RunBuddy: A Smartphone System for Running Rhythm Monitoring

Tian Hao
Michigan State University
haotian@msu.edu

Guoliang Xing
Michigan State University
glxing@cse.msu.edu

Gang Zhou
College of William and Mary
gzhou@cs.wm.edu

ABSTRACT
As one of the most popular exercises, running is accomplished through a tight cooperation between the respiratory and locomotor systems. Research has suggested that a proper *running rhythm* – the coordination between breathing and strides – helps improve exercise efficiency and postpone fatigue. This paper presents RunBuddy – the first smartphone-based system for continuous running rhythm monitoring. RunBuddy is designed to be a convenient and unobtrusive exercise feedback system, and only utilizes commodity devices including smartphone and Bluetooth headset. A key challenge in designing RunBuddy is that the sound of breathing typically has very low intensity and is susceptible to interference. To reliably measure running rhythm, we propose a novel approach that integrates ambient sensing based on accelerometer and microphone, and a physiological model called Locomotor Respiratory Coupling (LRC), which indicates possible ratios between the stride and breathing frequencies. We evaluate RunBuddy through experiments involving 13 subjects and 39 runs. Our results show that, by leveraging the LRC model, RunBuddy correctly measures the running rhythm for indoor/outdoor running 92.7% of the time. Moreover, RunBuddy also provides detailed physiological profile of running that can help users better understand their running process and improve exercise self-efficacy.

Author Keywords
Smartphones; Exercise Monitoring; Running Rhythm

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION
In animals and humans alike, respiration is often coupled to locomotion in order to efficiently sustain endurance exercise [8]. When moving about, the locomotory system is responsible for meeting the mechanical requirements. The respiratory system supplies necessary amount of oxygen for metabolism, and removes metabolic byproducts from the circulatory system. These two systems are both critical to locomotion, and do not work independently of one another. In the case of running, the strides can be viewed as a driver signal, and the breaths are the dependent signal that varies based on the frequency of the driver signal.

The running rhythm, which characterizes the coordination between breathing and strides, varies throughout the run depending on factors such as the duration and intensity of the exercise, and training and fitness level of the runner [3]. The rhythm can also be improved over time by training: a more stable rhythm has been found in both experienced runners [3, 25] and cyclists [36, 16]. Interestingly, sound-induced stablization of such rhythm – listening to an external auditory stimulus with a proper tempo during rhythmic exercise – has been shown to result in significant reduction in oxygen consumption and improvement in exercise efficiency [12].

Unfortunately, to date, there has been no convenient and unobtrusive way of measuring running rhythm continuously. Cardiopulmonary exercise testing (CPET) is a widely adopted clinical tool to evaluate exercise capacity. It provides an analysis of respiratory gas exchange and cardiac function during exercise. However, CPET is usually limited to hospitals and clinics, due to its complicated procedure and high cost (about $20,000 /unit). Recently, several wireless CPET products have been developed. For example, Oxycon Mobile [31] integrates various lightweight sensors into a vest, which is worn by the subject during the exercise. However, designed for short-term evaluation of professional training, such devices are too bulky to wear for everyday use.

This paper presents RunBuddy – the first smartphone-based system for running rhythm monitoring. RunBuddy is designed to be a convenient and unobtrusive exercise feedback system, which provides the user continuous measurement of their running rhythm. It only utilizes commodity devices including smartphone and bluetooth headset which is often worn by runners to listen to music and make phone calls. Moreover, RunBuddy provides fine-grained and continuous measurement of the user's running rhythm, which can improve the exercise experience and help the user better understand his/her exercise self-efficacy.

Specifically, RunBuddy measures the user's running rhythm using a physiological metric called Locomotor Respiratory Coupling (LRC) [8] [1]. Figure 1 illustrates the LRC in humans with a stride to breath ratio of 2:1. The breath and stride signals are simulated using sinusoid wave and pulse signal,

[1]We use the terms "running rhythm" and "LRC ratio" interchangeably hereafter.

UbiComp '15, September 7–11, 2015, Osaka, Japan.

ACM 978-1-4503-3574-4/15/09...$15.00.
http://dx.doi.org/10.1145/2750858.2804293

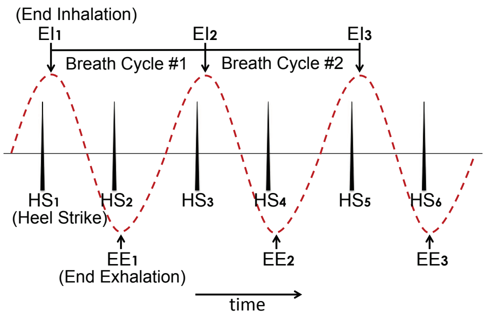

Figure 1. An example of LRC with a 2:1 stride to breath ratio.

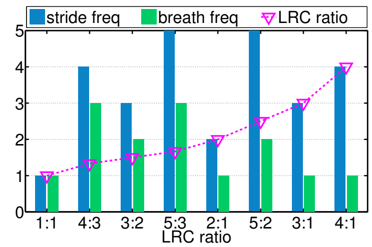

Figure 2. LRC ratios observed in humans while running [38].

respectively. The LRC ratio can be calculated as the number of strides within each breath cycle. In general, the degree of coordination is higher during running than during other activities (e.g., walking and cycling) [4]. LRC is formally defined as frequency and phase locking between two periodic systems (i.e., locomotory and respiratory system). Horses, for example, have an almost fixed LRC of 1:1 [8]. For humans, a small number of low integer coupling ratios are observed [8], which are shown in Figure 2.

In order to be convenient and unobtrusive, RunBuddy analyzes acoustic samples from the bluetooth headset worn by the user to detect breathing. However, the sound of breathing typically has very low intensity and is susceptible to various interferences from the environment (e.g., music in the gym, sound of wind or traffic). Moreover, the breathing pattern can vary significantly among different individuals during running, in terms of the loudness, acoustic frequency range. Lastly, in order to preserve users' privacy and provide real-time feedback, the system must process the data on the fly, and should not store or transmit raw sound samples. Therefore, the breath detection algorithms must be lightweight while maintaining satisfactory measurement accuracy.

To address above challenges, we propose a novel approach to the design of RunBuddy, which integrates ambient sensing based on accelerometer and microphone, and a physiological model LRC to improve the accuracy of running rhythm monitoring. LRC is a quantitative measure of the running rhythm: the coordination of locomotory and respiratory systems. According to the LRC theory, there exists a small number of low-integer coupling ratios (e.g., 2:1) between the stride and breathing frequencies [8]. RunBuddy leverages the LRC theory to calibrate the running rhythm measurement. Specifically, we develop a lightweight signal processing pipeline to detect breathing from the acoustic samples of bluetooth microphone. Due to the low sound intensity of breathing and the impact of environment noises, the acoustic breath detection results may be highly inaccurate. To address this issue, we detect strides through smartphone accelerometer, and correlate strides, possible LRC ratios, and preliminary acoustic breath results together to calibrate the running rhythm measurement. The LRC has been found in a number of rhythmic exercises including cycling [33], rowing [23], running [8] and walking [21]. Therefore, the approach of RunBuddy can potentially be applied to monitor various rhythmic exercises.

We believe a personal, convenient running rhythm monitoring system like RunBuddy holds great potentials to improve users' exercise experience and self-efficacy. Many beginning runners and people who are usually sedentary lose interest or even get frustrated quickly because they can not catch their breath during running, due to exercise hyperpnea [1]. By monitoring their running rhythm in real-time, the system could potentially guide them to achieve a more comfortable rhythm by playing music with proper tempos (e.g., by adjusting the workout soundtrack) [2, 15, 5]. Moreover, the system can also provide an analysis of users' physiological profile, such as the stability of LRC ratios during running, which is a good indicator of fitness level [25].

