Activity Monitoring Using Smart Glasses: Exploring the Feasibility of Pedometry on Head Mounted Displays

Zhiquan You[1], Farnaz Mohammadi[2], Emily Pascua[1], Daniel Kale[1], Abraham Vega[1], Gian Tolentino[1], Pedro Angeles[1], and Navid Amini[1](✉)

[1] California State University, Los Angeles, Los Angeles, CA 90032, USA
namini@calstatela.com
[2] University of California, Los Angeles, Los Angeles, CA 90095, USA

Abstract. Fitness tracking, fall detection, indoor navigation, and visual aid applications for smart glasses are rapidly emerging. The performance of these applications heavily relies on the accuracy of step detection, which has rarely been studied for smart glasses. In this paper, we develop an accelerometer-based algorithm for step calculation on smart glasses. Designed based on a salience-analysis approach, the algorithm provides a highly accurate step calculation. An activity monitoring application for a commercial Android-based smart glasses (Vuzix M100) is designed and realized for algorithm evaluation. Experimental results from 10 participants wearing the smart glasses running our application achieved average step detection error of 2.6% demonstrating the feasibility of our salience-based algorithm for performing pedometry on smart glasses.

Keywords: Smart glasses · Accelerometer · Activity monitoring · Salience · Peak-to-Peak

1 Introduction

In recent years, interest in smart glasses has been substantially growing, as has the number of influential companies, such as Amazon and Facebook, that are announcing their entry into the smart glasses market. It is expected that the combined market size for smart glasses will grow at an exponential rate of 76% annually and will reach $16B by 2025 [1]. The use of commercial smart glasses for fitness tracking [2, 3], indoor navigation [4, 5], fall detection [6, 7], vision enhancement in the visually impaired [8–10] has already gained momentum and it is expected that such applications will comprise a considerable sector of the smart glasses market. The performance of these applications as well as many other emerging services is dependent on the accuracy of pedometry (step detection and counting) on smart glasses.

Z. You and F. Mohammadi – Contributed equally.

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2020
Published by Springer Nature Switzerland AG 2020. All Rights Reserved
M. M. Alam et al. (Eds.): BODYNETS 2020, LNICST 330, pp. 153–167, 2020.
https://doi.org/10.1007/978-3-030-64991-3_11

Due to the proliferation of accelerometers in consumer products and by virtue of their low power consumption, accelerometer-based step calculation has emerged as the most popular approach to perform pedometry. Many existing step counting solutions have shown that when a user is walking, the measurements of the accelerometer (on the user's body) will regularly change and such changes can be used to calculate the number of steps taken by the user.

Plenty of research studies on activity monitoring, especially step detection have been conducted [11]. A number of such studies aim at detecting steps for handheld devices such as smartphones [12]. Other researchers have transformed the problem of step detection into a case in pattern recognition and employed a machine learning technique for pedometry [13, 14]. Another group of studies devised algorithms that target step detection in ankle-worn, shin-worn, or waist-worn devices [15, 16]. To the best of our knowledge, step detection techniques suited for smart glasses have not been adequately explored. Smart glasses cannot offer similar activity tracking utility levels compared to the typical handheld, waist-worn, or ankle-worn wearable devices [17]. Head-worn devices typically contain a limited number of sensors and they are vulnerable to external sources of error. Moreover, it is typical for smart glasses to require low computing power by virtue of their size, heat, and battery issues [18, 19].

In order to motivate the problem, we conducted an experiment, where we attached 6 inertial measurement units (IMUs) produced by Shimmer Research to different regions on a subject body and asked him to take 220 steps. We use the windowed peak detection technique, proposed in [20] as an optimal step detection method for wearable low-power devices, to derive the number of steps for each IMU. Table 1 summarizes the number of steps counted by each IMU; we observe that lower parts of the body and especially, the shin, thigh, and waist areas performed well; this can be attributed to the higher impact of each step (foot striking the ground) on those areas comparing to other locations.

Table 1. Counted steps vs. the location of IMU

Sensor location	Shin	Thigh	Waist	Forearm	Upper arm	Head
Steps counted	228	199	230	188	179	114
Error($\frac{real-est.}{real}$)(%)	4%	10%	5%	15%	19%	48%

Based on the above experiment, in this paper, we propose a step detection algorithm for smart glasses, which can identify steps accurately and in real-time. The algorithm is composed of a *signal preprocessing* phase, *axes combination* phase, *salience calculation and analysis* phase, and *a peak detection* phase. The proposed algorithm utilizes only the accelerometer data for step calculation since smart glasses are commonly designed with an accelerometer, while other sensors such as the GPS and gyroscope are not always available. Furthermore, the proposed algorithm requires a low level of computing power from smart glasses.

