Wearable Sensing Framework for Human Activity Monitoring

Mostafa Uddin, Ahmed Salem, Ilho Nam, and Tamer Nadeem
Old Dominion University
{muddin, asalem, inam, nadeem}@cs.odu.edu

ABSTRACT

Wearable computation is getting integrated into our daily life day by day. In this work, we propose a generic framework to continuously monitor users' daily activities. The framework proposes light computation tasks on the wearable device to reduce the amount of data communicated between the wearable, and its host. A 9-axis wristbands are being used to collect user's activities. The collected signals are subject to light weight preprocessing and segmentation on the wearable device prior sending to the host, were it goes through activity detection algorithms. In this paper, we elaborate the feasibility of the proposed framework thru presenting two case studies.

Categories and Subject Descriptors

C.2.1 [**Computer Communication and Networks**]: Network Architecture and Design—*Wireless communication*; C.5.3 [**Computer System Implementation**]: Microcomputers—*Portable devices*; I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*Motion*

Keywords

Wearable Computing, Activity monitor, Inertial measurement unit, Data Communication, BLE

1. INTRODUCTION

Recently, wearable devices got a lot of interests and wide acceptance due to their small sizes, reasonable computation power, and practical power capabilities. These wearable devices loaded with sensors (e.g. accelerometer, gyroscope) provides a good candidate to monitor users' daily behavior (e.g. walking, jogging, smoking) [4]. Nowadays wearable devices are used in several domains (e.g. activity detection), were health monitoring is one of the prominent.

Recent advancement of wearable technology have resulted in utilization of wearable and non-intrusive systems for health and activity monitoring. Such continuous monitoring of life and daily activities, motivate the users to maintain healthy living style. Moreover, wearable technology has empowered the user to quantify, and take control of their lifestyle. In the long run, such consciousness among people will help the society to be healthy and productive. Maintaining such healthy life will also reduce the cost of health-care by allowing the people to spend less time in the hospital or make fewer visit to the doctor.

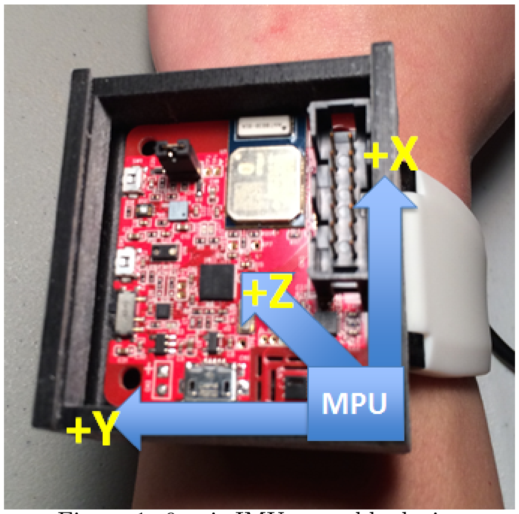

Figure 1: 9-axis IMU wearable device

Wearable technology faces three main challenges: communication capacity, computation power, and limited energy of the wearable device [10]. In this paper, we propose a framework to continuously monitor user activity using wearable devices. The framework provides a mechanism for wearable device to reduce the overhead of data communication. Thus, it allows wearable device to manage its power consumption in more efficient way. Furthermore, the proposed framework reduces the typical data processing overhead required by the monitoring applications. Moreover, it provides more flexibility for the applications to configure their monitoring requirements.

In the proposed framework we split the activity monitoring task between both the wearable device and a corresponding host such as the smartphone. The wearable device is responsible for collection, cleaning, and segmentation of the raw data. On the other hand, the monitoring application on the host device will process these data segments according to its activity detection interest.

Moreover, the framework provides the application with full control of the data collection phase (e.g. sensor data selection, and sensor frequency). We envision this separation approach to reduce the communication overhead in trade of light computation on the wearable device. The proposed work adopts an architecture that can be easily used by various platforms (e.g. iOS, Android). As a proof of a concept, we preset two monitoring activities: hand washing, and drinking. These studies shows how the proposed framework using 9-axis inertial measurement unit (IMU) wristband (i.e. MPU-9250 [1]) shown in Fig. 1, and an Android phone to detect these activities.

