# Swimming Stroke and Count Detection

Mauricio A. Rovira Galvez
University of Victoria
Victoria, BC, Canada
marovira@uvic.ca

Liam Day
University of Victoria
Victoria, BC, Canada
liamday@uvic.ca

Nathan Harmsworth
University of Victoria
Victoria, BC, Canada
NHamswo@uvic.ca

## ABSTRACT

Due to the rise of activity trackers, recognition and acquisition of data for user athletic activities has become more relevant. In this paper we propose a machine learning system that can recognize and identify swimming strokes based on accelerometer data.

## KEYWORDS

swimming, stroke, detection, accelerometer

**ACM Reference Format:**
Mauricio A. Rovira Galvez, Liam Day, and Nathan Harmsworth. 2017. Swimming Stroke and Count Detection. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION

With the rise of fitness trackers such as [2] and [4], there has been an increased interest in systems that can determine not just the activity that the user is performing, but also provide additional information such as distance, heart rate, calories burned, amongst others. As a result, wearable trackers come equipped with a variety of sensors that are used to detect activities and provide useful information. These sensors include gyroscopes, accelerometers, GPS, heart-rate sensors, etc.

Swimming is a sport that, due to the nature of the environment it takes place in, presents with a variety of challenges for activity and feature identification. Indeed, most heart rate monitors do not work when submerged, as water is not a conductive medium. Fortunately, sensors such as accelerometers and gyroscopes still function within a properly sealed unit. As a result, these tools have been used to detect information such as stroke types and counts, laps, speed, and calories. This is possible due to the unique cadence and patterns that are associated with each one of the 4 competitive strokes: freestyle, backstroke, breaststroke, and butterfly. By recognizing these patterns, trackers are able to determine with high degrees of accuracy what the user is doing in the water, even if their technique is not perfect.

In this paper, we present a technique for identifying stroke types based on accelerometer data captured from an Android device.


*Conference'17, July 2017, Washington, DC, USA*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 2 RELATED WORK

Several studies have been made that utilize data mining techniques to identify athletic activities. [3] aims to measure outdoor athletics in the long term. The study focuses on using android and external sensors to gather information. The authors put special interest into sensors on the feet and GPS tracking. [1] focuses on capturing and identifying sports with the information provided by consumer-level sports trackers.

In [6] the authors use a chest mounted accelerometer to gather information about swimmers. They use motion in the Y-axis to identify traits like whether the swimmer is moving or resting and when the swimmer has completed a lap. A decision tree was developed by the authors in order to classify the swim stroke used from the accelerometer data. On the other hand, in [7] they use an accelerometer attached to the right palm of a swimmer to collect the acceleration in the X, Y and Z directions for different strokes. The data is then analyzed to find out the stroke count and stroke style. Using graphs created from the data they were able to see the trends in the acceleration for each stroke style.

In this paper, we present a system that is based on the one presented in [7]. Accelerometer and gyroscope data will be captured and then processed. Once this step is done, we will use the techniques outlined in [8] and [5].

## 3 IDENTIFYING STROKES

### 3.1 Overview

The idea used in this paper is not too dissimilar to other techniques employed in the field of data mining. In essence, the process remains as follows:

(1) Capture data,
(2) Process data to extract features,
(3) Train model with data,
(4) Test model with new data.

This process is shown in Figure 1. We will now examine each step in more detail.

### 3.2 Data Acquisition

Acquiring accelerometer data from any land-based activity is a fairly simple task, as all that is required is to attach a sensor to the test subject and capture it's output. This can usually be done by either having the subject hold the sensor, or attaching it by using simple straps. The data capture step can be done fairly easily via bluetooth or internal storage.

Due to the fact that swimming takes place in water, this presents with numerous challenges that need to be overcome. First and foremost is that the device must be encased in such a way to make it water-proof without making it too large or awkward to position

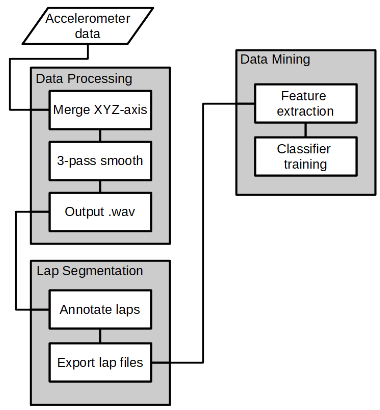



**Figure 1: The process of capturing, analyzing, and classifying swimming data.**

on the swimmer's body. This leads to the problem of attaching the sensor. The fact that water has a higher density than air implies that the friction that is generated between the water and the swimmer is significantly greater than say, a runner against air. Any additional surfaces on the swimmer's body will experience the same friction, which makes attaching the device more complex, as motions such as pushing off the wall increase resistance dramatically and could very well rip the device off.

