



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 Issue: VI Month of publication: June 2021

DOI: <https://doi.org/10.22214/ijraset.2021.34986>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Counting Repetitions using Sensor

Dr. Pooja Raundale¹, Prof. Harshil Kanakia², Devang Chhajed³, Taiyeeba Chikhalia⁴

¹H.O.D. Masters of Computer Application, Sardar Patel Institute of Technology, Mumbai, India

^{2, 3, 4}Master's of Computer Application, Sardar Patel Institute of Technology, Mumbai, India

Abstract: This paper describes the step by step process for counting the number of repetition using different Mobile Phone Sensors. The Algorithm is tested on various datasets acquired by the people. To keep it simple and accessible to low-end devices, the algorithm needed to be simple and fast. We identify the most volatile axis, and then smooth the curve to remove any noise, 1 cycle being the 1 complete motion we calculate the number of complete cycles by the axis wave which gives us the total number of repetitive motions completed.

Keywords: sensors, motion sensors; accelerometer; gyroscope; linear acceleration; gravity; magnetometer; mobile phone sensor

I. INTRODUCTION

In recent years, the world has seen a boom in fitness bands like mi band, fitbit, amazebit, etc. which ranges from Rs. 2000 to Rs. 6000. Its hardware is fixed and cannot be updated[1]. Today everyone owns a smartphone, which is a very powerful device bundled with various types of sensors for measuring the acceleration, rotation, gravity, proximity, etc. With the boom in the smart wearables market, they have become an integral part of people's life. Our paper aims to leverage the sensors in them to detect the repetitive motion of the user[2]. This data can further be used for analysis.

II. LITERATURE SURVEY

There are various ways to count the repetitive motions which are currently being used which are based on Dynamic Time Warping(DTW)[3], neural network-based detection[4], etc[5][6][7]. Most of them require either a machine learning algorithm or some part of machine learning algorithm which require a lot of processing power.

Most of the current methods to count repetitive motion compares two waves which occur consecutively to detect motion. It needs an activity(motion) to compare the user's current activity with, which requires a user to pre-select the activity which they want to track.

III. THE PROCESS

A. Data Collection

We developed an application to log the user movement and asked different people to perform various activities. We recorded data of 3 main sensors i.e. accelerometer, gyroscope, rotation sensor [8][9][10].

B. Re-align all the axis to start from the same point

We use various sensors such as accelerometer, gyroscope, rotation sensor, etc to capture movements. But as we can see in fig1 below, all three axes start from a different position. Hence deciding which axis to work on becomes a little harder since every axis is starting from a different point. Therefore, we need to get all the axes to start from the same point, i.e. 0.

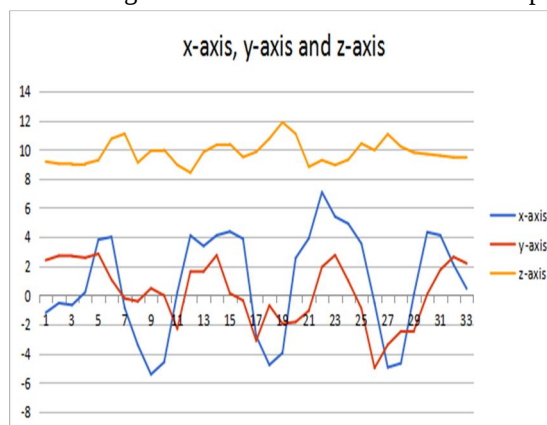


Fig 1: Raw Data from Accelerometer for moving left-right 3 times.

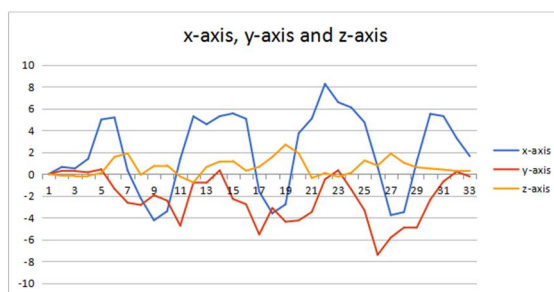


Fig 2: Same Data after realigning

C. Find Maximum Travelled Axis

We need to find the maximum travelled axis out of the three axes since maximum movement (curves) is found on that axis. Once we find which axis has been used the most i.e. has been travelled the most, it gets easier to detect the repetitive motion and get count[11]. To find the maximum travelled axis we add the absolute value of points in an axis (i.e. |value|) to get the total distance travelled by an axis.

For example, in fig 2, the maximum travelled axis would be x-axis.

D. Smooth the Graph

First, We smooth the graph by taking the average of $i-1$, i , $i+1$ points and assigning it to i for all i greater than 1 and less than length of $x-1$.

$$\sum_{i=2}^{i=n-1} ((axis[i-1] + axis[i] + axis[i+1]) / 3)$$

Smoothening the graph will eliminate the basic noise from the graph.

