Do not Lean in: A zoom-in adaptation using activity detection in front of the laptop

Erzhen Hu
eh2qs@virginia.edu

Jianguo Chen
jc2hk@virginia.edu

Johannes Johnson
jjj6qt@virginia.edu

Yike Guo
yg7jg@virginia.edu

ABSTRACT
Bad postural habits are dangerous health challenges when people are focused on working, who might keep the same wrong posture in front of the computer for a long time without moving. The difference between text sizes of different web pages and documents and the limited size of laptop screens result in the fact that context size may sometimes be too small to read comfortably, forcing us to lean in to read text better at the cost of keeping a good posture. Prior research points out that the wrong posture due to the small text on the computer casts great challenge to human's vertebrates and brains. Alternatively, our hands may be occupied with other tasks when we try to interact with a device (for example, when we are trying to cook from a written recipe). In this paper, we proposed a zoom in adaptation and activity recognition method. The proposed method adopted the gyroscope data and extracted critical features in the frequency domain. Moreover, small activities like typing or moving items in front of laptop were taken into account to mitigate the negative influences of random noises. Finally, a thorough experimental analysis was presented by taking into account various complicated situations, and showed that the proposed method is able to achieve relatively stable performance.

INTRODUCTION
Poor postures can have great harm to people's health. According to the U.S. National Library of Medicine[1], poor postures might be caused by certain muscles tightening up or shortening while others lengthen, leading to injuries and health problems including kyphosis and scoliosis. Poor posture with continuous computer use deforms the thoracic and cervical spine and could result in musculoskeletal disorders of neck and shoulders. When using a computer, small text size might cause users to unconsciously hunch over the desk to see more clearly. When the text size changes between different web pages, users might not automatically change back to their original positions, which causes great danger to their health. The poor computer using posture might also cause people to lean forward even if they are not using a computer.

Prior attempts including sit to stand workstation (e.g. Treadmill desk) and desk adjustments to alleviate the posture problem. However, it is admitted that not all workplace has enough space and budget to install such sit to stand workstations for every worker, and the possibility still exists when workers need to get back to the sitting position that they gradually change to the same poor posture again.

Our research explores another possible option for alleviating poor postures without too much installation requirements: an interface that could zoom in automatically based on the position of the user and other factors by detecting the lean in posture, so that the user have to force back and the possibility of the user leaning to the screen would ideally decrease. Based on this motivation, we attempted two scenarios: (1) The user' s hands are occupied and he/she intends to zoom in the screen using gestures. We solved this situation with a wrist rotation to trigger the zoom adaptation. (2) The user is sitting in front of the laptop without noticing his/her bad posture. We solved this scenario by detecting the lean in posture.

We used the pyphox app installed on a smartphone which is strapped to the user's upper arm to collect data on the user's wrist movements as well as to detect a user's lean in motion. For data collection, we designed four tasks for the participants, including moving in front of the computer due to text size changes and picking items up beside the computer. We evaluated our design based on the two scenarios to verify if the approaching intention type and gestures are correctly detected.

Contribution
In summary, given all the issues we found with current poor posture intervention tools and through a pilot study, we designed an interface that could automatically zoom in and out according to the page sizes on the screen based on two scenarios. Our evaluation demonstrated the effectiveness of our detection and provided insights into other possible scenarios as well.

RELATED WORK
Extracting spatio-temporal information is an important focus in the design of human-computer interfaces. In particular, gesture recognition presents a challenge for researchers because it involves both detection and classification. There are several popular approaches for wearable devices to implement real-time recognition.
The time domain approaches include thresholding, peak detection, autocorrelation, etc. The frequency domain approaches focus on the frequency content of successive windows of measurements based on short-term Fourier transform (STFT), FFT, and continuous/discrete wavelet trans-

forms (CWT/DWT), and can generally achieve high accuracy, but suffer from either resolution issues or computational overheads.

In the context of wearable devices. a variety of sensor configurations and technologies has been employed. Hand-gesture recognition systems based around wearable inertial measurement units that utilizes SVMs and recurrence quantification analysis [3]. Previous approaches have also combined sensor data with LSTM-based recurrent architectures [2]. Past work has not been limited to these two architectures and includes different sensor types as well as sensors coupled with mobile devices. This work pursues mobile devices for purposes of gesture recognition and depth prediction.