In order to avoid having to design custom hardware, we decided to employ a Samsung Galaxy s5 Neo as our sensor. The data was captured using an app called Accelerometer Analyzer, which saves the output to a simple text file stored on the device. A water-proof casing was produced by vacuum-packing the phone, which ensure that it would be able to capture data, as well as allowing the swimmer the ability to interact with the screen to start and end the data capture. To solve the problem of attaching the device, the swimmer was required to hold the phone perpendicularly across the palm while pinning it between a paddle and their hand. This allows a much more even distribution of forces when swimming, and makes the experience easier to manage. The setup can be seen in Figure 2.

The swimmer starts the data acquisition prior to placing the phone atop the paddle. While this does result in additional noise being generated, it is easily identified and removed in later steps. Once the paddle is in place, the swimmer proceeds with their workout by swimming a specified distance employing a specific stroke. As soon as the set is complete, the swimmer removes the paddle and stops the data recording. The app saves the file with the data into a compressed text file which can then be transferred to a computer using a USB cable.

## 3.3 Data Processing and Feature Extraction

Once the data has been transferred to a computer, it can then be analyzed and the relevant features extracted. A sample of the captured data can be seen in Figure 3.

The purple sections at the beginning and end of the plot represent the noise that was generated by placing and removing the phone from the paddle. The red section outlines a single lap, while the

**Figure 2: The setup used to capture data. The phone is packed and then placed horizontally across the swimmer's hand, pinned between a paddle and the palm.**

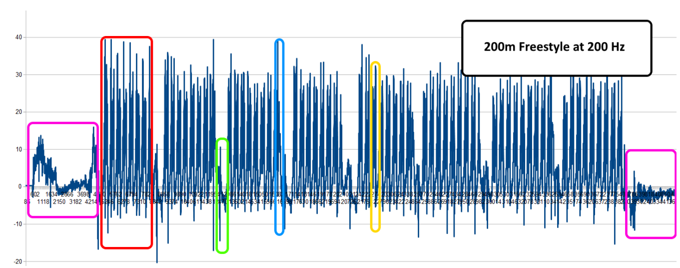


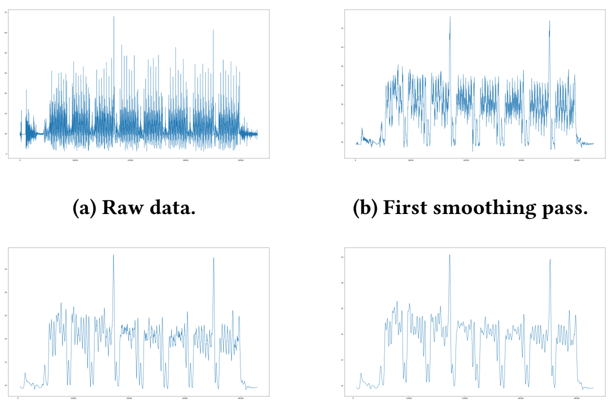

**Figure 3: A sample of the X-axis accelerometer data for freestyle.**

blue and yellow sections are a single stroke. Finally, the green area is the flip turn that occurs at the end of each lap.

It can be seen that the data is reasonably periodic, and in fact closely resembles an audio signal. With this in mind, the data processing can be divided into the following steps, which will be discussed below:

(1) Data smoothing,
(2) Lap segmentation, and
(3) Feature extraction.

In order to make the data capture as precise as possible, the device samples the acceleration sensors at a rate of 200Hz. While this leads to more accurate readings, it does introduce a high amount of noise, which would ultimately add imprecision to the training data. In order to solve this, the data is smoothed using a 3-pass rolling mean. This retains the general shape of the wave-form but removes any unnecessary data. The process can be seen in Figure 4.



