2015 17th UKSIM-AMSS International Conference on Modelling and Simulation

Maxxyt: An Autonomous Wearable Device for Real-time Tracking of a Wide Range of Exercises

Danish Pruthi, Ayush Jain, KrishnaMurthy Jatavallabhula, Ruppesh Nalwaya
Dept. of Computer Science & Information Systems
Birla Institute of Technology and Science Pilani
Pilani, India
Email: {f2011037, f2011066, f2011706, f2011145}@pilani.bits-pilani.ac.in

Puneet Teja
Research & Design
Gymneus Sports Technology Design
Vienna, Austria
pt@gymneus.com

***Abstract*—Strength training exercises are an important part of fitness training. Although several fitness devices exist to track and analyze cardio-training activities like walking and running, devices to track repetitive movements during strength training exercises are few and they have technical limitations. We introduce Maxxyt - an autonomous wearable device that can count repetitive movements during strength training in real-time. Maxxyt functions without any prior exercise-specific training and can self-adapt to any generic exercise involving repetitive movements. It has been designed with the motive of optimizing the experience of a typical user training in a gym. It uses an Arduino-compatible microcontroller, with primary memory of 4 KB. Maxxyt provides haptic and visual feedback to the user within 2 seconds from starting an exercise. The proposed method to count repetitions involves three phases: *i) pre-processing*, *ii) peak-detection* using *adaptive techniques*, and *iii) aggregation* of results from various axes. We present results on test data collected in a gym consisting of 1171 repetitions from 100 sets of 10 distinct exercises. The repetition count accuracy of the device is 2 repetitions 98% of the time. These results suggest a new paradigm of responsive, wearable strength training fitness devices.**

Keywords–strength training; repetition counting; sensors; wearable computing

I. Introduction

Regular exercise plays a crucial role in improving fitness and reducing obesity [4]. High-intensity strength training exercises are an effective and feasible means to preserve bone density while improving muscle mass, strength, and balance [5].

Automatic tracking can motivate regular fitness activity, in particular pedometry, which has been extensively studied [6]. Setting and tracking goals positively affects motivation and enhances performance [7]. A multitude of wearable fitness trackers exist, they range from pedometers and GPS trackers to sensors that monitor heart rate, calorie consumption etc. However, wearable devices having the capability to address strength training are rare and pose unique technical challenges.

Weight training involves a combination of different exercises on multiple machines with varying weights and repetitions. Several people note their exercises manually on paper or in applications to keep track of their progress, which is onerous and time-consuming and therefore negatively affects motivation. Catering to this need, Maxxyt allows users to focus on the exercise alone, automating the repetition counting and logging progress. The progress would be instantly available for the user over mobile or desktop devices for post workout analysis.

Most fitness trackers use accelerometers to recognize activity and understand context [3, 8, 9, 10]. Maxxyt uses a tri-axial accelerometer that returns a real valued estimate of acceleration along x, y and z-axis and a gyroscope that measures orientation in terms of angular momentum across the three axis. These sensors provide inputs for Maxxyt to capture variety of movement patterns across different exercises.

In this work, we present a novel, responsive and computationally inexpensive way to count repetitions. Since the approach doesn't require any prior training, it is readily scalable and generic. Our approach consists of 3 phases namely: *preprocessing* for data smoothing, *peak detection* for repetition counting along 3 axes for both accelerometer and gyroscope, and *aggregation* of these results to give the final count. The key challenges faced are:

- Making the algorithm generic and therefore applicable for a wide range of strength training exercise. The algorithm needs to be robust to handle different kind of movements involving varying amplitude, frequency of signal and complex peaks per repetition.
- Each user performs a given exercise differently, leading to wide variety not just across exercises but also across users.
- Availability of limited resources in terms of memory and processing capabilities make it challenging to design a system that is responsive and accurate simultaneously.

The rest of the paper is organized as follows. Section II reviews previous work related to repetition counting and exercise tracking. Section III outlines our algorithm. Results of simulated and real tests have been presented in Section

978-1-4799-8713-9/15 $31.00 © 2015 IEEE
DOI 10.1109/UKSim.2015.62

IV. Section V describes the Maxxyt device. Section VI concludes the paper, whilst describing the proposed extensions to the current work.