PILOT STUDY

Wizard of oz

To get more insight regarding "how our zoom-in adaptation can be built in two different scenarios", We conducted a Wizard of oz experiment. The user followed the moderator's instructions to perform certain interactions with a phone strapped to his wrist (via multiple rubber bands) with the pyphox app installed. The system operator tracked sensor data from the pyphox app to determine what data sourcescould be useful in classifying user movements.The moderator tested each gesture as outlined in our ideation report with the user, telling the user to also discuss their interactions out loud and answered questions from the user. We found out that rotational movement caused discomfort when rotating the device towards the users'body (instead of away). The Wrist rotation and lean-in posture was generally noticeable by the gyroscope.

According to the Wizard of Oz result, the sensors in phyphox are accurate for gesture detection, yet a rotational movement might cause discomfort.

The main question is still, in several scenarios, users might not notice their bad gestural habits. A hands-off approach can be effective for scenarios when users' hands are occupied with other tasks. Yet, it is not valid when users in front of screens cannot notice their bad posture, not to say using gestures to force themselves back. Our system might distinguish these two scenarios by detecting different frequency patterns when the user approaches the screen.

In addition, we found that small activities or petty actions might influence the detection results by mistaking small activities into lean-in or wrist rotation. Hence, in order to avoid those noises and petty actions, we want to build an interface to correctly detect them, and meanwhile, we want to distinguish the lean in (posture) and wrist rotation from those small activities.

DESIGN

In our experiments, we utilize angular velocities measured by the gyroscope in phyphox, due to the following aspects:

1) The gyroscope is more sensitive and more accurate than the accelerometer, especially when wrist rotation and lean-in posture are related to rotation rate;

2) Accelerations suffer from jitters and local minima, whereas angular velocities are smooth.

Accordingly, we attempted the following two scenarios:

- **Scenario 1:** The user' s hands are occupied with other tasks and he/she intends to zoom the screen using gestures. (i.e., cooking from a written recipe)
  Solution: The user applies wrist rotation gesture to zoom in the screen.

- **Scenario 2:** The user is sitting in front of his/her laptop without noticing his/her bad posture (too close to the screen).
  Solution: The system detect the lean-in posture and zoom in the screen to force the user back to origin.

Also, we might have several small activities in front of the laptop like, .

- Typing on the keyboard or not moving

- Moving items (i.e., grab a coffee cup and drink, switch the placement of your notebook to somewhere else.)

Location

To get higher accuracy, we place the sensor on the upper arm to better detect lean in posture. This is because the sensor on the upper arm can detect lean in posture more accurately than putting on the wrist or other locations.

Symbol	Activities	Action
A	Typing	Small activities
B	Moving Items	Small activities
C	Wrist Rotation	Scenario 1
D	Lean-in posture	Scenario 2

Table 1. Activities in front of the laptop

IMPLEMENTATION

Data Collection

In this section, we describe the data collected by phyphox. To build and optimize the activity recognizer and detect whether the user intentionally or accidentally approaches the screen, we present four tasks for data collection. In Task 1, a participant will be presented a video of pages (i.e., emails/docs) with various font/page sizes. Users will be told to approach or keep away from the screen as far as they can clearly read the sentence to simulate the lean in movement in front of the screen. Task 2 would be simulating the behavior when the participant picks some items beside the laptop. In Task 3, the user typing a paragraph for 30 seconds. As for Task 4, we asked the participant to perform the wrist rotation gesture. After collecting data by the gyroscope, we will distinguish these two attempts by processing the data using Fast Fourier Transformation in frequency domain. The signal data with different axes have been shown in Figure 1, 2, 3.

Spectrum Analysis

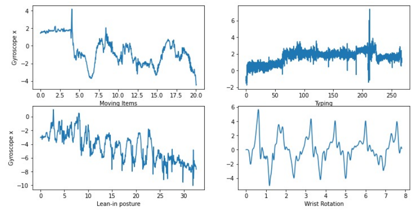

Figure 1. The gyroscope x axis with respect to four different activities

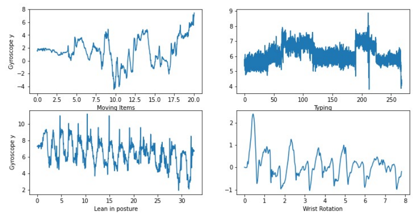

Figure 2. The gyroscope y axis with respect to four different activities

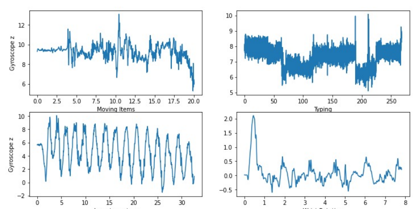

Figure 3. The gyroscope z axis with respect to four different activities

Based on the data collected by phyphox, FFT is applied to transform the time domain angular velocities to following frequency domain data