**(a) Raw data.** **(b) First smoothing pass.**

**(c) Second smoothing pass.** **(d) Third smoothing pass.**

**Figure 4: The progressive smoothing for a 200 backstroke.**

Once the data is smoothed, the next step is to segment it into laps. This has a dual purpose: first it increases the data size by a factor of 2 (as every set is ultimately a multiple of 50m and the data was acquired in a 25m pool) which helps to get more accurate training results. The second benefit is the removal of the flip-turn data, which for the purposes of stroke identification can be considered as noise. The segmentation is done manually using Audacity by adding labels at the appropriate intervals. The data is then exported into separate files.

The final step is to extract the features from the laps, which is done by using [9]. The data from each of the strokes is combined into a single file, which is then processed into a `.arff` file which can then be sent to Weka for training and validation.

### 3.4 Training and Validation

The resulting `.arff` file is loaded into Weka, were several classifiers are tested for performance. This allows us to obtain an accuracy range for comparison. The used classifiers are:

- ZeroR,
- Naive Bayes,
- Bayes Net,
- SMO,
- J48, and
- Random Forest.

The result can be seen in Table 1.

## 4 DISCUSSION

We begin our discussion by noticing that, with the exception of the ZeroR classifier, all of the F-scores are above 90%. This can be attributed to the following reasons:

(1) The data set is too small, and
(2) the data was captured from only one swimmer.

| Classifier | Backstroke | Freestyle | Average |
|---|---|---|---|
| ZeroR | 0.44 | 0.286 | 0.365 |
| Naive Bayes | 0.970 | 0.968 | 0.969 |
| Bayes Net | 0.970 | 0.968 | 0.969 |
| SMO | 1 | 1 | 1 |
| J48 | 0.933 | 0.941 | 0.937 |
| Random Forest | 1 | 1 | 1 |

**Table 1: The F scores for each classifier over each stroke, and their average.**

The first problem is directly tied to the way that the data is captured, as the setup is far from ideal, as the paddles increase the resistance that the swimmer encounters when swimming, thereby limiting the number of sets that can be recorded. Initially, the phone was strapped to the swimmer's palm employing tape. While this does work reasonably well, it does not account for the unevenness that is caused by the phone's flat surface. In fact, swimming in this manner is not too dissimilar to swimming with a single paddle. This results in an imbalance in the stroke and increased stress on the corresponding shoulder. In addition, the added weight of the phone on the swimmer's hand affects the recovery phase of freestyle, backstroke, and butterfly. In this phase, which is when the arm is outside the water, the swimmer usually does not carry any significant weight, and it is meant to represent a break from the constant pulling force that is applied throughout the stroke. By adding weight, it increases stress on the swimmer's shoulder joint, thereby limiting the time that the swimmer can perform in these conditions.

The idea of employing paddles solves the problem of the unevenness in the stroke as it evenly distributes the load between both shoulders, but fails to address the weight problem. In addition, this also limits the strokes that the swimmer can perform, as butterfly and breaststroke apply considerable pressure at the front of the stroke, which is currently beyond what the swimmer can handle.

The fact that the data was captured from only one swimmer is also related to the previous point. Due to the fact that the entire data has to be recorded using paddles, this limits not only the distances, but the people that could be used for recording. The paddles introduce a high amount of resistance, which unfortunately if is not properly handled, can lead to shoulder injuries. This, in addition to time constraints, severely limited the number of people available for recording data.

The methodology as described here is also far from ideal, as the following problems were encountered:

(1) Lap segmentation, and
(2) Stroke counting.

Examining the plots for the data, it is relatively easy to see where the flip turns take place and where the division between laps should happen. Since the entire problem can be treated as a signal analysis or music problem, we attempted to use segmentation algorithms to separate into sections. While this was somewhat successful in freestyle, it was unable to segment a backstroke set. This is attributed due to the variability that exists on the last stroke prior to the flip turn. Depending on which arm is used, there could be a normal spike, which corresponds with the rest of the laps, or there could be a spike that is significantly higher than the rest. This behaviour can be seen in Figure 5. As this behaviour follows no discernible pattern, a segmentation algorithm would be unable to detect the different sections. This added to the somewhat irregular nature of the first stroke coming off the wall makes for very poor segmentation results in backstroke. As a result, manual segmentation had to be used.