II. Related Work

The most relevant previous work is that of Chang et al. [2], who address the problem of counting repetitions using two different approaches. First approach utilizes peak detection with an appropriate order of low pass filter. In techniques based on peak detection, complex peaks with more than two peaks in a single repetition disrupt the net count and it suffers from the problem of double counting. Treating the signal with appropriate order of low pass filter removes the complex repetitions. Their second approach models the problem as an HMM (Hidden Markov Model) to extract fixed set of shapes and then counting them. The first approach gives low miss count rate of 5%, however finding a correct order of low pass filter is tricky, an order that works well for an exercise doesn't necessarily suit another exercise. There is no notion of appropriate appropriate global order, which limits the approach to only a limited set of exercises. The second approach involves training a counting version of HMM for each exercise, making it non scalable and only manages a miss count rate of 10%. Maxxyt uses an approach similar to their first approach where the problem of double counting in complex repetitions is partly solved by blur filter and partly by the aggregation module. Maxxyt has a low miss count rate of 4.3% and is completely independent of the exercise analysed, thus making it highly scalable and generic.

In another work by Morris et al.[1], the result of counting repetitions relies on accurate segmentation and recognition of exercise. Their technique rejects peaks based on the minimum and maximum reasonable times needed for one repetition for the exercise recognized. We believe that repetition counting should be independent of exercise recognition for it to be a scalable system. Reliance on recognition makes the system less responsive and deteriorates the user experience. We achieve counts that are accurate to ± 1 repetition 95% of time and ± 2 repetition 98% of the time compared to their within ± 1 accuracy of 93% and within ± 2 accuracy of 97%.

An indirect comparison can also be made with the algorithms used for movement detection (i.e. starting and ending of any movement). This is because repetition counting involves detecting the start and end of any repetitive action (step during a walk or any movement in general). Specifically, work done by Jan Machek et al. [3] provide two unique algorithms for movement detection by locating the changes in the starting and ending of a movement from the accelerometer data. They use second differentiation, points of inflection, cumulative sum and peak detection in their algorithms. However, they limit their model by using experimental thresholds for the size of left and right neighborhoods of point of inflection as well as multiplication constant. This makes them less robust and involves a large working window size making the model unsuitable for real time scenarios.

III. The Algorithm

The repetition algorithm is divided into three main phases viz *Pre-Processing*, *Peak detection*, *Aggregation*. The data gathered from the sensors is divided into windows of small size each consisting of only 25 data points. All the three phases are applied to each window. There is 10% overlap of data points from the previous window to maintain a smooth transition. The number of peaks detected in each window is added to the results of the previous window [along each axis] and aggregation is applied on these incremental results. The repetitions are computed along 6 axes: x, y and z axis for both accelerometer and gyroscope.

Our technique allows the device to function with high accuracy even with window size as small as 25 data-points which are accumulated in about 2 seconds. Such small window size enables the device to be very responsive and give continuous feedback. A detailed description of our algorithm is given below.

A. Preprocessing

It is necessary to smoothen the data gathered from accelerometer and gyroscope sensors to remove spurious peaks that may interfere in the peak detection phase. Hence, it leads to the elimination of false peaks that do not correspond to an exercise repetition.

1) Comparison of Smoothing Techniques: A number of data smoothing techniques are readily available for noise reduction/removal. We experimented with several techniques namely blur filters (i.e. moving average filters), simple integration, Runge-Kutta integration [11], and median filters. Blur filters reduced abrupt noise and brought out an observable difference between the peak and non-peak values. Median filters eliminated abrupt noise in the signal, however blur filters performed better. Runge-Kutta integration eliminated noise better than any of the other techniques, but almost flattened out the entire signal and the calculations involved large numbers that are computationally intensive. Blur filter is employed in our application as it clearly outperforms other techniques for getting a smooth signal as seen in Fig. 1.