The contributions of this paper are threefold. First, we propose an algorithm for step detection and calculation on smart glasses based on accelerometer data. Second,

in two experiments, we investigate the step detection accuracy of our algorithm on commercial smart glasses as well as on IMUs attached to multiple on-body locations. Third, we incorporated the proposed algorithm into an activity monitoring application for Android-based smart glasses for real-time step calculation. The rest of this paper is organized as follows. Section 2 introduces some related studies. Sections 3 and 4 describe the designed data collection and data analysis phases, respectively. Section 5 covers our experimental results and Sect. 6 discusses constraints and future directions of our investigation. Finally, Sect. 7 concludes this study.

2 Related Work

There are numerous algorithms and research studies conducted on step detection based on smartphones and waist-mounted devices, with high accuracies [20]. Some algorithms explore other ways to detect steps beyond the basic peak-valley relationship, such as the method proposed by Kumar et al., [21] discussing the use of the linear relationship between the amplitude of acceleration and the frequency of steps, effectively detecting steps through their pilot trials. Most of the research in the field of step detection focuses on 5 body parts to mount the censor: wrist, pocket (waist level), thigh, ankle, and foot. The most reliable step count algorithms come from insole pressure sensors that have the ability to detect the pressure applied to the sensor once a subject takes a step [22, 23]. The impulse (force) generated from each foot striking the ground in walking is more pronounced in those areas. In general, lower parts of the body such as the ankle and foot absorb most of the impact of stepping and therefore, devices placed on these body parts leads to the most accurate results [24].

Smartwatches and smartphones are among the electronic equipment that is capable of recording and analyzing motion signals [25–27]. However, smartphones are not necessarily always taken in the same or relative location. In addition, these devices are extremely sensitive to non-ambulatory activities [28]. Moreover, hand and arm movements when the device is being carried, do not necessarily correlate with stepping and may create artifacts on the signal. On the other hand, smartwatches provide a tremendous amount of information by continuously monitoring of the user and measurement of their physiological parameters. Although, the data acquired from smartwatches are under the question of reliability, due to subtle and often wrist movements, compared with other parts of the body where the sensors are usually placed [26]. One inevitable shortcoming of data based on wearable devices is that they are not generalizable and unstable across different brands [29], which is the case in smartphones as well.

The most popular step detection algorithm uses peak-valley extraction to detect steps from the accelerometer data. The three-axis accelerometer data is combined into a single acceleration vector, which is then graphed where each peak is considered to be a step [30, 31]. A more advanced algorithm would be a filtering system which places thresholds for each peak to be considered a step, as there will be multiple peaks from the accelerometer that is not considerable enough to be detected as a peak.

Most step counting algorithms are created for devices inside the pocket [32]. For instance, Brajdic and Harle [20] provided evaluations considering various algorithms for step detection on smartphones. They evaluated several algorithms such as windowed

peak detection (WPD), hidden Markov model, and continuous wavelet transform, which were the most promising approaches. They found that the simplest algorithm with the best accuracy is the windowed peak detection algorithm.

A number of studies have employed a modified version of the peak-to-peak algorithm to detect steps from an accelerometer signal. For instance, the peak detection algorithm proposed in [33] produces a refined method that enables the algorithm to detect peaks in periodic and quasi-periodic signals. This method is noted to have high efficiency in peak detection within high and low-frequency signals. Another innovation based on step detection for head-mounted sensors is being used for detecting user movements when using virtual reality systems. The idea of these virtual reality systems is to immerse the user into a virtual world in which their movement in real life is synchronized to their virtual avatar [34].

3 Sensor Platforms and Pilot Experiments

We run a pilot study to investigate the step detection accuracy of our algorithm on commercial smart glasses as well as on the IMUs attached to multiple body parts. The first experiment is through activity monitoring via a smart glass device for the head signal, and another is using the IMUs for pocket, head, and foot-worn IMUs. Experiments are conducted in an outdoor environment at the California State University, Los Angeles campus. The participants include 10 students (average age 26.3, SD = 5.2, average height 170 cm, SD = 11.3 cm). For both experiments, each test consisted of walking 100 steps at their usual pace repeated 10 times per individual. For this pilot study, none of the participants reported any limitations in their mobility or vision. In the following, we explain both experiments.

Fig. 1. Vuzix M100 device on test subject for data collection.

3.1 Step Detection on the Smart Glass

A popular Android-based smart glass in the research and industry communities, namely the Vuzix M100, with a sampling frequency of 100 Hz is used in the first experiment. Figure 1 shows a subject wearing the Vuzix M100 with the activity monitoring Android application running. The application utilizes our salience-based algorithm to detect the steps in real time. Figure 2 is a screenshot of the Android application.