2. BACKGROUND AND CHALLENGES

Wearable sensor devices are becoming extremely useful in providing accurate and reliable information on people's activities and behaviors. Significant amount of research is currently undergoing in the development of a smart sensing systems to monitor human activities [14]. For example, measurement of body temperature using wearable sensors is used for determining the physiological condition as well as for other things such as activity classification [18]. Wearable motion sensors (e.g. accelerometer, and gyroscope) have been used for many purposes such as, human fall [14], body motion analysis [7], and postural orientation [5]. Even in sports and training there is an increasing trend of using various wearable sensors [13].

Research and scientific communities are working hard to come up with smart wearable devices for continuous monitoring of different human activities twenty four seven. Several challenges stands to design, development, implementation, and utilization cum continuous monitoring. In the following, we shed some light on these challenges and how the proposed framework can deal with them.

- *Heterogeneity in sensing*, is an important issue, where different wearable devices' hardware implies different sensing accuracy, sensing frequency, and transmission rate. Developing an activity detection algorithm, in addition to dealing with a variety of wearable devices' sensing and transmission capabilities, is challenging [4].

- *Data accuracy* is the challenge of pre-processing the data (e.g., cleaning, activity/non-activity recognition) before extracting the features. The process of preprocessing involves removing noises, motion artifacts, sensor error, data formatting, data normalization, and data synchronization. In continuous monitoring scenario, the sensor data might have many portions that represents no activity. Detection and differentiating such activity and non-activity region from the time-series sensor data is a challenging task [10].

- *Activity detection* is the challenge of appropriate features extraction and building recognition algorithm/-model. Different activity detection might require different set of features and recognition algorithm. There is no common standard of selecting right features for all activity detection applications. Moreover, depending on the application, achieving high accuracy of activity detection can be crucial and challenging [11].

- *Communication* between the wearable, and its host is also a big challenge. Sensors continuously collecting data, results in large portions of information to be transmitted to the host. A naive approach would be continuously send all data to the host. Even with the low energy consumption of the Bluetooth Low Energy (BLE) technology, such an approach can leads to an intense energy consumption of the wearable's resources [4].

- *Context awareness* whether the wearable is being used or not, and whether it has been left out by mistake or intentionally. Many surveys reported users forgetting to put on their devices. In this case, host apps should be smart enough to notify the user that the wearable is out of range. On the other hand the wearable should be realize its host is unreachable and its activities. This can minimize unnecessary energy consumption which is another major concern in wearable design [17]

- *Security, and Privacy* of transmitted data is another challenge with using wearables. Surveys reports shows how cautious users are with exposing there sensing information (e.g. medical data) while transmitted between the wearable, and its host [8].

- *Performance* is another challenge, due to scare resources (e.g., Memory, CPU) of wearable devices. Such a challenge enforce us to run any tasks as efficient as possible to avoid any waste of resources in terms of power, and computation.

3. PROPOSED FRAMEWORK OF WEARABLE SENSING

3.1 Design Principles

In this section, we list, and discuss major design issues we consider.

1. *Reduce data communication overhead between the host, and wearable.* Typically, a continuous monitoring wearable will send all its data to the host. In this scenario, the host will receive a huge set of data with a significant portion of informationless data. In this work, we increase the role of the wearable to filter out data from inactive periods. We envision this to decrease the communication overhead, with minimal effect on the wearable resources. If we assume communication through Bluetooth v4.0 –which is widely for wearables, with data rate of 1Mbps [16], and a smart watch with a typical battery size of 300mAh [6]. For example, the watch is sending 40 samples of accelerometer data per second (i.e. 480 bytes), to the host. In this case, continuous data sending will deplete the host's battery in 2 days. Moreover, if the wearable sends extra sensing data (e.g. gyroscope, and magnetometer), the battery can last less then 12hr. This highlights the inefficiency of sending all data to the host device. It is worth mentioning that Bluetooth v4.0 can is not likely to reach the 1Mbps data rate in practice, which increase communication overhead, and elevate the importance of what we propose. Our design objective is to send only the required sensor data according to the application requirement in order to reduce the data communication, and save more battery life.

2. *Decouple activity recognition from data processing.* Most of the motion sensor based human activity recognition application apply similar data processing technique Therefore, decoupling the activity application from data processing, allows multiple activity recognition applications to reuse common data processing components. Thus it also improves the efficiency of running multiple activity applications at a time.