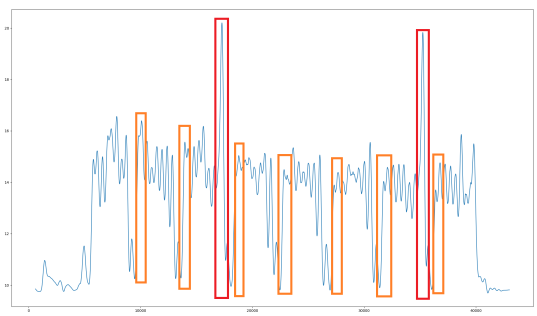


**Figure 5: The data for 200 backstroke. The red regions represent the last strokes where the arm that had the phone was used. The orange regions are the first strokes coming off each wall.**

In terms of stroke counting, the irregularity of the individual strokes makes this very difficult to solve. While there is a certain regularity in the data, any increases in stroke rates will completely change the wave-form, which invalidates any attempt at using preset values and thresholds. As a result of these two major difficulties, these features were left out.

## 5 FUTURE WORK

The approach presented here as some severe flaws, and also cannot be used in a more generic context. As a result, the following improvements, along with their possible solutions, need to be performed:

- Better data acquisition. This can be solved in one of two ways. The first is by using a device such as an Apple or Samsung watch which do provide access to the raw accelerometer data, or by the creation of a custom device. This would remove the weight problem by allowing the swimmer to wear a regular watch, and therefore paddles would no longer be necessary. This means that more data can be captured per session, as well as a wider variety of strokes and swimmers. In addition, depending on the recording method used, it would dramatically reduce the noise that exists at the beginning and end of each recording session.
- Better lap detection. The current setup does not scale well. If the previous point is implemented, then the amount of data that can be captured will increase dramatically, making manual tagging impractical. As a result, the following solution is proposed: in a separate recording session, record the data that corresponds to both flip and open turns. A classifier can then be trained to identify whether a turn exists in a given set, how many, and where they are located. This information is then used to perform the segmentation.
- More data. This comes as a direct consequence of the previous two points. By training on a larger and more varied data set, the accuracy scores would become more relevant. It would also enable different features to be extracted from the data and tested to see which one delivers the best results.
- Stroke counting. This can be done as a byproduct of the classifier training. Depending on the extracted features, the classifier could readily be able to determine the number of strokes that exist within a single lap. The calling system would just need to aggregate these numbers to produce the final result.

## 6 CONCLUSION

In this paper, we have proposed a design for a system that is able to recognize strokes based on accelerometer data captured from an Android device. In spite of it's flaws, the system is able to produce data that can be easily classified and produces good results.

## REFERENCES


[1] Iztok Fister, Duan Fister, and Simon Fong. 2013. Data mining in sporting activities created by sports trackers. In *Computational and Business Intelligence (ISCBI), 2013 International Symposium on.* IEEE, 88–91.

[2] Fitbit Inc. 2017. Fitbit Trackers. (2017). https://www.fitbit.com/en-ca/home.

[3] P Kugler, Dominik Schuldhaus, Ulf Jensen, and B Eskofier. 2011. Mobile recording system for sport applications. In *Proceedings of the 8th international symposium on computer science in sport (IACSS 2011), Liverpool.* 67–70.

[4] Garmin Ltd. 2017. Garmin Swim Watch. (2017). https://buy.garmin.com/en-CA/CA/p/97287.

[5] Kevin P Murphy. 2012. *Machine learning: a probabilistic perspective.* MIT press.

[6] Yuji Ohgi, Koichi Kaneda, and Akira Takakura. 2014. Sensor data mining on the kinematical characteristics of the competitive swimming. *Procedia Engineering* 72 (2014), 829–834.

[7] Meng-Shiuan Pan, Ku-Chen Huang, Tzu-Hsiu Lu, and Ze-Yan Lin. 2016. Using accelerometer for counting and identifying swimming strokes. *Pervasive and Mobile Computing* 31 (2016), 37–49.

[8] Pang-Ning Tan et al. 2006. *Introduction to data mining.* Pearson Education India.

[9] George Tzanetakis. 2006. Marsyas (Music Analysis, Retrieval and Synthesis for Audio Signals). (2006). http://marsyas.info/index.html.