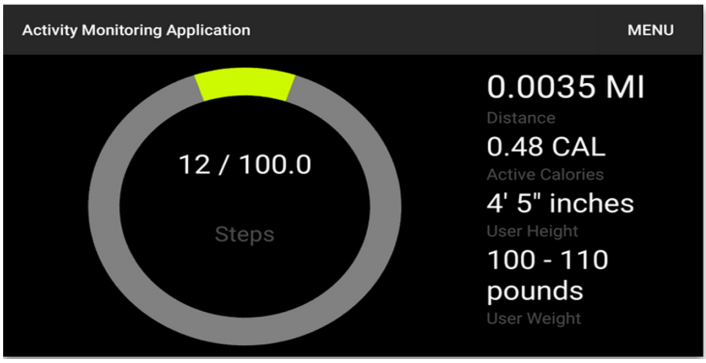

Fig. 2. Graphical user interface of application on Vuzix M100.

3.2 Step Detection on IMUs

In experiment 2, we employ the Shimmer3 [35] IMU unit to collect the acceleration signal from the head, foot, and pocket locations. The step detection is performed off-line on a desktop computer. Figure 3 shows the Shimmer worn by the same subject on the head.

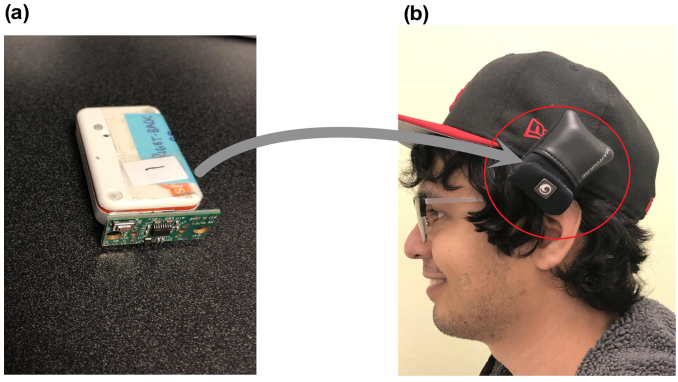

Fig. 3. (a) The Shimmer3 IMU, (b) the Shimmer sensor is worn by a subject on the head.

4 Methods

We propose a novel algorithm to identify the steps using the accelerometer signal from a smart glass device and compare the accuracy of our results against the WPD method. The WPD uses a moving average to smooth the accelerometer signal, and it detects single peaks via a fixed size sliding window; it finds the maximum value in the window, shifts the window, and discards the chosen maximum if it persists for two windows in a row. This algorithm has been identified as an optimal step detection approach for wearable devices owing to its computationally inexpensive nature [20, 36]. Figure 4 presents the block diagram of our step detection approach; the description of each step will follow in the preceding subsections.

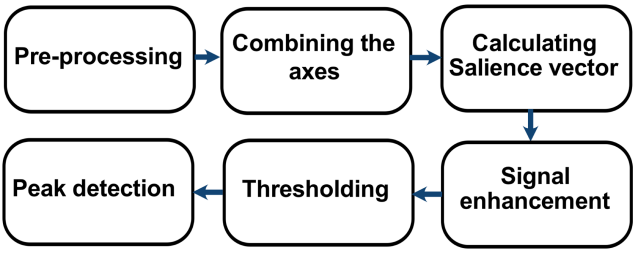

Fig. 4. The block diagram representing the flow of the approach.

4.1 Pre-processing

The signal preprocessing consists of calibration and low-pass filtering. First, the calibration process is utilized to reduce the drift errors and offsets from the raw acceleration signals. Second, a moving average filter is used to suppress the high-frequency noise of the calibrated inertial signals. Details of the preprocessing phase can be found in [29].

4.2 Combining the Axes

The three axes (x, y, z) are combined to a single magnitude measure. We use (1) to calculate the magnitude of combined acceleration in all directions using the Euclidean distance method:

$$r = \sqrt{A_x^2 + A_y^2 + A_z^2} \tag{1}$$

4.3 Peak Detection with Salience Algorithm

We calculate the salience of each acceleration sample point. Considering all possible intervals in a given signal of size N, salience is the length of the longest interval for which a sample is a maximum [37, 38]. The key property of the salience values algorithm is

that the starting point of each step has a large salience value. Hence, steps can be found by locating these prominent points.

As an example, salience can be visualized as shown in Fig. 5. Considering the fourth sample in the signal shown in Fig. 5; despite the fact that it has a higher magnitude than the seventh sample, it has a smaller salience comparing to the seventh sample, $s(4) = 4$ as opposed to $s(7) = 8$. The term salience vector represents the resulting signal containing the salience of each sample in the original signal. The list of salience vector of the example signal shown in Fig. 5 can be found in (2). We use salience signals to find each stride that has the largest salience samples over a time threshold of 30 s. Once the time threshold is met, the salience vector is then processed through a peak detection algorithm to determine the steps of the filtered data.

$$s(k) = \{14, 1, 2, 4, 1, 13, 8, 1, 4, 2, 1, 5, 2, 1, 15\}, \quad 1 \leq k \leq 15 \tag{2}$$