RELATED WORK

Recently, cardiopulmonary exercise testing (CPET) has become an important clinical tool to evaluate exercise capacity. It also has been increasingly used in exercise training. It provides a fine-grained, breath-by-breath analysis of respiratory gas exchange and cardiac function. However, CPET is usually limited to hospitals and clinics, due to its complicated procedure and high costs (about $20,000 /unit). Several wireless commercial CPET products are available on the market. For example, Oxycon Mobile [31] integrates various sensors along with batteries into a vest, which is worn by the subject during the test. However, it is mainly designed to evaluate the training of professional athletes, and not suitable for everyday use due to the bulky size and high cost.

Several efforts have been made to develop smartphone-based exercise monitoring systems. For example, a system for monitoring the workload (a variation of heart rate) is proposed in [37]. It leverages machine learning techniques to predict heart rate variation from the acceleration and speed during walking. In [41], the authors present a smartphone-based system to estimate caloric expenditure of bicyclists. They improve the accuracy by considering multiple inputs such as GPS signal and map information. SpiroSmart, a smartphone-based spirometer using the built-in microphone is proposed in [17]. To measure air flow and volume of a single breath, the user has to blow air to the phone's microphone in close proximity in a quiet environment. Therefore, it is not suitable for continuous monitoring during exercise.

A few methods have been proposed to detect strides based on acceleration. In [7], common stride detection algorithms are evaluated based on large data sets from smartphone sensors. A gait analyzer is proposed in [13] that can identify gaits such as walking and running solely using acceleration. Reliable stride detection has also been used in many other applications, such as systems for indoor positioning [20] and energy expenditure monitoring [18].

In addition to the exercise monitoring systems mentioned above, several systems have been developed to keep users motivated and engaged during exercise, by providing real-time information about the user's performance. For example, by continuously monitoring the user's heart rate or activity level, the smartphone may provide music suggestions or feedbacks that guide users' steps [10, 5] or help users achieve their target heart rate [29, 30] during exercise.

Compared with the existing exercise monitoring systems, RunBuddy differentiates itself as the first smartphone-based system that continually monitors user's running rhythm. It is designed to be a convenient and low-lost solution for everyday use. As shown in Figure 3, RunBuddy only utilizes commodity devices including smartphone and bluetooth headset, without relying on any custom hardware. In addition, RunBuddy is also easy-to-use and truly unobtrusive. Users only need to wear the bluetooth headset and carry the smartphone during running.

SYSTEM REQUIREMENTS AND CHALLENGES

RunBuddy is designed to be a convenient and unobtrusive exercise feedback system, which provides the user continuous monitoring of their running rhythm. Specifically, RunBuddy measures the running rhythm in terms of LRC ratio in a time window (e.g., 10 seconds) continuously during running. Based on the measured LRC and stride frequency, RunBuddy can also estimate real-time breathing frequency (number of breaths per minute). However, we believe LRC is a more desirable metric for runners since it quantifies the coordination between breathing and stride. Research has shown that LRC is a good indicator of fitness level [24].

RunBuddy is designed to meet the following requirements: (1) Since RunBuddy operates during running, it needs to be unobtrusive. The user should not feel any kind of uncomfort or added burden when using the system. (2) RunBuddy needs to provide fine-grained and continuous measurement of the user's running rhythm. (3) Such measurement needs to be robust and accurate across different users and exercise environments. (4) The system needs to strictly protect the users' privacy. Since the system relies on sound recognition for breath detection, users might have concerns about possible privacy breach, such as their conversation during running being recorded. Therefore, the system should process data on the fly, and only keep the exercise-related data (e.g., breathing/stride frequency over time). Moreover, online processing is also required by applications that need to provide real-time feedback (e.g., music suggestions) to the user.

To meet these requirements, three major challenges need to be addressed in developing RunBuddy. First, in order to make RunBuddy convenient and unobtrusive, the system samples and analyzes acoustic signals from the bluetooth headset worn by the user to detect breath. However, the sound of breathing has low intensity, making it challenging to capture. Second, the sound of breathing can be different among different individuals, in terms of its loudness and frequency range. Moreover, noises from the exercise environment such as music in the gym may also have a major impact on the accuracy of breath detection. Lastly, RunBuddy must adopt extremely lightweight signal processing algorithms as it needs to process high-rate sensor stream (e.g., 16 *KHz* sound and 100 *Hz* acceleration) in real-time.

SYSTEM OVERVIEW AND APPLICATIONS

RunBuddy aims to continuously monitor use's running rhythm. It requires no extra hardware except a bluetooth headset and a smartphone. Figure 4 shows the overview of RunBuddy, which consists of four major components: *breath detection*, *training*, *stride detection*, and *LRC-based correlation*.

The key novelty of RunBuddy lies in the integration of ambient sensing and physiological models to improve the accuracy of running rhythm detection. Specifically, RunBuddy detects breaths based on its unique sound features. Due to the low sound intensity of breath and the impact of environmental noises (e.g., music in gym and wind outdoors), the acoustic breath detection result may be highly inaccurate, leading to poor accuracy in LRC ratio measurement. Leveraging the LRC theory, RunBuddy addresses this challenge by correlating the detected breath with the more reliable and accurate stride detection result to find the most likely LRC ratio.

The breath detection is based on sound, which is continuously collected through the bluetooth headset worn by the user during running. First, the raw acoustic signal is processed by a low-pass filter, where high-frequency noises are filtered out. Second, RunBuddy frames the filtered signal and extracts MFCC features from each frame. It then uses the first 7 features (12 variables in total) for *breath detection*. In order to improve the accuracy of breath detection, RunBuddy requires a straightforward one-time training to capture the sound features of the breath from a particular user. To complete the training, the user only needs to breathe several times. The training sound captured by the bluetooth headset is fed into the same low-pass filter used for breath detection. Next, the energy spectrum of the filtered signal is calculated using *FFT*. Then, the energy spectrum over time is fed to *breath feature extraction*, where RunBuddy detects frames that contain the user's breath based the energy, and extracts their MFCC features. These extracted features are considered as the *breath signature*, which represents the unique sound features of the user's breath. At runtime, RunBuddy is able to detect a possible breath by calculating the similarity between extracted MFCC features and the user's breath signature. However, the breath detection result at this stage may be inaccurate due to the impact of environmental noise. As a key novelty, RunBuddy calibrates such preliminary breath detection results by correlating with strides based on possible LRC ratios.

To detect strides, RunBuddy first captures motion through sampling the built-in accelerometer. The collected acceleration is first processed to obtain the vertical acceleration (with the same direction of gravity), by projecting the raw data onto the global coordination system. Then the vertical acceleration is taken as the input of *stride detection*, where several filters

Figure 3. A typical setting of RunBuddy. The user is required to wear a bluetooth headset and carry a smartphone while running.

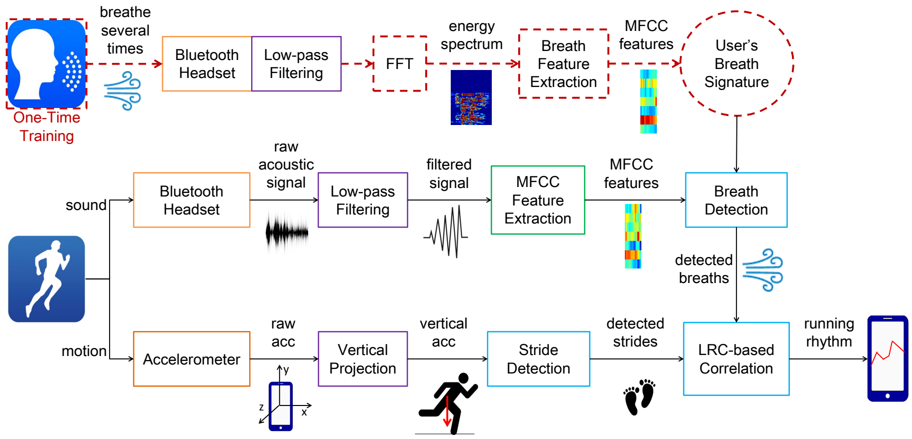

Figure 4. Overview of the RunBuddy system. RunBuddy requires a one-time training, and comprises 3 major components: breath detection, stride detection and LRC-based calibration.

are employed to enhance the performance of the detection [2]. In the *LRC-based correlation*, RunBuddy utilizes the detected strides to calibrate the breath detection results. Based on the theory of LRC, there exists a small number of low-integer coupling ratios (e.g., 2:1) between the stride frequency and the breathing frequency. The coupling is especially strong during running, due to the more frequent leg movement, and the increased intensity. RunBuddy is able to reliably measure the running rhythm, by correlating the detected strides and breathing based on possible LRC ratios for humans.