3. *Providing flexibility to the Activity Application developer.* In designing the framework,we will provide a set of APIs that enables the application developer to configure data processing components in the wearable devices. Moreover, we provide the developer with flexibility of smartly handling the transmission of sensing data from wearable device to host device. Thus, reduce the communication overhead, and reuse the data processing component of the wearable devices.

3.2 Framework Overview

At least two devices are involved in wearable human activity monitoring applications; 1) wearable device, and 2) host device. Wearable device is responsible for generating the sensing data for activity recognition. Typically wearable devices has little storage and computation power. It also has low power battery (e.g., 300mAh). On the other hand, host device is computationally more powerful compare to wearable device, and it has much larger storage capacity. In addition, it has comparatively larger battery (e.g., 2500mAh). Therefore, host device typically does computational job like feature extraction, activity recognition etc. In the framework, we have separated our components between host device and wearable device. Figure 2 shows the detail architecture of our wearable human activity monitoring framework. In following subsection we have described each of the components in details.

3.2.1 Wearable Device

The Wearable device is responsible for collecting the sensor data, applying data preprocessing, and provide the required/desired sensing data to the host device. Following are the brief description of the components in the wearable device as shown in figure 2.

Data Collection, Responsible for collecting raw sensor data from different sensors, and forward it to the "Data Preprocessing" components (e.g. Pebble watches [2] have similar subsystem called "data logging"). The "data logging" API is responsible for collecting raw sensing data and has the mechanism for short-term data buffering. Typically data logging can either send its buffered data to the application running in the host device (Android or iOS) or it can send it to the application that is running in the wearable device (i.e., smart watch). In our case instead of sending the data to the host, we send it to the the application, which basically consists of our data handling components running in the wearable device. Note that sensor data are time series data, therefore in addition with the raw sensor data we also need to record the timestamp.

Data Preprocessing Often raw wearable sensor data has noises, motion artifacts, and sensor errors. Therefore, a preprocessing of the raw data is necessary before taking further processing on the data. The Data Preprocessing components involves in, (1) filtering out unusual data to remove artifacts, (2) Interpolating missing sensing data, (3) removing high frequency noises, and (4) normalization and synchronization of the sensor data. In filtering artifacts, one of the simple technique is to apply threshold-based methods to filter unusual sensor data [9]. Typically statistical approach is taken to interpolate the missing data points [3]. In order to remove high frequency noises from the sensor data several technique have been applied that involved methods in frequency domain such as such as power spectral density (PSD) fast Fourier transforms (FFT). However, low-pass/high-pass filtering tools are among the common and simple to remove the frequency noises from the sensing data. When the data is gathered from numerous wearable sensors, normalization and synchronization of sensor data is the most essential task. The Data Preprocessing takes care of data formatting, data normalization, and data synchronization. Once data is preprocessed, it is forwarded to the "Data Segmentation" component.

Data Segmentation: After preprocessing the sensing data, not necessarily all sensing data is useful for collecting the features and activity recognition. Often in continuous sensing, there are times when there are no sign of any real activity. The main action of the "Data Segmentation" component is to identify the activity and non-activity region of the preprocessed time-series data. Then for the activity region, it forward the sensor data to the upper layer component, and for the non-activity region, it only sends the meta-data of the starting and the ending of the non-activity region. The common technique of identifying the activity and non-activity region, is to use a fixed size time window, then slide the window over the sensing data to see where it shows high activity above certain threshold to confirm the activity region. However, one of the challenge of such technique is to find out proper size of the time window and activity threshold. There is a rich history of research work in speech domain, about detecting speech segments, or silence/unvoiced portion removal from the speech signal (an example of time-series signal). We beileve adopting some of the suitable technique that don't depend on fixed time window, such as entropy-based[15], stochastic approach [12], etc., could help us to segment the activity region from the non-activity region. Note that ,not necessarily activity can be seen from all the sensors or all the axis at a time. For example, there can be scenario, where gyroscope 'x'-axis is showing activity pattern, but at the same time accelerometer 'x'-axis might not show any activity pattern. Note that, The activity and the non-activity data segmentation is done per-axis per-sensor.