$$X(k) = \sum_{n=0}^{N-1} \omega(n)(e^{-j\frac{2\pi}{N}})^{nk}$$

where $k = 0, 1, ..., N-1$ and $\omega(n)$ is the angular velocity in each gyroscope axis. The frequency of the n-th point after transformation, denoted f_n, can be calculated as follows:

$$f_n = (n-1) \times \frac{f_s}{N}$$

where f_s is the sampling frequency and equals to 100Hz.

In order to investigate the patterns of different activities, the spectrums of the angular velocities in each gyroscope axis with respect to four usual activities are plotted in Figures 4, 5 and 6.

As can be seen from Figures 4, 5 and 6, the spectrums of angular velocities display distinct features in different activities. It can be seen that when people are typing or not moving, for gyroscope x, y, and z axis, the amplitudes are less than 0.2. Inspired by the observation, we propose our ideas by comparing the amplitudes of different axes to identify the four usual activities. Specifically, the maximum amplitude for each gyroscope axis between 0Hz and 25Hz, denoted by ω_{xm}, ω_{ym} and ω_{zm}, and then the the four usual activities are identified if and only if the following conditions are satisfied:

if $\omega_{ym} > \max(\omega_{\mathrm{xm}}, \omega_{\mathrm{zm}})$ and $\omega_{ym} > 0.5 \longrightarrow$ approach

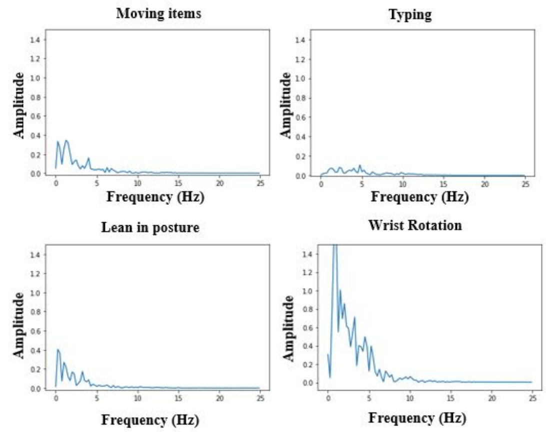

Figure 4. The spectrum of the gyroscope x axis with respect to four different activities

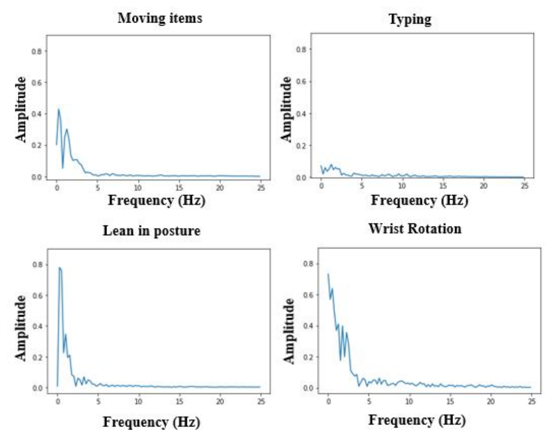

Figure 5. The spectrum of the gyroscope y axis with respect to four different activities

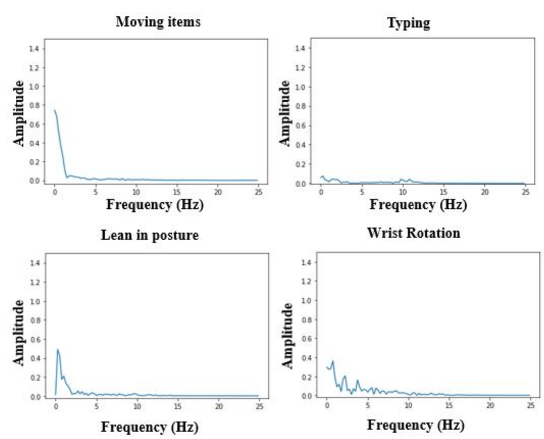

Figure 6. The spectrum of the gyroscope z axis with respect to four different activities