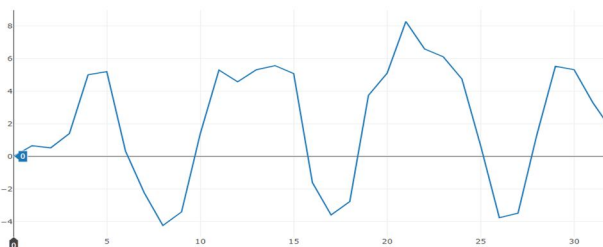


Fig 3 : Un-smoothed graph

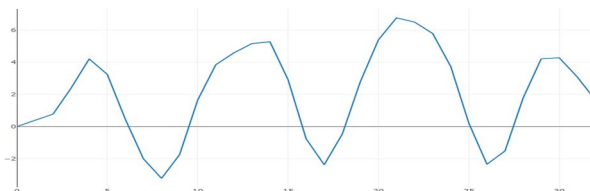


Fig 4 : Smoothed graph

We need to keep in mind the number of times a graph is smoothened as smoothening it too much will cause a lot of points from the graph to be eliminated thus changing the original shape of the graph which will further result in incorrect counting of repetitive motions. While implementing this algorithm we found that we can smooth the graph a maximum of 3 times to eliminate as much noise as possible without changing the actual shape of the graph too much.

E. Get Positive and Negative peak Points(no of max/min points)

Get the peak points from both positive and negative quadrants[12]. We get the peak points to count the repetitive motions. A motion from the graph is found by detecting cycles in it. A cycle consists of at least one positive peak and at least one negative peak. It is not necessary that a cycle will have only one positive peak and one negative peak. Hence we apply smoothening so that getting cycles to get easier. Detecting the peak points gives us a rough idea of the cycle in the graph.

F. Get Average Speed of the Graph

We need a line from where we can calculate whether the graph is going up or down, i.e. to calculate positive peaks and negative peaks since it is not necessary that the zero axis would be the centre of the graph. To understand it better consider fig 3 and fig 5. Fig 3 is a graph which has a line of reference as the zero axis. Consider fig 5. If we consider the zero axis as the line of reference for calculating peaks, the 2nd and 3rd negative peak might be considered as noise by the computer since they barely cross the zero axis.

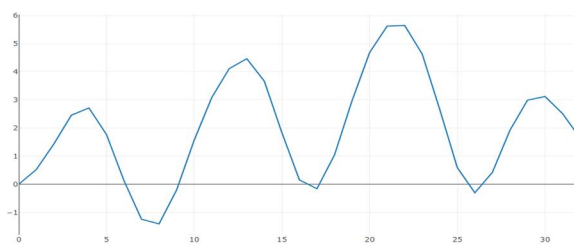


Fig 5: Graph whose line of reference isn't zero axis

Sometimes it may happen that the amplitude of the peaks above/below the zero axis may be as small as to consider it as noise as in the above example. In these cases we might consider noise too as a part of the cycle or we may skip a part of the cycle by misunderstanding it as noise resulting in an inaccurate count. Hence we need to find a new line of reference where we can differentiate between positive peaks and noise. To get the line of reference where the peaks should be calculated from, we find the average speed of the graph, i.e. the new line of reference.

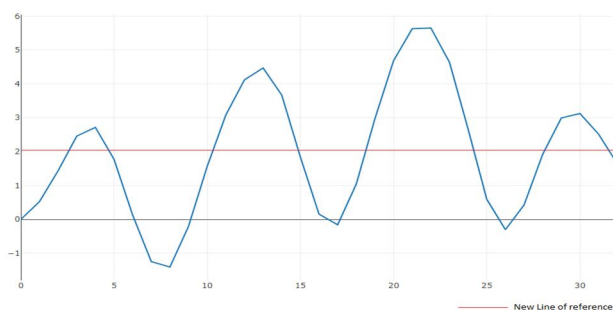


Fig 6: New line of reference

G. Take Average of the Peaks

There is still noise present in the graph. So, while finding peaks it is possible that noise may be counted too. Since we calculate the count using peaks, noise might be included too resulting in inaccurate count. We can differentiate noise and the actual movements by the amplitude. Usually the amplitude of a particular activity is much higher than that of noise. Hence, to avoid counting noise we need to count only certain peaks which have amplitude greater than a certain value. To find that value we take the average of both the positive and negative peaks respectively to get the median of both sides from the new line of reference. We then consider peaks crossing the positive median and negative medians only while counting cycles.

H. Calculate Threshold with 0.5 Error Rate

Since thresholds are calculated by taking the average of the peaks it is possible that some peaks which are a part of the cycle might not cross the threshold i.e. there is a chance for an error. We need to consider those peaks too to get an accurate count. Hence we took 0.5 as the error rate and subtract it from the positive threshold and add it in the negative threshold i.e. the peaks which miss the threshold by error rate will also be considered for the cycles.