B. Peak Detection

Smoothing of data is followed by detection of peaks corresponding to each repetition in an exercise.The ideal peak detection algorithm should work with varying amplitudes and frequency.We developed a novel technique where a global maxima[or minima] is declared a maximum[minimum] peak if it is δ larger[smaller] than the neighborhood values. We declare the maximum[minimum]

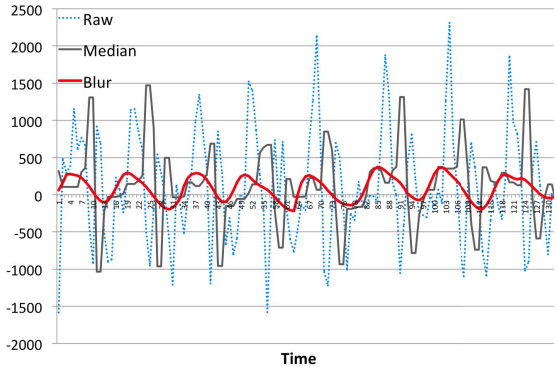

Figure 1. Smoothing Comparison Results

TABLE I
OVERVIEW OF EXERCISES AND RESULTS.

Exercise Sets	No. of Total Sets	Total RepCount	Total Miscount
Cable Row	15	154	13
Inverted fly	15	152	6
Hyperextension	24	336	9
Leg raise	17	274	8
Pulldown	17	176	8
Fly	4	26	2
Bench Press	4	27	2
Bent over row	4	26	0
Total	**100**	**1171**	**48**

value till now as a peak, if there exists some data point which is δ less [more] than it. Once a maximum[minimum] is declared, the maximum[minimum] till now is refreshed to a very small[large] value.

The δ is appropriately learned with time and hence adapts to the varying amplitude across signals. The learning process is governed by the following equation.

$$\delta_{new} = \alpha(prevMax - prevMin) + \beta\delta_{old} \qquad (1)$$

In the above equation, α and β are empirically tuned constants, and $prevmax$ and $prevmin$ are the previous maximum and minimum respectively.

All the smoothing and peak-detection operations across windows with little overlap between consecutive windows. A simulation of our algorithm with varying windows sizes helped us decide an ideal window size of 25 data points which is equivalent of 2-3 seconds.

C. Aggregation and Combining Rep Counts

The peak counting algorithm is applied separately to the 3 gyroscope axes and 3 accelerometer axes generating a total of 6 repetition counts. These repetition counts, are combined along with empirically determined rules rules in a novel way to get the final repetition count. These rules are based on the following observations:

- In most instances our peak detection algorithm [for each individual axis] counts exact number of repetition or more.
- It often counts additional peaks due to noise in the signals at the beginning and end of the exercise for some axes.
- Sometimes, when an axis has multiple peaks for a single cycle of exercise the peak detection algorithm gives the rep-count which is near integer multiple of actual count.
- In some cases, repetition counts from 2 or more axes coincide with each other and also with the actual count.
- For a given exercise, either the gyroscope signals and the corresponding rep-counts are good enough for final result or the accelerometer signals and its corresponding rep- counts.

We build upon these observations to develop a simple yet novel technique, constructing an if-else ladder. It utilises (i) consensus among the rep-count values from different axes, (ii) standard deviation in the rep-count values from gyroscope (iii) standard deviation in the rep-count values from accelerometer (iv) ratio of rep-count values to determine near integer multiple to detect multiple peaks in single cycle of exercise and (v) mode (smallest if not unique) of the rep-counts to map the 6 rep-count values to a single value and feedback it to the user.

IV. RESULTS

Our experiments included 100 sets of exercises performed by multiple users comprising of 8 distinct type of exercises. These exercises were performed in a typical gym. An brief overview of all the exercises and their corresponding results can be seen in Table I.

In order to select an ideal window size for the 3 phases mentioned in the algorithm section, we experimented with window size varying from 5 to 100 and calculated the corresponding accuracy. This experiment was really crucial as window size determines the device latency to compute the result. There were two types of accuracy that were calculated, one within an error of ± 2 repetition and other within error of ± 1 repetition . This means that if the rep-counts calculated for a particular exercise set from the device is within an error of ± 1 [or ± 2], it is considered as accurate. We chose a window size of 25 as it clearly gives the highest accuracy in comparison to others [Fig 2]. This was a major factor in increasing the responsiveness of Maxxyt.