As running rhythm is essential in reflecting the user's physical state, RunBuddy has great potentials to benefit users' exercise experience in many scenarios. First, as suggested by previous study [25], the ability of maintaining a stable running rhythm can potentially be used to infer the runner's fitness level. For example, compared to non-runners, regular runners are able to maintain a more stable LRC ratio during running [3, 25]. Second, based on the running rhythm, the system could potentially provide real-time and personalized feedbacks such as music suggestions to enhance users' exercise experience. For instance, many new runners lose interest or even get frustrated because they can not catch their breath during running. By monitoring their running rhythm in real-time, the system could help them achieve a more comfortable rhythm by playing music with proper tempos (e.g., by adjusting the workout soundtrack) [2, 15, 5]. Once feeling comfortable, new runners will be more willing to engage in running.

SYSTEM DESIGN

In this section, we first describe the feature extraction used for breath detection. Next, we explain the training process and how we utilize training data to detect breath. Then, our approach to recognizing stride from acceleration is described. Lastly, we present the LRC-based correlation, where we leverage the correlation of detected strides and breathing to improve the accuracy of running rhythm measurement.

[2]Some exercise monitoring applications may already include stride detection algorithms. In such a case, RunBuddy may directly use the real-time output of these algorithms.

Acoustic Feature Extraction

During running, the system continuously collects acoustic signal through the microphone at a sampling rate of 16 *kHz*. Prior to feature extraction, we use a low-pass Butterworth filter of order 2 to suppress high frequency noise in the raw data. Since the typical sound frequency of human breath falls in the range of 500 to $3,500$ *Hz*, we set the cutoff frequency of the filter at $3,500$ *Hz*. After pre-processing, the acoustic signal is framed using a moving window. Each *frame* includes 40 *ms* of acoustic signal and has 30 *ms* overlap with nearby frames. For each frame, we calculate the features of mel-frequency cepstral coefficients (MFCCs) within the frequency range of 100 to $5,000$ *Hz*. The energies of the first 7 MFCC channels are used as the feature vector for breath detection.

Training for Breath Detection

A one-time training process is necessary for sound-based breath detection. This is primarily because the sound of breathing is different among individuals, in terms of its duration, strength and frequency range, resulting in diverse acoustic profiles. Therefore, it is necessary to customize the breath detection for each user based on training. The training process is designed to be simple and easy to follow. Before the first time use of the system, the user is required to breath several times in a relatively quiet environment. This ensures that the system can capture the sound of breathing with as little noise as possible. Moreover, the system can automatically detect the sound of breathing. This spares the user's effort of manually labeling the breath events from raw data. The detected breath events are then used to generate the user's breath signature represented by MFCC features.

Breath Detection

The goal of breath detection is to recognize acoustic frames that contain breath. Basically, it estimates the likelihood of a frame containing breath, by calculating the similarity between its feature vector and the training data. Specifically,

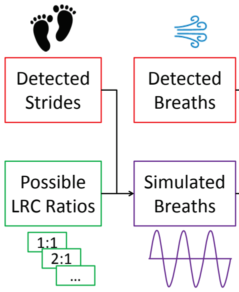

Figure 5. The process of LRC-based correlation. The detected breaths and strides are taken as inputs.

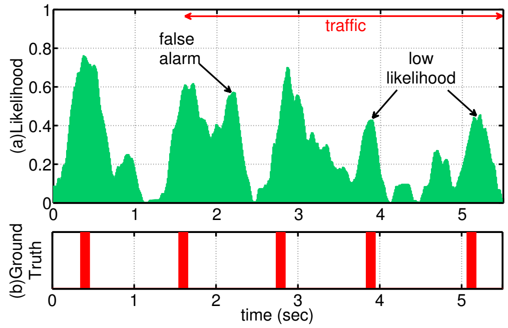

Figure 6. An example of inaccurate breath detection result under the noise caused by passing traffic.

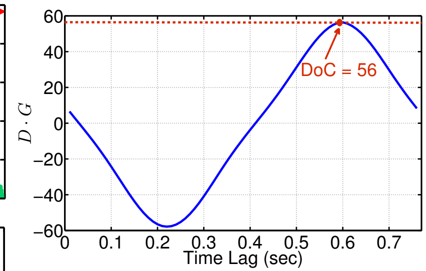

Figure 7. An example of computing Degree of Correlation (DoC) for 2:1 LRC ratio (using 10-second data). The arrow points to DoC, the maximum value of $D \cdot G$.

let $bv_1, bv_2...bv_n$ be the feature vectors of n breath frames extracted from the training data. First, we calculate the mean vector $\overline{bv}$ of all the breath frames as follows,

$$\overline{bv} = \frac{\sum_{i=1}^{n} bv_i}{n} \tag{1}$$

Then, we calculate the cosine similarity $S(bv_i, \overline{bv})$ between bv_i and $\overline{bv}$. The calculation is given below.

$$S(bv_i, \overline{bv}) = \frac{bv_i \cdot \overline{bv}}{\| bv_i \| \| \overline{bv} \|} \tag{2}$$

The resulting cosine similarity ranges from -1 (the minimum similarity) to 1 (the maximum similarity). The cosine similarity has been widely used in information retrieval to measure the similarity between two texts. We use cosine similarity to measure the similarity between two vectors. Since cosine similarity is independent of magnitudes of acoustic signals, it is effective at capturing the characteristic of MFCC features and robust against environmental noises. Lastly, we use the minimum value of the similarities as the threshold T. Given $\overline{bv}$ and threshold T from the training data, we define the likelihood that a frame with feature vector bv^*, which is captured at runtime, contains breath as,

$$L(bv^*, \overline{bv}, T) = \begin{cases} \frac{S(bv^*, \overline{bv}) - T}{1-T} & \text{if } S(bv^*, \overline{bv}) > 0 \\ 0 & \text{if } S(bv^*, \overline{bv}) \leq 0 \end{cases}$$

where $S(bv^*, \overline{bv})$ is the cosine similarity between bv^* and $\overline{bv}$.

Stride Detection

Stride detection aims to recognize strides from collected acceleration data. The stride detection method adopted by RunBuddy is based on searching for peaks in the change of acceleration during running. Although this method is similar to several existing methods [22, 7], we include the details here for completeness. Specifically, RunBuddy first projects the sampled acceleration data onto global coordination system. Then it detects strides by searching for peaks in the vertical acceleration (the gravity direction).

By projecting acceleration onto a global coordinate system, we don't have to consider the phone's orientation change in the following processing. Moreover, the vertical acceleration is more sensitive to the shock caused by each step. Prior to projection, RunBuddy estimates the direction of the gravity by averaging the acceleration along each axis within a short time (e.g., 5 seconds) [26]. Then, it projects the acceleration data onto global coordinate system. Let $\vec{a}$ be the acceleration reading and $\vec{g}$ the estimated gravity. The vector representing vertical acceleration $\vec{v}$ can be calculated as follows,

$$\vec{v} = \vec{a} \cdot \frac{\vec{g}}{\| \vec{g} \|} \tag{3}$$

We can also get the horizontal acceleration $\vec{h} = \vec{a} - \vec{v}$. Next, we feed $\| \vec{v} \|$ into a series of filters to detect stride which contains one left step and one right step. Since the phone is attached to the user during running, our stride detection method is based the assumption that each major peak in the $\| \vec{v} \|$ is caused by one step. Therefore, detecting stride in this scenario is essentially a peak detection problem. In order to improve the accuracy, prior to the peak detection, we process $\| \vec{v} \|$ with several filters included in the Pan-Tompkins algorithm [32, 34] to enhance the peak in the signal.