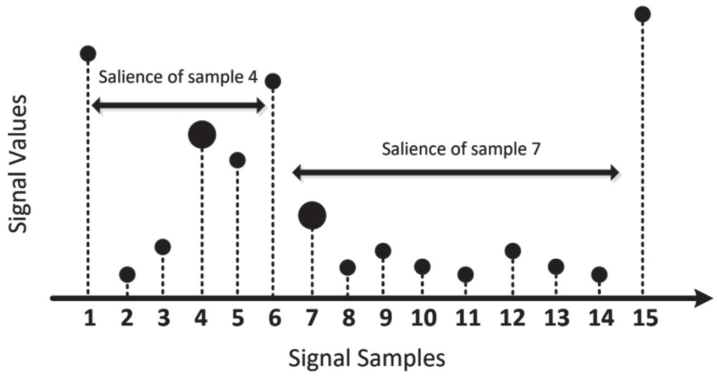

Fig. 5. Salience of an example data sequence.

We have implemented all three algorithms found in [38] for computing the salience vector, *s*, of a signal. These algorithms include: 1) basic salience computation, 2) partial salience allocation, and 3) sliding window analysis. We found the sliding window approach to be the most suitable one for the purpose of step detection, as it offers faster execution time and it addresses the issues pertaining to the signal's origin. The following briefly explains three algorithms with which we can calculate salience that is done after combining the three axes of the signal and obtaining the *r* vector [37].

Basic Scheme. To find the length of the longest interval for which a sample is maximum in a signal, the following steps are required:

1. Initialization: Since each sample is a maximum with respect to itself, all salience values are set equal to 1. Thus, the length of the analysis interval (n) is set as $n = 2$.
2. Subdividing the input signal of length N into analysis intervals of length n. The length of the last interval will be between 1 and n.
3. Finding the maximum point within each interval and assigning a salience of n to the corresponding sample.

4. Increasing the size of the analysis interval by one and going back to step 2 until n reaches N.

The last calculated salience vector will be reported.

Partial Salience Allocation Scheme. As an alternative to the basic scheme, partial salience allocation is used in the case of computationally expensive functions. This algorithm utilizes tabular representation of the samples as instead of finding maxima over intervals with increasing lengths, one can determine the maximum over frames of decreasing lengths. With j being the position of the global maxima in the signal, the value in this position is a maximum for the previous $j-1$ first samples. Accordingly, assuming that j is known, one can compute the maximum over $j-1$ first samples only and assign it a salience of $j-1$ and so on, until a salience has been assigned to the first sample.

Sliding Window Salience Computation Scheme. Partial salience allocation is faster than the basic scheme, however, this method cannot find salience values for all the samples in the input signal. Furthermore, the problems related to the samples near the boundaries are not addressed in this scheme. The intuition behind the sliding window scheme is to apply a window to the input signal and move the window sample by sample in order to address the boundary problem. In this method, a sliding window $w_M(i)$, of length M, where $M < N$, and origin i, is placed at the beginning of the signal and is shifted sample by sample towards the endpoint of the signal. Saliences $S(k,i)$, $S_{left}(k,i)$, $S_{right}(k,i)$ are then computed with respect to the sliding window. If the proper window size is chosen, the running saliences obtained will suffice for most applications. We define local saliences with reference to the position of a given window. Moreover, running saliences $S^*(k)$, $S^*_{left}(k)$, $S^*_{right}(k)$, are defined as the maximum of the local saliences of sample k for all window positions (previous and current). Thus, running saliences are non-decreasing. Here we outline the sliding window analysis scheme:

1. Initialization: all saliences S^* are set equal to 1 and all saliences S^*_{left} and S^*_{right} are set equal to zero.
2. Applying the partial salience allocation scheme to the samples within the sliding window with origin i and computing the local $S(k,\ i)$, $S_{left}\ (k,\ i)$, and $S_{right}\ (k,\ i)$ saliences, where $k = i, i+1, \ldots, i+M-1$.
3. Updating running saliences as follows: for each sample k, $S^* = S(k,\ i)$, $S^*_{left}(k,\ i) = S_{left}(k,\ i)$, $S^*_{right}(k) = S_{right}(k,\ i)$ if their values will be increased, otherwise, we skip this step.
4. Shifting the sliding window to the right by one sample, i.e., $(i \rightarrow i+1)$ and going back to step 2.

Once the sliding window reaches the right boundary of the input signal, the final salience vector, $S_{final}\ (k)$ is given by (3):

$$S_{final}(k) = S^*_{left}(k) + S^*_{right}(k) + 1, \quad k = 1, 2, \ldots, M \tag{3}$$

We discard the first and last $M - 1$ element of the salience vector since these values are in the boundary. Due to this problem, it is important to properly choose the size of the sliding window.