Sensing Proxy: Sensing Proxy maintains a list of "Rules" which are set by the Wearable Sensing Middleware from the host device through Local Controller. A "Rule" is basically some syntactic expression that represents, what sensor data to send to the host device and what to filter out, under what condition. Note that, the Wearable Sensing Middleware make sure the "Rules" that are set in the Sensing Proxy are conflicting each other. For example, a Rule can be specific as following; *Activity in gyroscope x-axis and activity in gyroscope z-axis send accelerometer x,y -axis.*, or it can be very general; *Activity in accelerometer x,y,z-axis send all sensor data.* The basic idea of the Sensing Proxy component is to filter out unnecessary sensing data that are not the interest of the Activity Applications running in the host device. Thus it reduces the overhead of sending sensing data

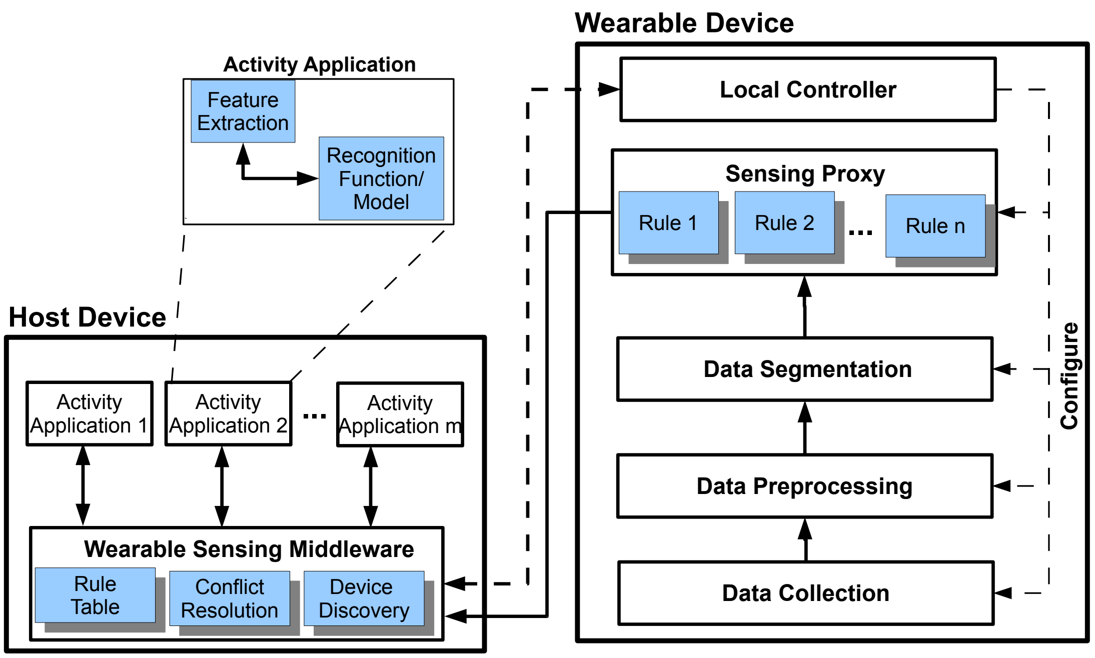

Figure 2: Wearable Sensing Framework for Human Activity Recognition.

to improve the power efficiency. Note that, ”Rules” are basically simple conditions (e.g., comparison with a threshold value), that has no computation intensive services. We believe, such ”rule” based scheme can allow the developer to efficiently reduce the required sensor data from the wearable device with little overhead.

Local Controller: Local Controller is responsible for directly interacting with the Middleware that is running in the host device (2). The interaction involves in helping the Middleware to configure the parameters of the components running in the wearable device. For example, through Local Controller, the host device Middleware can set the frequency of collecting sensor data in the ”Data Collection” component. In another example, it can also set the cut-off frequency of the low pass filter in the ”Data preprocessing” component. Furthermore, Middleware also tell the Local Controller, what ”rules” to add in the Sensing Proxy component. Before adding any new ”rules”, Local Controller remove all previous ”rules” from the Sensing Proxy component. The Local Controller also respond to the Middleware during device discovery process.

3.2.2 Host Device

In the host device we have two components Activity Application and Wearable Sensing Middleware. The Wearable Sensing Middleware provide the platform for the host device to run Activity Applications and provide the APIs to smartly interact with wearable sensing devices.