LRC-based Correlation

The basic idea behind LRC-based correlation is to improve the accuracy of running rhythm measurement by estimating the most likely LRC ratio based on the correlation between detected stride and breathing. Since the motion sensor (accelerometer) is attached to user's body, the result of stride detection suffers far less from the environmental noise than that of breath detection. Specifically, there are two major factors that affect the performance of acoustic breath detection. One is the environmental noise, such as music in the gym, sound of wind or traffic, and etc. Another factor is the difficulty in detecting the breath, due to its low acoustic amplitude. Therefore, the estimated breath described in Breath Detection can not be directly used to calculate running rhythm.

Figure 6 shows an example of the breath detection result under the noise caused by passing traffic. The result is obtained using the algorithm described in Section Breath Detection. We can see that noise may cause both false alarms and low detection likelihood in breath detection. The false alarm is usually caused by noises that have similar MFCC features as the user's breath signature. For example, at 2.2 seconds, the noise causes a false alarm with high likelihood, where there exists no actual breath event. Noise can also interfere with the sound of actual breath. Due to the interference, the MFCC features of the actual breath may largely deviate from

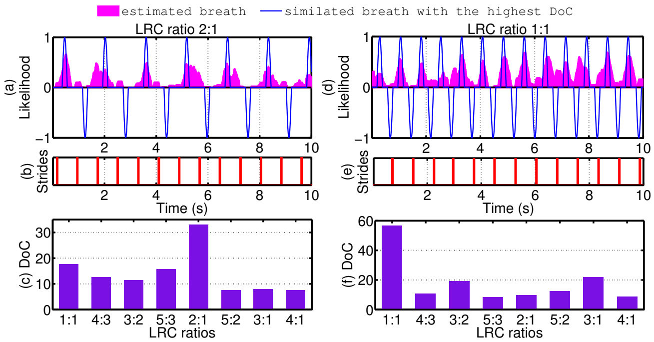

Figure 8. Two examples of LRC-based calibration. The sine waves plotted in (a) and (d) are the simulated breath signals that have the highest correlation with detected breaths. (b) and (e) indicate the detected strides. (c) and (f) indicate *DoC*s associated with each LRC ratio.

the user's breath signature, resulting in low detection likelihood. For example, the likelihood of the 4th and 5th breaths is only about half of that of the first three breaths.

According to the LRC theory, there exist several coupling ratios between the rhythm of stride and that of breath. The fundamental reason for the coupling effect is because breathing is responsible for supplying oxygen for metabolism, which keeps generating energy for locomotion. The design of LRC-based correlation in RunBuddy is based on two key observations. First, during running, breathing and stride frequencies are relatively stable within a short period of time (e.g., 10 seconds). Second, there only exists a limited number of LRC ratios in humans, and the coupling effect is stronger while running. This is because, as the intensity of the exercise gets higher (e.g., running), the rate of metabolism increases, resulting in a more obvious coupling effect. Therefore, RunBuddy is able to infer the accurate running rhythm using the most likely LRC ratio. To find the most likely LRC ratio, RunBuddy first use detected strides to generates breath signal, referred to as *simulated breath*, for each possible LRC ratio. RunBuddy then computes the correlation between the estimated breath and the simulated breaths. The LRC ratio that gives the highest degree of correlation is hence the mostly likely LRC ratio that represents the running rhythm.

Figure 5 shows the overview of the correlation process. First, we generate simulated breath signals using the detected strides for all 8 possible LRC ratios shown in Figure 2. Specifically, let T_{stride} be the duration of the stride cycle, and R_{LRC} be the ratio of stride frequency to breath frequency. The duration of simulated breath cycle T_{breath} can be calculated as follows,

$$T_{breath} = T_{stride} \times R_{LRC}; \tag{4}$$

We use sinusoid signal to represent the breath. The width of the sinusoid signal is fixed (e.g., 20 *ms*), regardless of the selected LRC ratio.

After generating the breath signals associated with different LRC ratios, we calculate the degree of correlation (DoC) between the simulated breaths and the detected breaths. Suppose $D = d_1, d_2, ..., d_n$ represents the estimated likelihood of breath for each frame within a short time window, $G(R) = g_1, g_2, ..., g_m$ is the corresponding breath signal simulated

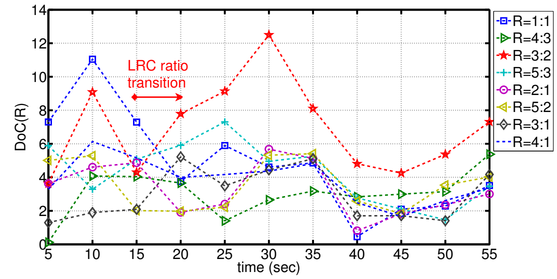

Figure 9. An example of degree of correlation (*DoC*) calculation for all possible LRC ratios where there exists a LRC ratio transition from 15 to 20 seconds.

based on the strides and a particular LRC ratio R. The degree of correlation for LRC ratio R is given by,

$$\begin{aligned} DoC(R) &= \max_{s \in [1, T_{breath}]} \{D \cdot G(R, s)\} \\ &= \max\{\sum_{i=1}^{n} d_i \times g_{i+s}\} \end{aligned} \tag{5}$$

where s denotes the time lag of the simulated breath signal $G(R)$, and T_{breath} is the length of the breath cycle associated with $G(R)$. The basic idea of the degree of correlation is based on cross-correlation, where the maximum result of the sliding dot product reflects the similarity of the detected breaths and simulated breaths. Figure 7 shows an example of computing $DoC(R_{2:1})$ when LRC ratio is 2:1, and $G(R_{2:1})$ is generated based on the detected strides and 2:1 LRC ratio. We can see that the value of $D \cdot G(R_{2:1})$ varies as the time lag changes. And the maximum value is chosen as the $DoC(R_{2:1})$. In order to determine the most likely LRC ratio, we generate simulated breath signal for each LRC ratio (i.e., $G(R_{1:1})$, $G(R_{2:1})$,..., $G(R_{4:1})$), and calculate their corresponding *DoC*s (i.e., $DoC(R_{1:1})$, $DoC(R_{2:1})$,..., $DoC(R_{4:1})$). The LRC ratio that gives the highest *DoC* is chosen as the most likely LRC ratio representing the current running rhythm. Moreover, as the LRC ratio defines the frequency ratio between stride and breath, we can obtain a more accurate breathing frequency than that derived from the estimated breath, using the most likely LRC ratio and the stride frequency (Equation 4).

In our implementation, we use a 10-second moving window with 50% overlap for the calibration. For example, Figure 8 shows the LRC-based correlation of two 10-second windows. The blue sine waves shown in Figure 8(a) and (d) indicate the simulated breaths associated with the most likely LRC ratio. We can observe that it gives the highest *DoC* among all LRC ratios (shown in Figure 8(a) and (d)). Moreover, we can see that it is difficult to accurately identify breaths by solely relying on the estimated breath likelihoods. Therefore, by cross-correlating the detected breaths and simulated breaths of different LRC ratios, we can effectively find the most likely LRC ratio representing the running rhythm.