I. Check for Peaks Crossing Threshold

This step will eliminate the points generated by the noise. Count the peaks from above step to get the number of cycles i.e repetitive movements.

IV. RESULT

We tested various activities and movements to get count.

eg,

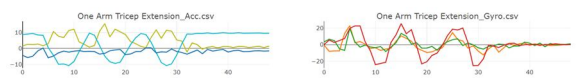
A. Arm circle anti clockwise Data



Result

```
./csv/Arm Circle Anti-Clockwise_Acc.csv
x max
./csv/Arm Circle Anti-Clockwise_Gyro.csv
All : 2.532796106557009
- 8.541532693187014
- 7.916463366582555
- 6.643886680520915
- 7.403411056298441
- 8.749631321219335
PPeaks : 7.850985023561653
Reps Count : 5
```

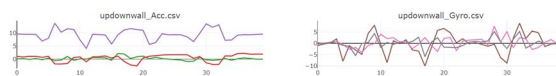
B. One Arm Tricep Extension Data



Result

```
./csv/One Arm Tricep Extension_Acc.csv
All : 0.284782248480091
- 0.3437585555555557
- 18.3025475842944
- 23.683393477700772
- 22.649324208478266
PPeaks : 16.244755956507248
Reps Count : 3
```

C. Up Down Data



Result

```
./csv/updownwall_Acc.csv
All : 1.2178787071175623
- 15.457570962962961
- 19.54907700958526
- 16.064584693648644
- 2.611467204249496
PPeaks : 13.420674967611589
Reps Count : 3
```

V. CONCLUSION

We tested this algorithm with different moves and the algorithm was able to detect the repetitive moves accurately. As this is a light algorithm it does not require a lot of resources and is comparatively faster. This can be used in smart wearables as well. Since the algorithm is not dependent on comparison of two different waves, it does not require any preselection of the activity. This algorithm currently cannot differentiate between two different activities and hence restricts the users (like other algorithms) to manually start/stop recording the activity.

REFERENCES

- [1] F. Ioana-Iuliana, D. Rodica-Elena, "Detection of daily movements from data collected with two tri-axial accelerometers," in Proc.34th International Conference on Telecommunications and Signal Processing (TSP), Hungary 2011
- [2] Li Qiang, J.A. Stankovic, M.A. Hanson, A.T. Barth, J. Lach, Gang Zhou, "Accurate, Fast Fall Detection Using Gyroscopes and Accelerometer-Derived Posture Information,"
- [3] Müller, Meinard "Information Retrieval for Music and Motion", 2007, pp.69-84
- [4] Michal Sustr, "Counting repetitions of exercises from body-worn sensors", Czech Technical University in Prague, Faculty of electrical engineering, Department of cybernetics,
- [5] Koichi Ogawara, Yasufumi Tanabe, Ryo Kurazume, Tsutomu Hasegawa, "Detecting Repeated Motion Patterns via Dynamic Programming using Motion Density", 2009 IEEE International Conference on Robotics and Automation
- [6] Kirell Benzi, Benjamin Ricaud and Pierre Vanderghyest, "Principal Patterns on Graphs: Discovering Coherent Structures in Datasets", Ecole Polytechnique Fédérale de Lausanne, Lausanne, Vaud, Switzerland
- [7] ChunMei Lu, Nicola J. Ferrier, Repetitive Motion Analysis: Segmentation and Event Classification, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 26, NO. 2, FEBRUARY 2004
- [8] Ozlem Durmaz Incel, "Analysis of Movement, Orientation and Rotation-Based Sensing for Phone Placement Recognition", 2015, Department of Computer Engineering, Galatasaray University,, Ortakoy, Istanbul 34349, Turkey
- [9] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten 1, Paul J. M. Havinga, "Fusion of Smartphone Motion Sensors for Physical Activity Recognition", 2014, Department of Computer Engineering, Galatasaray University, Ortakoy, Istanbul 34349, Turkey
- [10] Jan Machek1, Jakub Parak1, Jan Havlik1, "MOVEMENT DETECTION IN THE ACCELEROMETER DATA", Department of Circuit Theory, Faculty of Electrical Engineering, CTU in Prague
- [11] Bobak Mortazavi, Mohammad Pourhomayoun, Gabriel Alsheikh, Nabil Alshurafa, Sunghoon Ivan Lee, Majid Sarrafzadeh, "Determining the Single Best Axis for Exercise Repetition Recognition and Counting with SmartWatches", Wireless Health Institute Computer Science Department, UCLA, Los Angeles, CA 90095
- [12] Sriram Ganapathy, Samuel Thomas, Petr Motlicek2 and Hynek Hermansky, "APPLICATIONS OF SIGNAL ANALYSIS USING AUTOREGRESSIVE MODELS FOR AMPLITUDE MODULATION", 2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)