4.4 Signal Enhancement

In this step, we compute a vector u, which is defined as:

$$u = \frac{(r.s)}{\max(s)} \tag{4}$$

where s is the salience vector and dot (.) represents an element-wise multiplication. The idea behind deriving u is to make peaks of r more pronounced and to diminish the rest of the samples.

4.5 Thresholding and Peak Detection

We extract all the elements of u that exceed a certain threshold, that is given by (5) as potential cycle indices [38].

$$threshold = \frac{2}{3}\max(u) \tag{5}$$

Then the difference vector, d, between adjacent extracted indices is calculated. We normalize d around its mean and extract one of the two indices of points that fall within this range, i.e., $\mid d - \text{mean}(d) \mid < \text{mean}(d)$. This elimination phase helps to increase the accuracy of our stride detection algorithm by counting two close peaks only once. The number of such points is indicative of the number of steps taken and in fact, these points are the start/endpoints of each step. Finally, we return a list of indices that these points correspond to.

Among the three algorithms to calculate the salience vector, the sliding window scheme is more efficient than the basic and partial salience allocation. Hence, we will base our results on the calculations from the sliding window algorithm. We choose the window size of 94 [38] and calculate the salience vector.

Figure 6a shows the combined accelerometer signal, r, and the red lines represent the occurrence of a step. Figure 6b exhibits the salience vector of the combined acceleration signal. Figure 6c depicts the enhanced salience vector where the peaks (associated with steps) become pronounced, and that this subfigure corresponds to the vector, u, where we enhance the signal. As shown, the peaks are more significant and distinguishable from the rest of the signal.

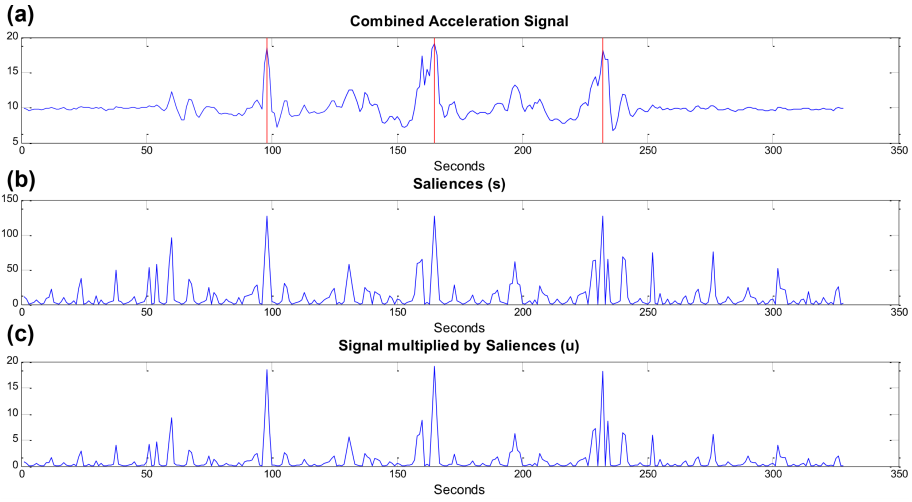

Fig. 6. Original signal (a), the signal's salience vector (b), and the salience vector after normalization (c).

4.6 Error Determination

To quantify the differences between the signals from different parts in the body namely foot, head, and pocket, we calculate the error for each trial and each person as written in (6) and (7). The error in our experiments is defined as the absolute difference of the number of steps calculated by the algorithm from the ground truth [36]:

$$e_j(k) = \frac{\left|steps_j(k) - \widehat{steps_j}(k)\right|}{steps_j(k)} \tag{6}$$

with j being the subject number, and k being the trial number. The term $steps_j(k)$ is the true number of steps that subject j took in trial k, and $\widehat{steps_j}(k)$ is the step calculated by the algorithm. Each subject performs each trial 10 times, thus, the averaged error for subject j and overall repeats (e_j) is given by (7):

$$e_j = \frac{\sum_{k=1}^{n} e_j(k)}{n} \tag{7}$$

5 Results

We evaluate the step detection performance of our algorithm on commercial smart glasses as well as IMUs attached to the head and other parts of the body. In the pilot trial, we collected gait data from 10 subjects and applied our proposed algorithm to these signals. In this section, we compare the performance of the algorithm for the foot, pocket, and head signals, and provide error analysis for the results.

5.1 Accuracy Comparison

Figure 7 graphs the changes in step detection error from the WPD method to our salience-based approach. The average step detection error of our approach (2.6% with SD of 2.8) was significantly lower than that of the WPD method (21.8% with SD of 9.7).

