteer, the algorithm described in **Figure 2** was able to accurately determine the number of activities, the type of activity and the order in which they were completed (see **Figure 3**). Start and stop times for each activity were predicted to be within 13.83 seconds of the ground truth on average. This result was not within the range we were hoping for, which was between 3 and 5 seconds. To understand why, we compared the accuracy of the run predictions with the bike predictions. We found that the run predictions were within 8.49 seconds of groundtruth on average, and the bike was within 19.41 seconds. According to the video, the point at which the algorithm begins predicting the user is riding a bicycle correlates to when they place their left hand (the hand with the watch) on the handlebars and roll the bicycle forward. The correlation was made for all three volunteers, and it is likely that the data for rolling the bike while holding the handlebars and actually riding the bike are very similar. This may be overcome by improvements in the training data, or through improvements in feature extraction. Another anomaly was found while reviewing the recordings of the run portion of the combined workouts. One of the

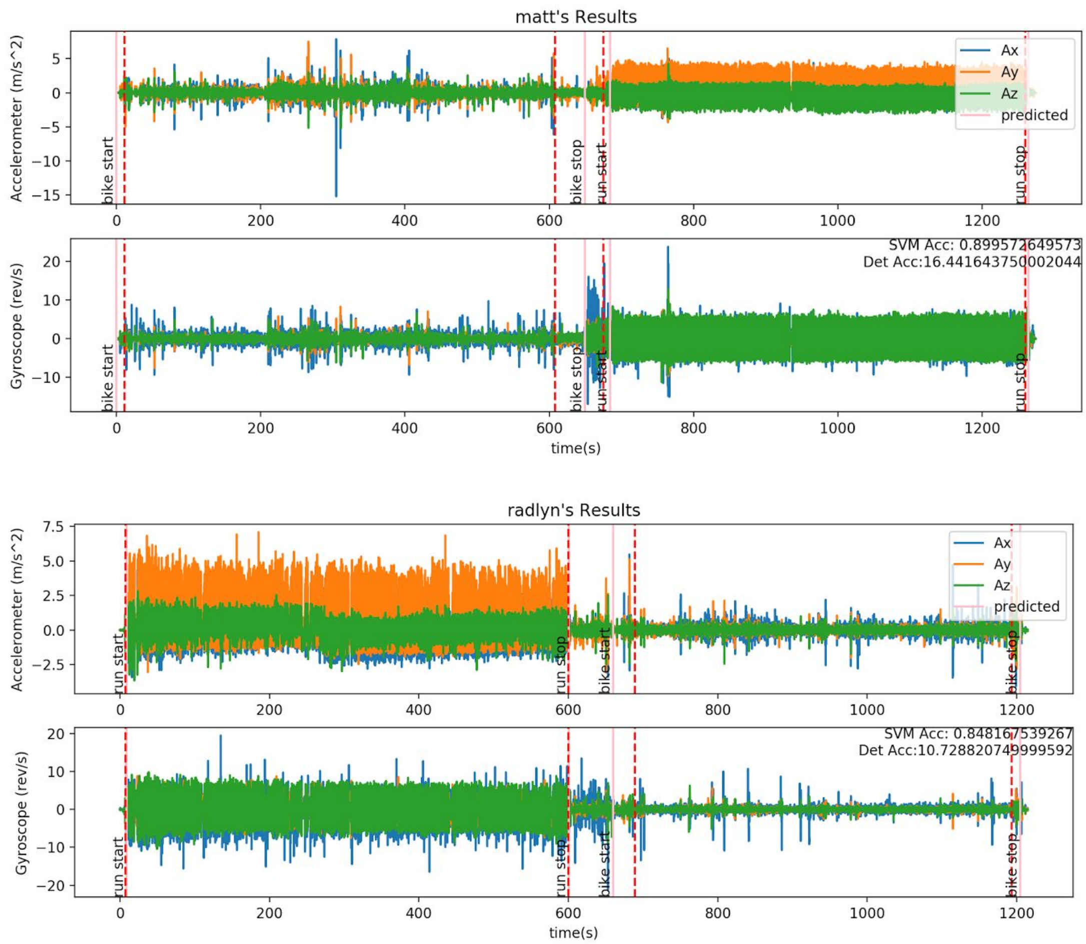

Figure 3: The results from two volunteers. The pink line indicates the predicted start and stop times for each activity and the red dashed line indicates the groundtruth. Both bike and run were correctly identified. The SVM accuracy as a % and the Detection Algorithm accuracy in seconds are listed in the top right for each volunteer's workout.