Note that the LRC-based calibration needs to be repeated using a 10-second moving window because runners may switch to a different LRC ratio during running. According to our experimental result, the frequency of LRC ratio change is highly dependent on each individual. In general, non-runners switch more frequently than regular runners [11, 4]. Figure 9 shows

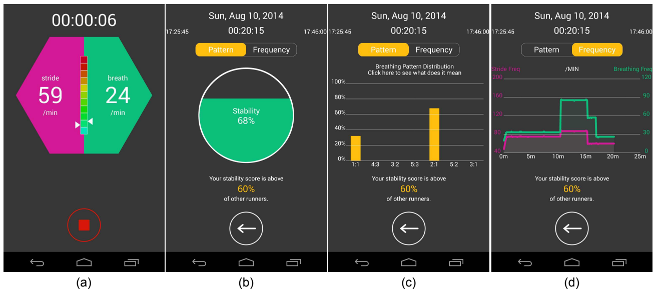

Figure 10. The user interface of the app. (a) The screen showing the real-time breathing and stride frequency during running. (b) The screen showing the estimated fitness level according to the stability of running rhythm. (c) and (d) are the screens showing the details about the running rhythm distribution and the runner's breathing and stride frequency.

an example of calculated *DoC* of all possible LRC ratios over time, where the runner switched from LRC ratio 1:1 to 3:2 during 15 to 20 seconds. As mentioned above, RunBuddy calibrates the breath using a moving window of 10 seconds and 50% overlap, resulting in a *DoC* calculation every 5 seconds. Therefore, RunBuddy can effectively detect the LRC ratio switch during running.

IMPLEMENTATION

RunBuddy is implemented on Android 4.3. The size of the application file is about 1 MB. While running, RunBuddy requires about 20 MB RAM allocation. The displaying and processing functions are implemented in separate threads, in order to ensure timely sampling and processing as well as quick response to user's interaction. RunBuddy continuously samples the microphone at 16 *kHz* and accelerometer at 100 *Hz*. The acoustic samples are framed into 100 *ms* frames before detection. The detection results of breathing and stride within a 10-second moving window with 50% overlap to measure LRC ratio, resulting in a measurement every 5 seconds.

We have released RunBuddy as a mobile app on the Google Play Store [3]. The screen shots are shown in Figure 10. The app is easy to use. After performing a one-time training, the user only needs to put on the bluetooth microphone and start the monitoring before running. During running, users are free to carry the phone in their preferable way (e.g., inside pocket or armband). As the app prevents the CPU from sleeping, the user is able to turn off the screen to save battery. It allows users to check their instant breathing and stride frequencies during running by simply turn on the screen. After running, the user needs to stop the monitoring to see an overall result reflecting the fitness level calculated based on the stability of running rhythm. For each run, the app also provides detailed information including the distribution of running rhythm, as well as breathing and stride frequencies over time.

[3]Our mobile app released on Google Play Store is branded as "iBreath". And the Google Play link is `https://play.google.com/store/apps/details?id=com.tian.ibreath&hl=en`. A video introducing iBreath is at `http://youtu.be/hZMZqt4Pae4`. iBreath was awarded the Best Mobile App Award, Third Place, at MobiCom Mobile App Competition 2014. While iBreath has the identical design with RunBuddy, it presents the user breathing frequency (number of breaths per minute) that is derived from the measured LRC.

EVALUATION

Experimental Settings

In order to evaluate RunBuddy, we recruited 13 subjects and collected data from 39 runs in total (526.1 minutes). Our study along with its data collection procedure was approved by the Institutional Review Boards (IRB) at Michigan State University. All the subjects voluntarily agreed to help with the data collection, and signed a consent form. In order to collect data, each subject used a smartphone (Google Nexus 4 [28]) and a Bluetooth headset during running. We used three different models (Jabra Wave [14], Voyager Legend [39] and Voyager Pro HD [40]) in our experiments to investigate the impact of different Bluetooth headsets. RunBuddy performs similarly with different Bluetooth headsets. During data collection, the acoustic data used for breath detection is collected through the bluetooth headset, and the acceleration data used for stride detection is collected using the phone's built-in accelerometer. The sampling frequency was empirically set to 16 *kHz* for the sound, and 100 *Hz* for the acceleration. The collected data was stored in the phone for off-line analysis.

The ground truth for breath detection is collected by either of the following two methods. In the first method (used for 34/39 runs), an in-line mic is attached under the subject's nose during running. Since the in-line mic is very close to the subjects nose and mouth, the recorded breathing sound is clear enough to be automatically labeled off-line through simple threshold-based peak detection. In the second method (used for 5/39 runs), we manually labeled the breath event if the subject felt uncomfortable to wear the in-line mic under nose, or the in-line mic accidentally fell off during running. Specifically, the labeling is done by listening to the audio recorded by the Bluetooth headset and finding the time of the beginning of each breath event. It only involves identifying and counting the instances of breathing within each detection window (10 seconds) for running rhythm measurement. The recorded audio clips where the breathing sound is not clear enough for manual labeling were discarded and not used in the evaluation (2 of such clips were discarded). To collect the ground truth for stride detection, an extra smartphone was attached to the user's lower leg using an armband.

Table 1 shows the general information about the subjects who participated in our experiments. In order to assess their capability of aerobic exercise, we asked the subjects to fill out a questionnaire before the experiment, which includes several questions such as "How many days per week do you exercise?". Based on their self-report information, we divided the subjects into three categories as follows: **Non-runner**, subjects who rarely take any forms of exercise; **Occasional runner**, subjects who occasionally take short aerobic exercise, but do not have a regular routine; **Regular runner**, subjects who usually take more than 30 minutes of exercise for at least twice a week. To assess the subjects' body shape, we also calculated their body mass index (BMI) based on their weight and height. According to the standard specified by the World Health Organization (WHO) [6], the body shape can be classified as follows: **Underweight** ($BMI \leq 18.5$), **Normal weight** ($18.5 < BMI \leq 24.9$), **Overweight** ($25 < BMI \leq 29.9$) and **Obesity** ($BMI \geq 30$).

category	subject	sex	# runs (in/out)	avg. duration	BMI
Non-runner	1	F	1 (1/0)	6.1 min	16.9
Non-runner	2	F	1 (1/0)	5.8 min	17.9
Non-runner	3	M	2 (1/1)	5.1 min	27.5
Non-runner	4	F	3 (2/1)	5.2 min	18.3
Non-runner	5	M	1 (1/0)	8.3 min	28.4
Occasional	6	M	7 (4/3)	6.3 min	25.4
Occasional	7	M	2 (2/0)	16.1 min	27.2
Occasional	8	M	3 (3/0)	12.1 min	24.4
Occasional	9	M	3 (3/0)	14.5 min	22.1
Occasional	10	F	3 (3/0)	13.8 min	20.5
Occasional	11	M	4 (2/2)	12.9 min	23.1
Occasional	12	F	3 (3/0)	15.6 min	19.1
Regular	13	M	6 (3/3)	30.7 min	22.4

Table 1. General information about the subjects.

Prior to the experiment, each subject was instructed to run as they normally do. We did not set a specific duration for the run, and the subjects were asked to keep running as long as they felt comfortable. Note that the subject is free to choose where to place the smartphone during running. In our experiments, the ways of carrying the phone include holding in one hand, placing inside the pants' pocket, and attaching to the arm using sport armband. Our result shows that, regardless of the phone placement, RunBuddy can reliably detect strides with an accuracy of over 99%.

Metrics

We processed the collected data using the methods described in System Design. The metric we use to evaluate the performance of RunBuddy is the *percentage of correct measurements* (PCM). PCM = X% means RunBuddy correctly provides the LRC ratio measurement X% of the time during the run. For instance, for a run about 8 minutes consisting of 100 measurement windows, 95% PCM indicates 95 measurements are correct. The reason that we use PCM instead of accuracy or error rate as the metric is because the measurement of running rhythm in terms of LRC ratio is represented by discrete values. Also, we note that in most false measurements, RunBuddy produced a LRC ratio close to the ground truth (e.g., 4:3 while the correct ratio is 3:2).