Activity Application: The Activity Application get registered with the Wearable Sensing Middleware and let it know it’s interest of sensor activity. The Activity Application mention its interest as a ”rule”, which is a syntactic expression, through our APIs. The ”rule” represents, under what condition, which sensor data the application is interested in. For example, an Activity Application can mention its interest to gyroscope y-axis data if there is any activity both in accelerometer y and z -axis. On the other hand, an Activity Application can show interest in all sort of sensor activity, without filtering out any sensor activity based on pre-condition. Thus, our framework provides flexibility to the Activity Application developer for tuning their sensor data requirement. Once It get the sensor data from the Wearable Sensing Middleware, it applies its own feature extraction and activity recognition algorithm. Activity Application can also provide some parameters, such as frequency of sensor data collection, cut-off frequency etc., to the Wearable Sensing Middleware through APIs.

Wearable Sensing Middleware(WSM): This component is responsible for wearable device discovery and maintaining the connectivity with the wearable device. It also let the registered Activity Application know if the wearable sensing device is out-of-reach or offline. This component in collaboration with the Activity Application send command to the Local Controller of the wearable device to configure or add new rules. WSM maintain a table of rules and configuration for the registered running Activity Applications. It is the responsibility of the WSM component, is to resolve the conflict of the rules before sending the request of adding new ”rules” in the Local Controller. In general, WSM takes the union of the ”rules” of the applications, and send it to the wearable device. Note that union of the ”rules” represents the super-set of all the sensor data requirement by the running activity applications in the host device. Once receiving the sensing data from the wearable device, WSM again apply the ”rules” provide by the activity application, before sending it to the application.

4. CASE STUDIES

4.1 Hand-Washing

The habit of regular and proper hand-washing is absolutely essential for hygienic life. Therefore monitoring the habit of hand-washing can be a useful use-case scenario for

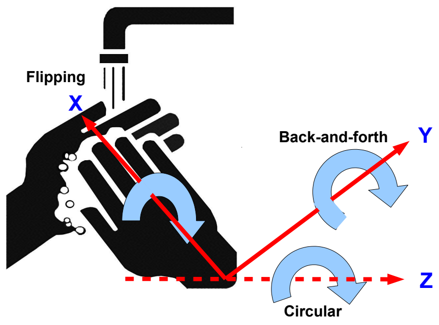

Figure 3: Hand motion around three axis during hand washing.

human activity application. In hand-washing case scenario, the user wear a 9-axis IMU wearable device in their right hand, and an Android phone is used for our host device.

In hand washing, there can be motion in different direction. But, we see two common hand motion, one is the "flipping motion" that make the hand front facing to back facing or vice-versa. This "flipping motion" is around the x-axis (from figure 3). The other motion is rubbing motion, which can be two type; circular motion and back-and-forth motion. The "circular" rubbing motion is around z-axis, and the "back-and-forth" rubbing is around the y-axis (figure 3). The figure 4, shows the gyroscope reading of 'x', 'z' -axis. In the case scenario, we capture the sensor reading from turning on the tape, washing the hand, and then to turning off the tape. For shake of clarity and space we didn't plot the gyroscope 'y'-axis reading. However it shows similar reading of rubbing motion as the 'z'-axis but with less clarity.

In figure 4, the big spike in the gyroscope 'x'-axis reading represents the flipping motion of the right hand. Note that, between the flipping motions, some portion of 'z'-axis shows variation with high magnitude compare to other. This high magnitude of 'z'-axis data represents the situation when right hand is circularly rubbing the left hand. According to our framework, wearable device collect the raw gyroscope sensor data, and apply data preprocessing. Then the data segmentation remove the non-activity region, which in our case remove 15% of the sensor data. Finally, the hand-wash application set the rules of receiving only the gyroscope x,z -axis data, if there are activity in all 3 gyroscope axis reading. After passing through the rule, the hand-wash application apply feature extraction on the receiving the sensor data, and then apply recognition algorithm/model to detect the hand washing activity.

4.2 Drinking

Drinking is mandatory to avoid dehydration, and maintain vital activities. Any decrease in the daily water consumption rates will directly impact the health status, especially if accompanied by a another sickness (e.g. low sugar level). The motivation to monitor drinking is very clear, hence we should analyze the action.