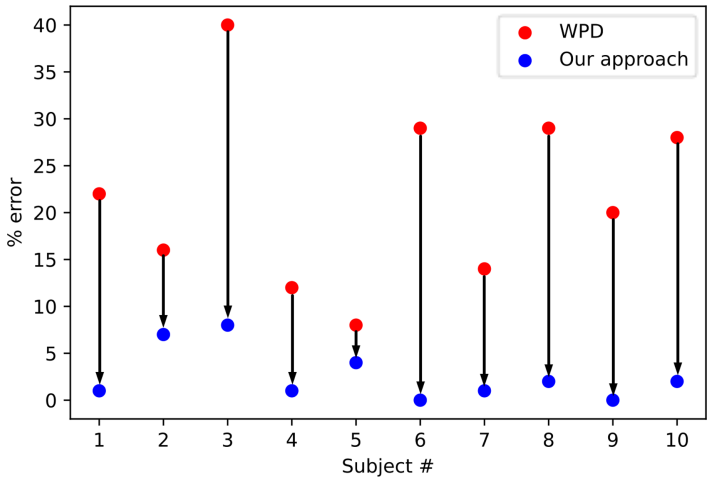

Fig. 7. Comparing the error for step counting from the head signal between the WPD method and our approach.

5.2 Error Analysis

Table 2 lists the average step detection for three IMU locations in experiment 2. For both the WPD method and our approach, the foot location performs remarkably well and achieves similar results with an average error of 2.9% (SD = 1.6) and 2.7% (SD = 2.4), respectively.

We note moderate improvements in step detection for the pocket location with our approach (6% with SD of 3.5) comparing to the WPD (7.2% with SD of 6.5). The head location benefitted the most from our salience-based approach, where the step detection error declined from 25.6% (SD = 9.4) to 3.9% (SD = 2.4). The error range for the head location was from 13% to 39% in the WPD method; however, with our approach, the range was narrowed down to 0% to 8%.

Table 2. Average step detection error in experiment 2.

Subject	WPD			Our approach		
	Head	Pocket	Foot	Head	Pocket	Foot
1	31%	10%	1%	8%	1%	0%
2	23%	5%	5%	3%	13%	2%
3	35%	2%	2%	5%	2%	2%
4	21%	6%	0%	4%	8%	6%
5	15%	1%	2%	0%	6%	7%
6	38%	9%	4%	5%	4%	4%
7	20%	23%	5%	2%	7%	1%
8	13%	10%	3%	3%	5%	0%
9	21%	1%	3%	2%	9%	4%
10	39%	5%	4%	7%	5%	1%
Average	**25.6%**	**7.2%**	**2.9%**	**3.9%**	**6.0%**	**2.7%**
St. Dev.	**9.4**	**6.5**	**1.6**	**2.4**	**3.5**	**2.4**

6 Discussion and Future Work

During the process of this research, we are attempting to develop an algorithm that further improves on the current step detection algorithms by collecting additional data for testing and applying the salience algorithm to filter the noisy data received from accelerometer sensors. We focused on analyzing smart glass data received from the sensors. There are several factors that make it difficult to achieve the same accuracy from head data than it is from the foot data. One of the most challenging issues with collecting data from the head is all of the movement the head experiences throughout walking. Every slight acceleration from the head can be falsely interpreted as a step, which is where the salience algorithm aids in fixing. Moreover, in our trials, the subjects are asked to walk at their normal pace that is relatively slow which is more challenging. In the future, we intend to investigate the performance of our algorithm for various speeds. As people age, the angular displacement of the head changes and the body will have a decreased ability to attenuate accelerations from trunk to head [39]. As another future direction, we will investigate our algorithm's performance on older adults.

Providing an improvement in accuracy on current step detection algorithms can also be applied on a myriad of other fields ranging from physical health measurements, medical studies, and personal localization. Although step counters are not commonly used in clinical research, steps per days can now be considered as vital signs for crucial medical information in the future [24].

Before activity monitors can be adequately used for clinical research, they must first prove there is a connection between steps taken and future occurrence of diseases. Having accurate measurements of physical activity is crucial during medical studies

as any slight mismeasurement could result in obscuring actual links between physical activity and certain ailments that affect physical movements [40].

Another future application of this research can be implemented in personal localized positioning systems based on steps detection and the calculated distance the user has travelled [41]. A group of researchers from Seoul National University worked on a method to determine the user's location in indoor settings. Due to the lack of GPS measurements for indoor environments, these researchers resorted to step detection in combination with pedestrian dead reckoning to capture the user's current location. They are able to achieve step detection through several conditions such as handheld texting, in pocket of pants front and back [14]. For indoor position to be accurate the direction in which the user is walking and their heading is crucial to be determined, which is accomplished by using the magnetometers to find their absolute direction [42]. Furthermore, the algorithm developed in this paper can be used on other data sets from other studies. The data set from [43] is of accelerometer data collected from two groups of individuals who are sighted and blind using the aid of walking cane or guide dog. The paper [43] discusses the interesting differences between the sighted and blind walkers.

Here, we perform the analysis assuming the data we are analyzing is walking signal, in other words, walking is granted, however, the salience algorithm can be employed to detect whether the subject is actually walking in the first place. The proposed method is computationally inexpensive, performs well on slow walking data, and is a low battery consumer. These qualities make this approach efficient and highly practical.