Overall Performance

In this section, we evaluate RunBuddy when the users run in typical gyms and outdoors. Figure 11(a) shows the average PMC for each subject based on data collected during 29 indoor runs. The overall PMC for the indoor runs is 93.3%, with the lowest PMC of 87.3% for subject 1, and the highest PMC of 95.2% for subject 6. We can make two interesting observations from the indoor result. First, RunBuddy tends to yield lower PMC for data collected from female subjects (subject 1, 2, 4 10 and 12). This is primarily because the sound of breathing of females typically have lower intensity than that of males, making it more difficult to be recognized. Another observation is that the accuracy for all five subjects who are non-runners (subject 1, 2, 3, 4 and 5) are below the average PMC (93.3%). The major reason is that subjects of lower fitness level usually take shorter and shallower breath during running. However, it is widely recommended by professional runners and running couches to take long and deep breath during running, which helps to obtain enough oxygen and prevent muscle and lung fatigue [9].

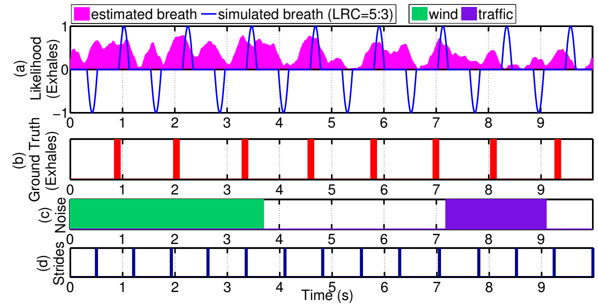

Figure 12. A 10-second example based on real data collected outdoors, demonstrating the impact of environment noises such as wind and traffic. (a) shows the estimated breath and simulated breath with the highest *DoC*. (b) is the ground truth for breath. (c) shows the time and duration of environment noises. (d) is the detection result of stride.

Figure 11(b) shows the PMC for each subject based on data collected during 10 outdoor runs. One key observation is that, for all subjects except subject 6, the average PMC of outdoor runs is higher than that of indoor runs, due to the impact of environment noises like wind and traffic. As an exception, all the 3 outdoor runs of subject 6 were collected in late evening in a quite neighborhood.

Comparison With Non-LRC Method

In order to evaluate the performance benefit of using LRC model, we compare our LRC-based approach with a simple non-LRC method, which does not involve LRC-based correlation. Basically, the non-LRC method detects the breathing instances solely based on the acoustic features (described in Section Breath Detection). As demonstrated in Figure 6, due to the interference from various noises, it is very difficult to reliably extract breath instances from the acoustic samples from microphone. Therefore, in the non-LRC method, we also consider the periodicity of breathing in a short time (e.g., 10 seconds). Specifically, for each detection window (10 seconds), we identify peaks from the breathing likelihood result. We then find the most likely breathing frequency for this particular detection window by correlating the identified peaks with all possible breathing frequencies from the minimum to the maximum frequencies in humans. Lastly, the LRC ratio is calculated using the most likely breathing frequency along with the detected stride frequency.

The performance of non-LRC method is shown in Figure 11. We can see that, for each subject and type of environment, the PMC of non-LRC method is substantially lower. This is primarily because, instead of considering the LRC model, the non-LRC method computes breathing frequency solely based on the acoustic features, which are susceptible to interferences such as sounds of wind and traffic. We can also observe that, the performance of non-LRC method is highly dependant on the noise level. A higher PMC is achieved when the data is collected in quiet gym (67.3%) or outdoor area (67.5%), compared with the gym with music (62.6%) and outdoors with wind and traffic (53.6%).

Impact of Environmental Noises

Next, we investigate the performance of RunBuddy in different environments. Figure 11(c) shows the PMC in 4 different scenarios based on all the data (39 runs). We can see that

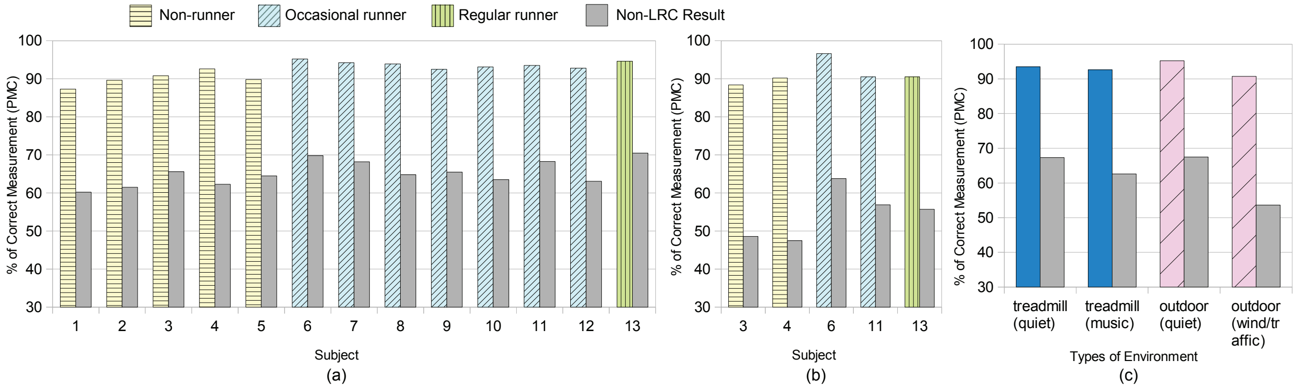

Figure 11. Overall performance of our approach and comparison with a simple non-LRC method. (a) shows the PCM for each subject based on data collected on treadmill in typical gyms. (b) shows the PCM for each subject based on data collected during outdoor runs. (c) shows the overall PCM in 4 typical scenarios: (1) running on treadmill in a relatively quiet gym, (2) running on treadmill in a gym with music and nearby runners, (3) running outdoors on a relatively quite route, (4) running outdoors by the street with traffic.

compared to outdoor running (avg. PMC 91.4%), the PMC is higher when running on indoor treadmill (avg. PMC 93.3%). This discrepancy is largely due to the outdoor noises, such as wind and traffic. However, we note that the highest PMC (95.2%) is achieved when running outdoors in a quite neighborhood, even 1.7% higher than running in a quiet gym. This is largely due to the fact that operating treadmills still produce substantial noise. In a typical public gym setting, the PMC drops to 92.6%, due to the raised noise level caused by nearby runners and ambient music. When running outside, wind is the most common environmental noise, especially when running against the wind. As the noise of wind has overlapping frequency with the sound of breath, it can largely affect the accuracy of breath detection.

However, in most cases, RunBuddy can still manage to accurately estimate the running rhythm by leveraging the LRC-based correlation, even in the presence of wind and traffic. Figure 12 shows an example of how RunBuddy mitigates the impact of environmental noises. Due to the interference of the wind, the breath detection algorithm yields continuous high likelihood of breath during the first 3.7 seconds. During 7.2 to 9.1 seconds, a vehicle was passing by the subject. We can see that, in detecting the 7th and 8th breaths, the traffic noise lowers the likelihood of breath by suppressing the sound of breath. An interesting observation is that, even though some of the detected breath do not match the breathes in ground truth, the number of detected breathes is still accurate, due to the fact that the LRC ratio remains constant and the strides are detected accurately. Therefore, by leveraging the LRC model, RunBuddy can largely mitigate the impact of various outdoor noises, including wind and nearby traffic.

Runners with Different Fitness Levels

RunBuddy is designed to help improve the running experience of both regular runner and non-runners alike. As regular runners typically breathe differently from non-runners, in terms of depth and duration, it is important to evaluate the performance of RunBuddy for runners at different fitness levels. According to the result of our initial experiment, the PMC of LRC ratio measurement increases as the fitness level of the subject gets higher. Specifically, the average PMC of indoor running for non-runners, occasional runners and regular runner are 90%, 93.6% and 94.6%, respectively. This is primarily due to the fact that regular runners typically breathe deeper and longer, resulting in more distinctive sound features that are easier for the system to capture.