Drinking consists of the following actions: it starts, and ends with hands pointing down, followed by raising the hand to hold the cup, then to the mouths level, then tilting the wrist up to drink. Upon quenching the users' thirst the

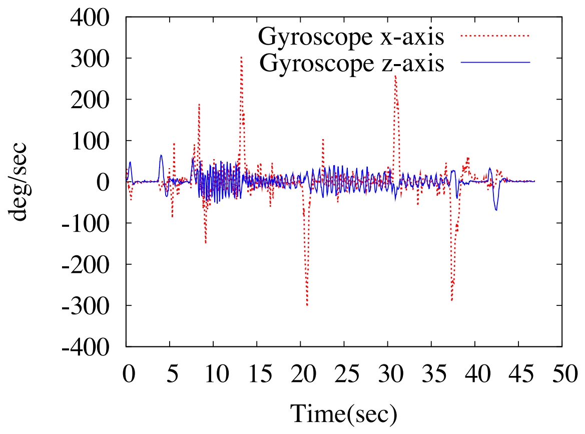

Figure 4: Gyroscope sensor reading for axis 'x' and 'z' during hand washing.

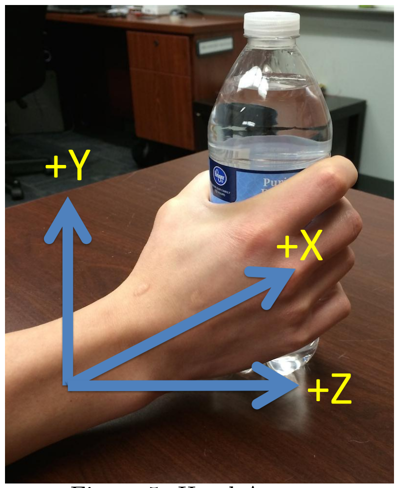

Figure 5: Hand Axes

previous actions are reversed. We repeated this action few times using an IMU wristband.

Drinking has two events we can detect: raising/lowering the arm to the mouth level, and tilting the wrist up/down to drink. Fig. 5 show two initial 2 spikes representing raising the hand then tilting the wrist to drink, followed by a steady period representing the drinking time. At the end the figure shows spikes similar to the initial ones but in the opposite direction representing the action of tilting back the wrist, and lowering the arm.

Mapping the Drinking action to the proposed framework is done as follows: *Data collection* is done by the IMU to collect raw data. *Data preprocessing* is passing the data on a low pass filter to remove some noise. *Segmentation* involves removing the low activity section, which in this case eliminates 59% of the data to be sent as shown in Fig. 6. Upon acquiring the segments expressing the activity, they communicated back to the host, which then applies activity recognition algorithms to detect the presence of drinking activity or not.

5. DISCUSSION

Communication vs. Computation: Wearable devices tends to use as data collectors, that send all their sensed data

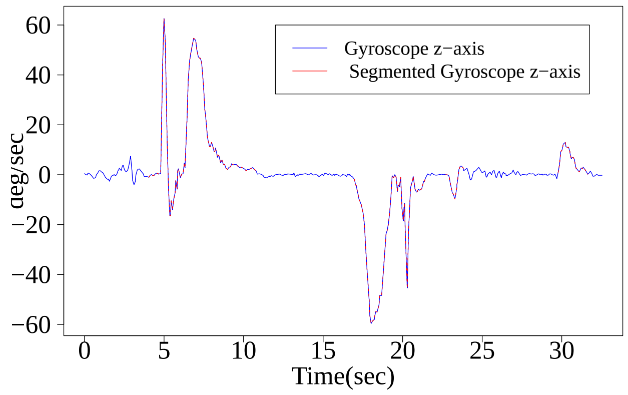

Figure 6: Gyroscope sensor reading for axis 'z' during drinking.

back to the host. In our proposed framework, in addition with the data collection, we also run light weight algorithms such as low pass filtering, normalization, activity/non-activity region detection etc., in wearable device. This little computations in wearable device allows the host device to flexibly eliminating the unnecessary sensing data (using "rule"). Hence, reduce the overhead of data communication. Our case study shows the likelihood of saving 15-50% of the communication overhead by smartly filtering out irrelevant sensing data. However, the trade off between the amount of computation and the reduction of data communication needs to evaluated carefully in real-scenario.