7 Conclusions

A biomechanical process (called Pronation [44]) occurs during ambulation that allows the body to naturally absorb shock as each foot strikes the ground. Since the head is the farthest body location from the ground, the accelerations become significantly dampened [45], making it an unsuitable signal to detect steps. On the other hand, applications in fitness tracking, indoor navigation, user authentication, vision enhancement for the visually impaired will comprise a considerable portion of the smart glasses market and they are all dependent on accurate step counts. With this motivation, we implement and evaluate an accurate and reliable method for step detection using head acceleration signals. Furthermore, our algorithm achieves superior step counting performance when applied to the acceleration signal collected from other body parts. Our results demonstrate the feasibility of our salience-based algorithm for performing pedometry on smart glasses.

References

1. Park, Y., Yun, S., Kim, K.H.: When IoT met augmented reality: Visualizing the source of the wireless signal in AR view. In: MobiSys 2019 - Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services, pp. 117–129 (2019)
2. Level Smartglasses by VSP Global. https://appadvice.com/app/level-smart-glasses/1336625786. Accessed 01 Aug 2020
3. Vue Glasses by Vigo Technologies Inc. https://www.vueglasses.com/. Accessed 01 Aug 2020
4. GlassUp F4 AR Smartglasses. https://www.glassup.com/en/f4/. Accessed 01 Oct 2020

5. MAD Gaze Smartglasses. https://www.madgaze.com/. Accessed 01 Aug 2020
6. abEye Smart Eyewear. https://www.abeye.tech/. Accessed 01 Aug 2020
7. Ellcie's Smartglasses. https://ellcie-healthy.com/en/home/. Accessed 01 Aug 2020
8. eSight by eSight Corp. https://esighteyewear.com/. Accessed 01 July 2020
9. OrCam MyEye 2.0. https://www.orcam.com/en/myeye2/. Accessed 01 Aug 2020
10. Amini, N., et al.: Design and evaluation of a wearable assistive technology for hemianopic stroke patients. In: Proceedings of the 2020 International Symposium on Wearable Computers, pp. 7–11 (2020)
11. Rhudy, M.B., Mahoney, J.M.: A comprehensive comparison of simple step counting techniques using wrist- and ankle-mounted accelerometer and gyroscope signals. J. Med. Eng. Technol. **42**(3), 236–243 (2018)
12. Pan, M.S., Lin, H.W.: A step counting algorithm for smartphone users: design and implementation. IEEE Sens. J. **15**(4), 2296–2305 (2014)
13. Zhang, H., Yuan, W., Shen, Q., Li, T., Chang, H.: A handheld inertial pedestrian navigation system with accurate step modes and device poses recognition. IEEE Sens. J. **15**(3), 1421–1429 (2015)
14. Park, S.Y., Heo, S.J., Park, C.G.: Accelerometer-based smartphone step detection using machine learning technique. In: International Electrical Engineering Congress, iEECON 2017, pp. 1–4 (2017)
15. Kim, J.W., Jang, H.J., Hwang, D.H., Park, C.: A step, stride and heading determination for the pedestrian navigation system. J. Glob. Position. Syst. **3**(1–2), 273–279 (2004)
16. Rantalainen, T., Karavirta, L., Pirkola, H., Rantanen, T., Linnamo, V.: Gait variability using waist-and ankle-worn inertial measurement units in healthy older adults. Sensors **20**(10), 2858 (2020)
17. Saeedi, R., Amini, N., Ghasemzadeh, H.: Patient-centric on-body sensor localization in smart health systems. In: 2014 48th Asilomar Conference on Signals, Systems and Computers, pp. 2081–2085 (2015)
18. Amini, N., Vahdatpour, A., Dabiri, F., Noshadi, H., Sarrafzadeh, M.: Joint consideration of energy-efficiency and coverage-preservation in microsensor networks. Wirel. Commun. Mob. Comput. **11**(6), 707–722 (2011)
19. Amini, N., Miremadi, S.G., Fazeli, M.: A hierarchical routing protocol for energy load balancing in wireless sensor networks. In: Canadian Conference on Electrical and Computer Engineering, pp. 1086–1089 (2007)
20. Brajdic, A., Harle, R.: Walk detection and step counting on unconstrained smartphones. In: UbiComp 2013 - Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 225–234 (2013)
21. Naqvib, N.Z., Kumar, A., Chauhan, A., Sahni, K.: Step counting using smartphone-based accelerometer. Int. J. Comput. Sci. Eng. **4**(5), 675 (2012)
22. Crea, S., Cipriani, C., Donati, M., Carrozza, M.C., Vitiello, N.: Providing time-discrete gait information by wearable feedback apparatus for lower-limb amputees: usability and functional validation. IEEE Trans. Neural Syst. Rehabil. Eng. **23**(2), 250–257 (2015)
23. Ngueleu, A.M., et al.: Design and accuracy of an instrumented insole using pressure sensors for step count. Sensors **19**(5), 984 (2019)
24. Bassett, D.R., Toth, L.P., LaMunion, S.R., Crouter, S.E.: Step counting: a review of measurement considerations and health-related applications. Sports Med. **47**(7), 1303–1315 (2016)
25. Pham, V.T., Nguyen, D.A., Dang, N.D., Pham, H.H., Tran, V.A., Sandrasegaran, K., Tran, D.T.: Highly accurate step counting at variouswalking states using low-cost inertial measurement unit support indoor positioning system. Sensors **18**(10), 3186 (2018)
26. Genovese, V., Mannini, A., Sabatini, A.M.: A smartwatch step counter for slow and intermittent ambulation. IEEE Access **5**, 13028–13037 (2017)