In addition to the depth and duration, the breathing of regular runners and non-runners also differs in its coupling with the strides. Figure 13 shows the detailed detection results of a 17-min run from a regular runner, and an 8-min run from a non-runner. In Figure 13(a), we can see that the subject gradually raised his breathing and stride frequencies in the first minute. Then, he maintained a steady stride frequency till 14:30 minutes. After that, he gradually reduced the speed and switched from running to fast walking. One interesting observation is that the subject doubled his breathing frequency for about 1 minute (around 14 minutes) at the end of the run. It is mainly because he raised the running speed by increasing the length of each stride. As a result, he maintained the same stride frequency while increasing the breathing frequency, which can also be reflected in Figure 13(b) as the corresponding LRC ratio shifts to 1 : 1. Therefore, RunBuddy is able to provide details of the dynamics in stride/breath frequencies to regular runners, which help them better understand their running progress. It is worth mentioning that one of our subjects who runs regularly is very enthusiastic about the running rhythm results provided by RunBuddy. He thought the result from RunBuddy is a valuable feedback, which could help even regular runners “push that extra little bit”.

As shown in Figure 13(c), it took the non-runner one and a half minutes to reach a relatively steady stride frequency. However, it can be seen that the subject kept changing the breathing frequency. As reported by the subject, she found it difficult to adjust her breath to her stride during running. Actually, the shortly dropped breathing frequency at around 2:45 and 7:50 minutes reflect her effort in trying to adjust her breath frequency. However, due to the lack of running experience and relatively low fitness level, she failed to find a comfortable running rhythm and kept adjusting the breathing frequency throughout the run. This also leads to a varying LRC ratio over time. Therefore, by continuously monitoring the running rhythm along with other information (e.g., stride), RunBuddy has potential to guide the non-runners to find a

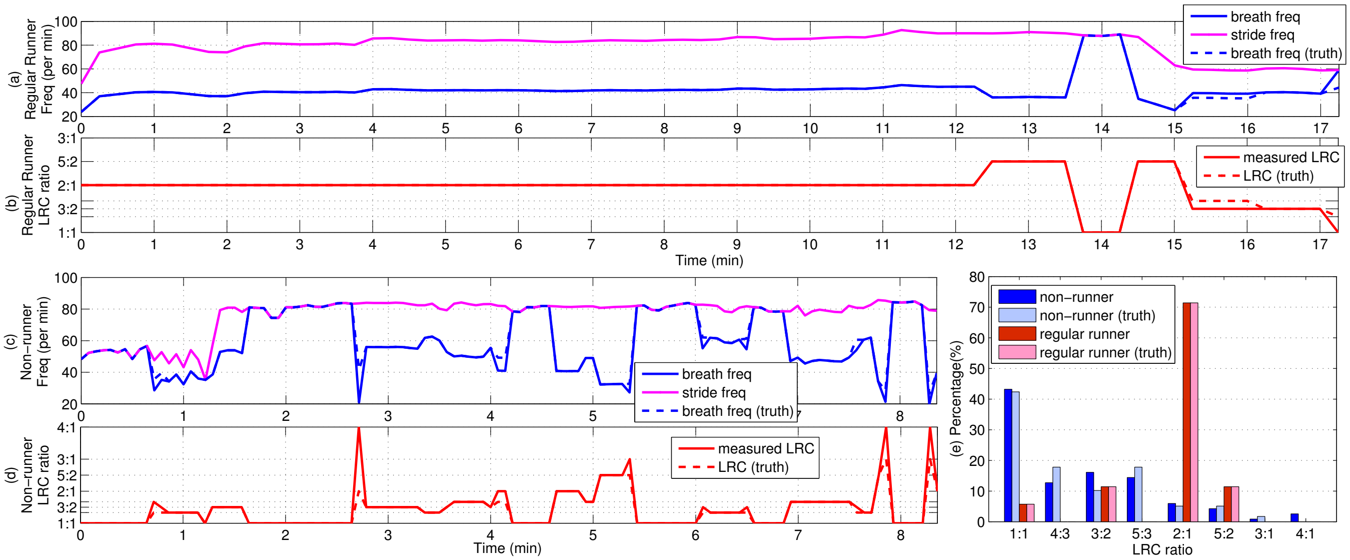

Figure 13. The detection result of typical runs from a regular runner (17.2 **min) and a non-runner** (8.4 **min). (a) and (c) show the frequency of breath and stride for regular runner and non-runner, respectively. (b) and (d) show their LRC ratios over time. (e) shows the distributions of their LRC ratio used during running.**

Model	CPU	RAM	Battery Cap.	cpu load (avg/max)	runtime (Audio)	runtime (Motion)	Battery Usage
Samsung Galaxy Nexus	Dual Core 1.2GHz	1GB	1750 mAh	4.7%/7%	595ms/5sec	3.4ms/5sec	5% per hr
Motorola Moto G	Quad Core 1.2GHz	1GB	2070 mAh	3.5%/4%	560ms/5sec	3.0ms/5sec	5% per hr
LG Nexus 4	Quad Core 1.5GHz	2GB	2100 mAh	8.2%/14%	522ms/5sec	2.8ms/5sec	5% per hr
LG Nexus 5	Quad Core 2.3GHz	2GB	2300 mAh	5.2%/8%	318ms/5sec	2.2ms/5sec	4% per hr

Table 2. The overhead of RunBuddy on different smartphones. The computational overhead is measured by the overall CPU load of RunBuddy and the time consumed for each pipeline to process 5-second data. Power consumption is measured by the battery usage per hour.

comfortable running rhythm (e.g. play music with proper tempo based on the analysis of the LRC measurements)

Figure 13(e) shows the percentage of each LRC ratio of the non-runner and the regular runner. We can see that the dominant LRC ratio (2:1) of the regular runner was used for around 70% of the run, whereas the non-runner used 8 different LRC ratios, with the dominant LRC ratio 1:1 being used for only 43% of the time. This result indicates that the coupling effect between breathing and stride of the regular runner is stronger than that of the non-runner. This observation is consistent with the literature of physiology [8], suggesting that tighter coordination between limb rhythms and respiration may reduce the metabolic cost of a movement.

Computation Overhead and Power Consumption

In this section, we evaluate the overhead of RunBuddy implemented on different Android platforms, including Galaxy Nexus [35], Moto G [27], Nexus 4 [28] and Nexus 5 [19]. Here we focus on the the evaluation of two major pipelines, which are, *audio pipeline*, including the low-pass filtering, MFCC feature extraction and breath detection, and *motion pipeline* consisting of vertical projection and stride detection components. Specifically, we measured the runtime for each pipeline to process 5-second data. The result is shown in Table 2. Overall, the results show that RunBuddy takes a small amount of processing time and CPU load because of the lightweight design of sensing pipelines, which in turn results in a relatively low battery usage (around 5%/hour).

CONCLUSION

In this paper, we present RunBuddy – the first smartphone-based system for continuous running rhythm monitoring. It only utilizes commodity devices including smartphone and bluetooth headset, and is convenient and unobtrusive to users. RunBuddy adopts lightweight and efficient signal processing algorithms for detecting breathes and strides using accelerometer and microphone, respectively. It then leverages a physiological model called Locomotor Respiratory Coupling (LRC) to correlate the sensing results, which significantly improves the performance. RunBuddy is evaluated through extensive experiments involving 13 subjects and 39 runs. Our results show that RunBuddy can accurately measure the running rhythm in terms of LRC ratio in 92.7% of the time. By leveraging LRC models, RunBuddy yields robust performance even in the presence of various environmental noises. RunBuddy is able to provide physiological details of running (e.g., the stability of LRC ratio) that can be used to help users to better understand the running process and improve the running experience. Moreover, the real-time running rhythm measurement provided by RunBuddy also has great potential in helping runners better coordinate their breathing and strides, and improve their running performance.