We also compared our results against the results of [1] and [2] in terms of accuracy, which are tabulated in Table II and in Table III. Since the number of sets of exercises, the type and the number of repetitions were very different in both these papers, we collected data over a large number of exercise sets. The ground truth and the repetition count

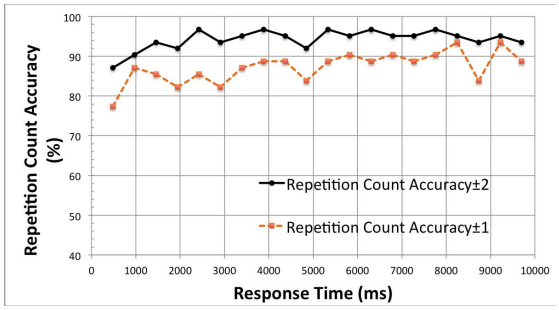

Figure 2. Simulated plot of accuracy [within ± 2 and ± 1] vs the response time (which directly depends upon the working window size)

TABLE II
AVERAGE MISCOUNT RATE IS RATIO OF ABSOLUTE ERROR TO ACTUAL REP-COUNT

Approach Used	*Average Miscount Rate*
Chang et al. (without HMM)	0.106
Chang et al. (with HMM)	0.057
Maxxyt	**0.043**

TABLE III
FRACTION OF SETS (100 SETS IN OUR STUDY AND 160 SETS IN [1]) THAT ARE COUNTED EXACTLY ± 1 AND ± 2.

Approach Used	*Exact*	*Within 1*	*Within 2*
Reco fit[1] (Actual boundaries)	0.5	0.97	0.99
Reco fit[1] (Segmenter)	0.77	0.93	0.97
Reco fit[1] (Training Data)	0.7	0.93	0.97
Maxxyt	*0.61*	*0.95*	*0.98*

was collected by the device for 100 sets of exercises, distributed over different type of exercises viz. Seated Cable Row, Inverted fly, Fly, Hyperextension, Leg raise, Pulldown, Bench Press and Bent over row. Each set of of every exercise had number of repetitions varying from 5 to 35. The accuracy results are listed in Table I. We compare the average miss count rate of [2] with our work. [2] lists the miss-count rate for each exercise separately using different order of low pass filter. We take the average over all of them and use it for comparison. In a different approach, [2] uses a counting version of Hidden Markov Model. It can be seen that we completely outperform [2] in both cases. We perform 59% better in terms of average miss count rate when compared to peak detection in [2] and 24% better than peak detection with HMM[2]. We compare the accuracy of [1] with our work in terms of exact count ± 1 and ± 2. [1] have calculated three different accuracies corresponding to the actual boundaries (in which the human oracle marks the beginning and end of the exercise in the test data), boundaries from Segmenter (which uses trained Segmenter to mark the beginning and end of the exercise in the test data), and training data (which corresponds to the training error). We get comparable results to [1] when compared using Actual Boundaries, as in Table III. Since the user marks the beginning and end of the exercise by the press of a button on Maxxyt, comparison with accuracy of Actual Boundaries in [1] is the most representative.

V. THE DEVICE

Maxxyt has the form factor of a large wristwatch, and is worn on the hand while performing an exercise. Its design can broadly be organized into two subsystems - the I/O subsystem and the controller subsystem. The former takes care of measuring the motion and orientation (pose)of the device in three-dimensional space and displaying output, while the latter implements algorithms for counting repetitions, managing I/O, and related tasks. The device was powered by a low-power Lithium-ion battery.

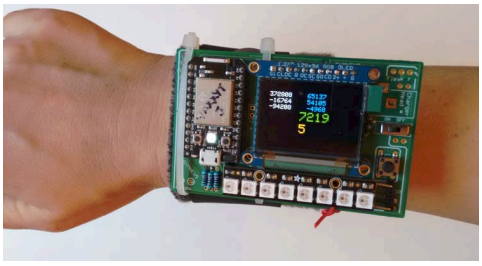

Figure 3. Maxxyt Prototype Device

A. The I/O Subsystem

In order to track the users orientation and motion, MPU-6050, a 3-axis accelerometer and a 3-axis gyroscope is used. It transmits to the controller the acceleration and angular velocities of the user at a fixed sampling rate. The sampling rate is determined experimentally and is implemented by sending an interrupt signal on the I2C bus connecting the controller and the MPU-6050. It is ensured that this signal is sent at uniform intervals by using a timer. Apart from the MPU-6050, this subsystem consists of an OLED display in order for the user to view output, a vibra motor which buzzes once for every two repetitions, an on/off switch, and another microswitch that is used to start and stop repetition counting. It also has an SD card for storing exercise data and the intermediate results that the algorithm computes, which is essential for enhancing the performance of the algorithm.