volunteers spent the last 20 seconds of their run holding their left arm up and looking at the watch. A change in orientation of the watch while running was not included in the training data, so the SVM classified the last 20 seconds as transition, and the detection algorithm ended the run 20 seconds early. Again, an improvement on coverage of the training data would likely solve this issue. When we exclude this volunteer's end time from the run statistics, the accuracy of the algorithm improves to 3.2 seconds, which is in the target range for the system.

7.2 User not Included: As a further test of the system, Justin's workout with 3 activities was processed (**Figure 4**). Given that Justin did not contribute to the training data, the results of 88% for the SVM and 9.3 seconds for the detection algorithm are encouraging, as these are comparable to the average result from the three volunteers, if not a little bit better.

8. Conclusion

We were able to successfully determine if an athlete is running, cycling, or in the transition period. While we observed some unexpected results (a window is classified as cycling as long as hand is on bicycle or holding the left arm up while running is classified as transition), but overall our system was able to give a rough estimate of which activity happened when. We

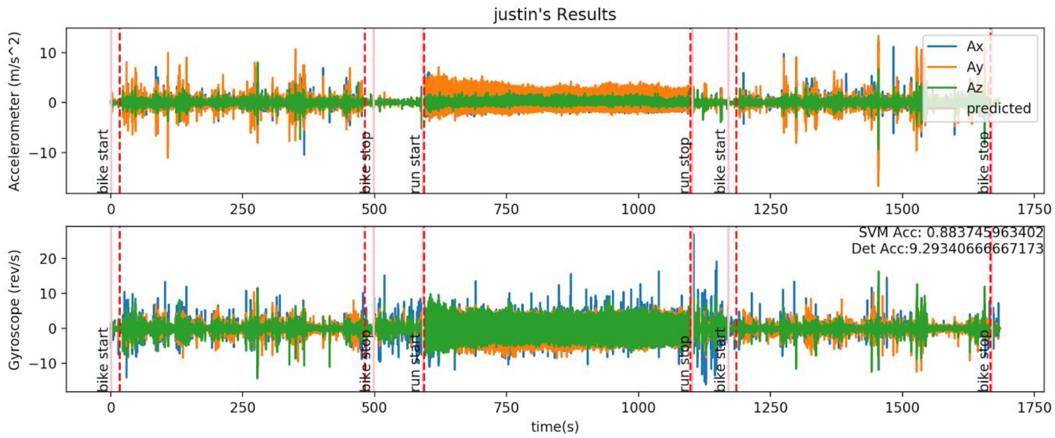

Figure 4: Justin's results from a bike-run-bike combination workout. The SVM accuracy in % and the Detection Algorithm accuracy in seconds is on the right. The pink lines indicate predicted start and stop times for each activity. The dashed red lines indicate the groundtruth.

believe better coverage for our training data would improve the results. Wearing the smartwatch on the ankle instead of the wrist may may be able to differentiate between walking a bicycle and riding as the legs would produce patterns that might be easier to distinguish.

Acknowledgements

We'd like to thank our intrepid volunteers, who showed up despite the cold weather and the wind. We'd also like to thank Dr. Shubham Jain for her valuable contributions.

REFERENCES

[1] Juha Pärkkä, Miikka Ermes, Panu Korpipää, et al, "Activity Classification Using Realistic

Data From Wearable Sensors" IEEE Transactions on Information Technology In Biomedicine, Vol. 10, No. 1, January 2006. Available: http://ieeexplore.ieee.org/abstract/document/1573714/

[2] Seon-Woo Lee, Kenji Mase, "Activity and Location Recognition Using Wearable Sensors" IEEE Pervasive Computing (Vol. 1, No. 3, July-Sept 2002. Available: http://ieeexplore.ieee.org/abstract/document/1037719/