if $\omega_{xm} > \max(\omega_{\mathrm{ym}}, \omega_{\mathrm{zm}})$ and $\omega_{xm} > 1 \longrightarrow$ rotation

if $\omega_{xm} < 0.2\ \omega_{ym} < 0.2$ and $\omega_{zm} < 0.2 \longrightarrow$ type/not move

otherwise $\longrightarrow$ move items

To sum up, we could classify the four different activities based on these features. The flow diagram of the proposed algorithm is illustrated in Figure 7.

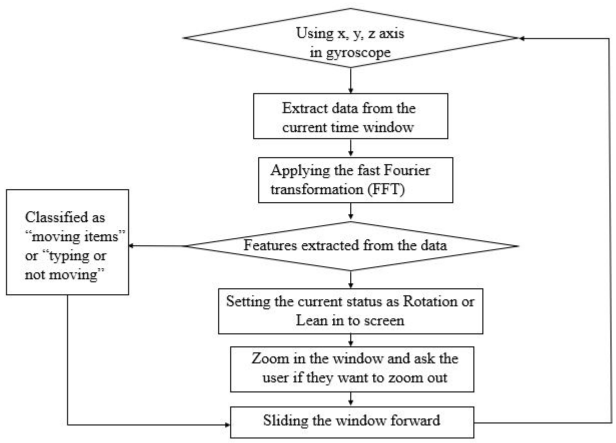

Figure 7. The flow diagram of the four activities classification

EVALUATION

In this section, extensive experiments are conducted and a thorough performance analysis is reported by the proposed method. To summarize, we collected three rounds of data, followed by three rounds of live testing. Following a short break, three more rounds of live testing were conducted.

Evaluation metrics

In order to have a clear knowledge about the performance of the status detection methods, we attempted precision (denoted P), recall (denoted R) and F1 score (denoted F) as our evaluation metrics. They are evaluated as below:

$$P = \frac{TP}{TP + FP} * 100\%$$

$$R = \frac{TP}{TP + FN} * 100\%$$

$$F = \frac{2}{\frac{1}{P} + \frac{1}{R}} * 100\%$$

,where TP is the true positive predicted status, FP is the false positive predicted status and FN is the false negative predicted status. This helped us to understand the detection not only by classification accuracy.

DISCUSSION

As shown in Table 2, **Typing** got higher f1_score because it is more stable, with amplitude generally smaller than 0.2 in the frequency domain. Particularly, although **Moving items** activity got a high recall(97.41%), which indicates a high sensitivity, yet it has a low precision (67.8%), which means that other activities (i.e., lean-in posture, and Wrist rotation) will sometimes be mis-classified as "moving items". **Lean in posture** has a relatively lower recall (77.86%). This indicates that it is relatively easily mis-classified as other activities.

Symbol	Activities	Precision	Recall	f1_score
A	Typing	95.01	98.34	96.65
B	Moving items	67.80	97.41	79.95
C	Lean-in posture	87.65	77.86	82.47
D	Wrist Rotation	89.29	91.94	90.59

Table 2. The results of activity detection

REFLECTION

Classifying different activities is still challenging. We evaluated it in only three rounds with a 'lab' environment. This means the environment was designed and meanwhile with less noises. But what if we considered testing it in a real life situation, inviting participants to do a long-term experiment? It will add more noises into our algorithm. Hence, considering more scenarios and how to increase accuracy would be the next step.

REFERENCES

1. MedlinePlus. 2017. Guide to Good Posture. (2017). `https://medlineplus.gov/guidetogoodposture.html`
2. Sungho Shin and Wonyong Sung. 2016. Dynamic Hand Gesture Recognition for Wearable Devices with Low Complexity Recurrent Neural Networks. *CoRR* abs/1608.04080 (2016). `http://arxiv.org/abs/1608.04080`
3. Vasileios Sideridis, Andrew Zacharakis, George Tzagkarakis, and Maria Papadopouli. 2019. GestureKeeper: Gesture Recognition for Controlling Devices in IoT Environments. *CoRR* abs/1903.06643 (2019). `http://arxiv.org/abs/1903.06643`