B. The Controller Subsystem

The controller used is an Arduino compatible microcontroller, which is a tiny, WiFi capable ARM microcontroller, with a main memory of 4 KB, and a flash memory that can hold programs of size upto 32KB. It can connect to a standard IEEE 802.11 b/g/n wireless access point, and serves two main purposes. First, it serves as the controller that takes in sensor readings, processes them, and displays results to the user, whilst maintaining timers used for sampling and synchronization of I/O. Second, at the end of an exercise, relevant data is stored on the SD card and transmitted to a server where the output can be analyzed better.

C. User Experience

Since Maxxyt is wearable and small in size, it gives optimal experience to user by giving continuous feedback to the user even when he is performing the exercise. It has no constraints in terms of device orientation and external device requirement, and is completely autonomous. Such features in our device are due to simple yet unique calculations in our algorithm that takes minimum amount of memory, storage space and computational power to calculate the final result. Moreover, Maxxyt provides continuous haptic and visual feedback to the user for motivation before and after completion of exercise. All the exercise data and results can be transmitted in real time to the cloud for post workout analysis as well.

VI. Conclusion

From the different experiments conducted and their results shown above, we can conclude that the device is responsive with a latency of only 2-3 seconds and generic as it handles variety of exercises without any prior training. It achieves higher level of accuracy in comparison to existing techniques for counting repetitions and tracking exercise. Moreover, the wearable device gives continuous haptic and visual feedback to the user for enhanced user experience.

References

[1] Morris, Dan, T. Scott Saponas, Andrew Guillory, and Ilya Kelner. *RecoFit: using a wearable sensor to find, recognize, and count repetitive exercises.* In Proceedings of the 32nd annual ACM conference on Human factors in computing systems, pp. 3225-3234. ACM, 2014.

[2] Keng-hao Chang, Mike Y. Chen, and John Canny. *Tracking Free-Weight Exercises.* Ubiquitous Computing 9th International UbiComp 2007, Innsbruck, Austria, September 16-19, 2007.

[3] Jan Machek, Jakub Parak1, and Jan Havlik. *Movement Detection in the Accelerometer Data*. Deparment of Circuit Theory, Faculty of Electrical Engineering, CTU in Prague.

[4] Janssen, I., LeBlanc, A.G. *Systematic review of the health benefits of physical activity and fitness in schoolaged children and youth.* Intl J Behavioral Nutrition and Physical Activity 7.40: 1-16, 2010.

[5] Miriam E. Nelson, Maria A. Fiatarone, Christina M. Morganti, Isaiah Trice, Robert A. Greenberg, and William J. Evans. *Effects of High-Intensity Strength Training on Multiple Risk Factors for Osteoporotic Fractures: A Randomized Controlled Trial* In Journal of American Medical Association 1994 Dec 28;272(24):1909-14.

[6] Chan, Catherine B., Daniel AJ Ryan, and Catrine Tudor-Locke. *Health benefits of a pedometer-based physical activity intervention in sedentary workers.* Preventive medicine 39, no. 6 (2004): 1215-1222.

[7] Steele-Johnson, Debra, Russell S. Beauregard, Paul B. Hoover, and Aaron M. Schmidt. *Goal orientation and task demand effects on motivation, affect, and performance.* Journal of Applied psychology 85, no. 5 (2000): 724.

[8] Ravi, Nishkam, Nikhil Dandekar, Preetham Mysore, and Michael L. Littman. *Activity recognition from accelerometer data.* In AAAI, vol. 5, pp. 1541-1546. 2005.

[9] Casale, Pierluigi, Oriol Pujol, and Petia Radeva. *Human activity recognition from accelerometer data using a wearable device.* Pattern Recognition and Image Analysis. Springer Berlin Heidelberg, 2011. 289-296.

[10] Mathie, M. J., Branko G. Celler, Nigel H. Lovell, and A. C. F. Coster. *Classification of basic daily movements using a triaxial accelerometer.* Medical and Biological Engineering and Computing 42, no. 5 (2004): 679-687.

[11] Schwartz, A., and E. Polak. *Consistent approximations for optimal control problems based on Runge-Kutta integration.* SIAM Journal on Control and Optimization 34.4 (1996): 1235-1269.