[3] Stephen J Preece, John Y Goulermas, Laurence P J Kenney, et al, "Activity Identification Using Body-Mounted Sensors--A review of Classification Techniques" Institute of Physics and Engineering in Medicine: Physiological Measurment, Vol. 30, No. 4, 2 April 2009. Available: http://iopscience.iop.org/article/10.1088/0967-3334/30/4/R01/meta

[4] Andrea Mannini, Stephen S. Intille, Mary Rosenberger, et al, "Activity recognition using a single accelerometer placed at the wrist or ankle" Med Sci Sports Exerc. Nov. 2013. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3795931/

[5] Xing Su, Hanghang Tong, and Ping Ji, "Activity recognition with Smartphone Sensors" Tsinghua Science and Technology, Vol. 9, No. 3, June 2014. Available: http://ieeexplore.ieee.org/abstract/document/6838194/

[6] Miikka Ermes, Juha Pärkkä, et al, “Detection of Daily Activities and Sports With Wearable Sensors in Controlled and Uncontrolled Conditions” IEEE Transactions on Information Technology in Biomedicine, Vol. 12, No. 1, Jan 2008. Available: http://ieeexplore.ieee.org/document/4358887/

[7] J. K. Aggarwal, M. S. Ryoo, “Human Activity Analysis: A Review” ACM Computing Surveys (CSUR), Vol. 43, No. 3, April 2011. Available: http://dl.acm.org/citation.cfm?id=1922653

[8] Andrea Mannini, Angelo Maria Sabatini, “Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers” Sensors, Vol. 10, No. 2, February 2010. Available: http://www.mdpi.com/1424-8220/10/2/1154/htm

[9] Tanmay Pawar, N. S. Anantakrishnan, Subhasis Chaudhuri, et al, “Transition Detection in Body Movement Activities for Wearable ECG” IEEE Transactions on Biomedical Engineering, Vol. 54, No. 6, June 2007. Available: http://ieeexplore.ieee.org/document/4203022/

[10] O. Thomas, P. Sunehag, G. Dror, et al, “Wearable sensor activity analysis using semi-Markov models with a grammar” Pervasive and Mobile Computing, Vol. 6, No. 3, June 2010. Available: http://www.sciencedirect.com/science/article/pii/S1574119210000064

[11] Jie Yang, Shuangquan Wang, Ningjiang Chen, et al, “Wearable accelerometer based extendable activity recognition system” 2010 IEEE International Conference on Robotics and Automation (ICRA), 3-7 May 2010. Available: http://ieeexplore.ieee.org/abstract/document/5509783/

[12] Pekka Siirtola, Juha Röning, “Recognizing Human Activities User-independently on Smartphones Based on Accelerometer Data” International Journal of Artificial Intelligence and Interactive Multimedia: Special Issue on Distributed Computing and Artificial Intelligence, Vol. 1, No. 5, 2012. Available: http://www.ijimai.org/JOURNAL/sites/default/files/IJIMAI20121_5_5.pdf

[13] Neil Davey, Megan Anderson, Daniel A. James, “Validation trial of an Accelerometer-based sensor platform for Swimming” Sports Technology, Vol. 1, No. 4-5, 8 December 2008. Available: http://onlinelibrary.wiley.com/doi/10.1002/jst.59/full

[14] Marc Bächlin, Gerhard Tröster, “Swimming performance and technique evaluation with wearable acceleration sensors” Pervasive and Mobile Computing, Vol. 8, No. 1, Feb 2012. Available: http://www.sciencedirect.com/science/article/pii/S157411921100071X

[15] Pekka Siirtola, Perttu Laurinen, Juha Röning, Hannu Kinnunen, “Efficient Accelerometer-based swimming exercise tracking” 2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), 12 july 2011. Available: http://ieeexplore.ieee.org/abstract/document/5949430/

[16] Cliff Randell, Henk Muller, “Context Awareness by Analysing Accelerometer Data” Dept of Computer Science, University of Bristol Aug 2000. Available: https://www.cs.bris.ac.uk/Publications/Papers/1000463.pdf

[17] Davide Figo, Pedro C Diniz, et al, “Preprocessing Techniques for Context Recognition from Accelerometer Data” Springer-Verlag London Limited, Online, 30 March 2010. Available: https://dl.acm.org/citation.cfm?id=1872975