27. Zhang, X., Xu, W., Huang, M.C., Amini, N., Ren, F.: See UV on your skin: an ultraviolet sensing and visualization system. In: Proceedings of the 8th International Conference on Body Area Networks, pp. 22–28 (2013)
28. Case, M.A., Burwick, H.A., Volpp, K.G., Patel, M.S.: Accuracy of smartphone applications and wearable devices for tracking physical activity data. JAMA – J. Am. Med. Assoc. **313**(6), 625–626 (2015)
29. Wang, L., Liu, T., Wang, Y., Li, Q., Yi, J., Inoue, Y.: Evaluation on step counting performance of wristband activity monitors in daily living environment. IEEE Access **5**, 13020–13027 (2017)
30. Lee, H.H., Choi, S., Lee, M.J.: Step detection robust against the dynamics of smartphones. Sensors **15**(10), 27230–27250 (2015)
31. Abadleh, A., Al-Hawari, E., Alkafaween, E.A., Al-Sawalqah, H.: Step detection algorithm for accurate distance estimation using dynamic step length. In: Proceedings - 18th IEEE International Conference on Mobile Data Management, MDM, pp. 324–327 (2017)
32. Kang, X., Huang, B., Qi, G.: A novel walking detection and step counting algorithm using unconstrained smartphones. Sensors **18**(1), 297 (2018)
33. Scholkmann, F., Boss, J., Wolf, M.: An efficient algorithm for automatic peak detection in noisy periodic and quasi-periodic signals. Algorithms **5**(4), 588–603 (2012)
34. Caserman, P., Krabbe, P., Wojtusch, J., von Stryk, O.: Real-time step detection using the integrated sensors of a head-mounted display. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings, pp. 003510–003515 (2017)
35. Burns, A., et al.: SHIMMERTM - a wireless sensor platform for noninvasive biomedical research. IEEE Sens. J. **10**(9), 1527–1534 (2010)
36. Apostolopoulos, I., Coming, D.S., Folmer, E.: Accuracy of pedometry on a head-mounted display. In: Conference on Human Factors in Computing Systems - Proceedings, pp. 2153–2156 (2015)
37. Amini, N.: Transmission Power Management for Wireless Health Applications. UCLA Electron. Theses Diss. (2012)
38. Mertens, C., Grenez, F., Schoentgen, J.: Speech sample salience analysis for speech cycle detection. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH (2009)
39. Miraftabi, A., et al.: Macular SD-OCT outcome measures: comparison of local structure-function relationships and dynamic range. Investig. Ophthalmol. Vis. Sci. **57**(11), 4815–4823 (2016)
40. Ewald, B., McEvoy, M., Attia, J.: Pedometer counts superior to physical activity scale for identifying health markers in older adults. Br. J. Sports Med. **44**(10), 756–761 (2010)
41. Vahdatpour, A., Amini, N., Sarrafzadeh, M.: Toward unsupervised activity discovery using multi-dimensional motif detection in time series. In: IJCAI International Joint Conference on Artificial Intelligence (2009)
42. Ilkovičová, Ľ., Kajánek, P., Kopáčik, A.: Pedestrian indoor positioning and tracking using smartphone sensors, step detection and map matching algorithm. In: International Symposium on Engineering Geodesy pp. 20–22 (2016)
43. Flores, G.H., Manduchi, R.: WeAllWalk: an annotated dataset of inertial sensor time series from blind walkers. In: Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility - ASSETS 2016, vol. 11, no. 1, pp. 1–28 (2016)
44. Chan, C.W., Rudins, A.: Foot biomechanics during walking and running. In: Mayo Clinic Proceedings, vol. 69, no. 5, pp. 448–461. Elsevier (1994)
45. Maslivec, A., Bampouras, T.M., Dewhurst, S., Vannozzi, G., Macaluso, A., Laudani, L.: Mechanisms of head stability during gait initiation in young and older women: a neuro-mechanical analysis. J. Electromyogr. Kinesiol. **38**, 103–110 (2018)