ACKNOWLEDGMENTS

This research was supported in part by the U.S. National Science Foundation under grants CNS-1218475, CNS-0954039 (CAREER), CNS-1250180 and CNS-1253506 (CAREER). We thank the anonymous reviewers and the paper shepherd for their valuable feedbacks.

REFERENCES

1. EA Aaron, KC Seow, BD Johnson, and JA Dempsey. 1992. Oxygen cost of exercise hyperpnea: implications for performance. *Journal of Applied Physiology* 72, 5 (1992), 1818–1825.
2. Mark H Anshel and Dan Q Marisi. 1978. Effect of music and rhythm on physical performance. *Research Quarterly. American Alliance for Health, Physical Education and Recreation* 49, 2 (1978), 109–113.
3. P Bernasconi, P Bürki, A Bührer, EA Koller, and J Kohl. 1995. Running training and co-ordination between breathing and running rhythms during aerobic and anaerobic conditions in humans. *European journal of applied physiology and occupational physiology* 70, 5 (1995), 387–393.
4. P Bernasconi and J Kohl. 1993. Analysis of co-ordination between breathing and exercise rhythms in man. *The Journal of physiology* 471, 1 (1993), 693–706.
5. Jacob T Biehl, Piotr D Adamczyk, and Brian P Bailey. 2006. Djogger: a mobile dynamic music device. In *CHI'06 Extended Abstracts on Human Factors in Computing Systems*. ACM, 556–561.
6. Body Mass Index 2004. (2004). http://apps.who.int/bmi.
7. Agata Brajdic and Robert Harle. 2013. Walk detection and step counting on unconstrained smartphones. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 225–234.
8. Dennis M Bramble and David R Carrier. 1983. Running and breathing in mammals. *Science* 219, 4582 (1983), 251–256.
9. Budd Coates and Claire Kowalchik. 2013. *Runner's World Running on Air: The Revolutionary Way to Run Better by Breathing Smarter*. Rodale.
10. Rodrigo De Oliveira and Nuria Oliver. 2008. TripleBeat: enhancing exercise performance with persuasion. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*. ACM, 255–264.
11. F Garlando, J Kohl, EA Koller, and P Pietsch. 1985. Effect of coupling the breathing-and cycling rhythms on oxygen uptake during bicycle ergometry. *European journal of applied physiology and occupational physiology* 54, 5 (1985), 497–501.
12. Charles P Hoffmann, Gérald Torregrosa, and Benoît G Bardy. 2012. Sound stabilizes locomotor-respiratory coupling and reduces energy cost. *PloS one* 7, 9 (2012), e45206.
13. Toshiki Iso and Kenichi Yamazaki. 2006. Gait analyzer based on a cell phone with a single three-axis accelerometer. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*. ACM, 141–144.
14. Jabra Wave Mobile Bluetooth Headset 2015. (2015). http://www.jabra.com/products/bluetooth/jabra_wave/jabra_wave.
15. CI Karageorghis, Leighton Jones, and DP Stuart. 2007. Psychological effects of music tempi during exercise. (2007).
16. J Kohl, EA Koller, and M Jäger. 1981. Relation between pedalling-and breathing rhythm. *European journal of applied physiology and occupational physiology* 47, 3 (1981), 223–237.
17. Eric C Larson, Mayank Goel, Gaetano Boriello, Sonya Heltshe, Margaret Rosenfeld, and Shwetak N Patel. 2012. Spirosmart: Using a microphone to measure lung function on a mobile phone. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 280–289.
18. Jonathan Lester, Carl Hartung, Laura Pina, Ryan Libby, Gaetano Borriello, and Glen Duncan. 2009. Validated caloric expenditure estimation using a single body-worn sensor. In *Proceedings of the 11th international conference on Ubiquitous computing*. ACM, 225–234.
19. LG Nexus 5 2015. (2015). https://www.google.com/nexus/5/.
20. Fan Li, Chunshui Zhao, Guanzhong Ding, Jian Gong, Chenxing Liu, and Feng Zhao. 2012. A reliable and accurate indoor localization method using phone inertial sensors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 421–430.
21. Stephen H Loring, Jere Mead, and Thomas B Waggener. 1990. Determinants of breathing frequency during walking. *Respiration physiology* 82, 2 (1990), 177–188.
22. Hong Lu, Jun Yang, Zhigang Liu, Nicholas D Lane, Tanzeem Choudhury, and Andrew T Campbell. 2010. The Jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 71–84.
23. DONALD A Mahler, BERNADETTE Hunter, TIMOTHY Lentine, and JOSEPH Ward. 1991a. Locomotor-respiratory coupling develops in novice female rowers with training. *Medicine and science in sports and exercise* 23, 12 (1991), 1362–1366.
24. DONALD A Mahler, CHRISTOPHER R Shuhart, ELIZABETH Brew, and THERESE A Stukel. 1991b. Ventilatory responses and entrainment of breathing during rowing. *Medicine and science in sports and exercise* 23, 2 (1991), 186–192.
25. William J McDermott, Richard EA Van Emmerik, and Joseph Hamill. 2003. Running training and adaptive strategies of locomotor-respiratory coordination. *European journal of applied physiology* 89, 5 (2003), 435–444.
26. David Mizell. 2003. Using gravity to estimate accelerometer orientation. In *2012 16th International Symposium on Wearable Computers*. IEEE Computer Society, 252–252.

27. Motorola Moto G 2015. (2015). http://www.motorola.com/us/consumers/moto-g/Moto-G/moto-g-pdp.html.

28. Nexus 4 2015. (2015). http://www.google.com/intl/ALL/nexus/4/.

29. Shahriar Nirjon, Robert F Dickerson, Qiang Li, Philip Asare, John A Stankovic, Dezhi Hong, Ben Zhang, Xiaofan Jiang, Guobin Shen, and Feng Zhao. 2012. MusicalHeart: A hearty way of listening to music. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 43–56.

30. Nuria Oliver and Fernando Flores-Mangas. 2006. MPTrain: a mobile, music and physiology-based personal trainer. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*. ACM, 21–28.

31. Oxycon Mobile 2015. (2015). http://www.carefusion.com.

32. Jiapu Pan and Willis J Tompkins. 1985. A real-time QRS detection algorithm. *Biomedical Engineering, IEEE Transactions on* 3 (1985), 230–236.

33. David J Paterson, Graeme A Wood, Alan R Morton, and John D Henstridge. 1986. The entrainment of ventilation frequency to exercise rhythm. *European journal of applied physiology and occupational physiology* 55, 5 (1986), 530–537.

34. Rangaraj M Rangayyan. 2002. *Biomedical signal analysis*. IEEE press New York.

35. Samsung Galaxy Nexus 2015. (2015). http://www.samsung.com/us/mobile/cell-phones/SPH-L700ZKASPR.

36. Ben C Sporer, Glen E Foster, A William Sheel, and Donald C McKenzie. 2007. Entrainment of breathing in cyclists and non-cyclists during arm and leg exercise. *Respiratory physiology & neurobiology* 155, 1 (2007), 64–70.

37. Mayu Sumida, Teruhiro Mizumoto, and Keiichi Yasumoto. 2013. Estimating heart rate variation during walking with smartphone. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 245–254.

38. Sébastien Villard, Jean-François Casties, and Denis Mottet. 2005. Dynamic stability of locomotor respiratory coupling during cycling in humans. *Neuroscience letters* 383, 3 (2005), 333–338.

39. Voyager Legend Mobile Bluetooth Headset 2015. (2015). http://www.plantronics.com/us/product/voyager-legend.

40. Voyager Pro HD Mobile Bluetooth Headset 2015. (2015). http://www.plantronics.com/us/product/voyager-pro-hd.

41. Andong Zhan, Marcus Chang, Yin Chen, and Andreas Terzis. 2012. Accurate caloric expenditure of bicyclists using cellphones. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 71–84.