Conflict of Configuration: Our framework provides flexible API to the activity monitoring applications, running in host devices, for configuring the data processing parameters, and setting "rules" for filtering out the irrelevant sensing data for communication. Note that, our framework allows multiple activity applications to access same sensor of the same wearable device. Such scenario can create conflict of configuration or setting rules in the wearable device. For example, lets assume `App1` and `App2` set the cut-off frequency of the low pass filter to 20Hz and 40Hz respectively. According to our framework "Wearable Sensing Middleware" will resolve this conflict by setting the cut-off frequency to 40Hz in the wearable device. Later "Wearable Sensing Middleware" needs to apply 20Hz low pass filter on the receiving sensing data before sending it to the `App1`. Therefore such conflict put extra data processing overhead on the host device.

Single application using Multiple Wearable Devices: An activity application might require sensing data from multiple wearable devices. Assume, an activity monitoring application is collecting sensor data from two wearable wrist band devices. In such scenario, how to collect the sensor data from two devices, and how to synchronize or process the data together, is a challenging task.

In our framework, Wearable Sensing Middleware can take the responsibility of collecting, synchronizing, and processing the data from multiple wearable devices before sending it to the Activity Application. Thus, our framework can provide APIs, and abstract the complexity of handling multiple wearable devices, to the Activity application developer.

6. ACKNOWLEDGMENT

This material is based in part upon work supported by the National Science Foundation under Grant Numbers CNS-1405882 and CNS-1454285.

7. REFERENCES

[1] 9-axis imu wearable device,http://goo.gl/bmks0l.
[2] Pebble watch, http://developer.getpebble.com/docs/.
[3] D. Apiletti et al. Real-time analysis of physiological data to support medical applications. *IEEE Transactions on Information Technology in Biomedicine*, 2009.
[4] V. Ayatollahitafti et al. Requirements and challenges in body sensor networks: A survey. *Journal of Theoretical and Applied Info Tech*, 72(2), 2015.
[5] Y. Chuo et al. Mechanically flexible wireless multisensor platform for human physical activity and vitals monitoring. *Biomedical Circuits and Systems, IEEE Transactions on*, 2010.
[6] T. P. Crompton. *Battery reference book*. Newnes, 2000.
[7] Y.-C. Kan and C.-K. Chen. A wearable inertial sensor node for body motion analysis. *Sensors Journal, IEEE*, 2012.
[8] O. D. Lara and M. A. Labrador. A survey on human activity recognition using wearable sensors. *Communications Surveys & Tutorials, IEEE*, 15(3):1192–1209, 2013.
[9] Y. Mao et al. An integrated data mining approach to real-time clinical monitoring and deterioration warning. KDD '12. ACM, 2012.
[10] E. McAdams et al. Wearable sensor systems: the challenges. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 3648–3651. IEEE, 2011.
[11] A. Parate et al. Risq: Recognizing smoking gestures with inertial sensors on a wristband. In *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '14, pages 149–161, New York, NY, USA, 2014. ACM.
[12] G. Saha et al. A new silence removal and endpoint detection algorithm for speech and speaker recognition applications. In *In the Proceedings of 11th National Conference on Communications (NCC)*, 2005.
[13] P. Salvo et al. A wearable sensor for measuring sweat rate. *Sensors Journal, IEEE*, 2010.
[14] T. Shany et al. Sensors-based wearable systems for monitoring of human movement and falls. *Sensors Journal, IEEE*, 2012.
[15] K. Waheed et al. A robust algorithm for detecting speech segments using an entropic contrast. *45th IEEE Midwest Symp. Circuits and System*, 2002.
[16] R. Want, B. Schilit, and D. Laskowski. Bluetooth le finds its niche. *IEEE Pervasive Computing*, (4):12–16, 2013.
[17] J. Williamson et al. Data sensing and analysis: Challenges for wearables. In *Design Automation Conference (ASP-DAC), 2015 20th Asia and South Pacific*, pages 136–141. IEEE.
[18] J. Winkley et al. Verity: an ambient assisted living platform. *Consumer Electronics, IEEE Transactions on*, 2012.