SoundWrite: Text Input on Surfaces through Mobile Acoustic Sensing

Maotian Zhang[1], Panlong Yang[1], Chang Tian[1], Lei Shi[1], Shaojie Tang[2], and Fu Xiao[3]
[1]College of Communications Engineering, PLA University of Science and Technology, China
[2]The University of Texas at Dallas, USA
[3]College of Computer, Nanjing University of Posts and Telecommunications, China
{maotianzhang, panlongyang, leishi9018, tangshaojie}@gmail.com,
tianchang163@163.com, xiaof@njupt.edu.cn

ABSTRACT

Interacting with explosively growing mobile devices is becoming imperative. This paper presents SoundWrite, a mobile acoustic sensing system that enables text input into commercial off-the-shelf devices without any accessories. SoundWrite leverages the embedded microphone to capture subtle audio signals emitted from writing text on common found surfaces (*e.g.*, a wood table). It then extracts distinguishable features from both time and frequency information of received signals to recognize the text. We prototype SoundWrite on Smartphones as an Android application, and perform in-depth evaluation. The evaluation results validate the effectiveness and robustness of SoundWrite, and demonstrate that it could achieve an average recognition accuracy of above 90%.

1. INTRODUCTION

With the explosive increase of mobile devices, interacting with them is becoming imperative. Despite there exists various input methods, such as wireless keyboard for PCs and on-screen soft keyboards in smartphones, increasingly shrinking device interfaces (*e.g.*, Apple Watch) call for more creative, comfortable and speedy input schemes.

Several input systems have been recently developed to meet the challenge. Existing ring-form wearable devices leveraged multiple embedded sensors and multi-dimensional sensing information to either detect finger motions [1] or build virtual keyboard underneath the hand [2]. However, these accessories demand additional cost and may be less prone to popularize. Another example was UbiK [3], which also required a keyboard outline printed on a piece of paper and then used microphones in smartphones to localize the keystrokes. While PhonePoint Pen [4] just used mobile phones to recognize input text, it incurred constrained conditions such as large character sizes.

In this paper, we present SoundWrite, a new method to enable mobile text input on commercial off-the-shelf (COTS) devices. SoundWrite exploits acoustic sensing techniques to recognize text written on common found surfaces, such as a wood table. The input text could be sensed using microphones in mobile devices. And then, SoundWrite extracts features from captured signals to recognize the text. SoundWrite is usable with any mobile devices that support audio capturing and processing, and is flexible for most scenarios, for instance, on a meeting room table, cafe table, or office desk.

Our goal is to design a comfortable and speedy input scheme without additional accessories. Developing such system poses a key challenge: *Extracting distinguishable features from subtle acoustic signals emitted by sliding the surfaces.*

Although previous acoustic sensing techniques have been utilized for ranging [5] or developing mobile games [6], these approaches needed self-generating audio of mobile devices, and so that they failed to deal with our scenario in which text need be detected and recognized from subtle and external audio signals. To address the challenge, SoundWrite firstly divides character or text into the combination of multiple primitive strokes (*e.g.*, a “—" and “|"). We then leverage the time and frequency information of received acoustic signals to recognize and distinguish strokes. Particularly, the stroke input could be detected by the energy burst in time domain, and the strokes could be characterized by the amplitude spectrum density (ASD) of sliding sound in frequency domain.

We have implemented SoundWrite as a prototype application in Android platform. It could classify seven strokes, and meanwhile show robustness to ambient noises. The evaluation results validate the effectiveness of SoundWrite, and demonstrate that SoundWrite could achieve above 90% accuracy.

The main contributions of the paper are as follows:

- We present SoundWrite, a mobile acoustic sensing system that enables comfortable and speedy text input on COTS devices without any accessories.
- We propose to leverage both time and frequency information of received audio signals to extract distinguishable features of written strokes. Careful and adequate measurement is provided to verify the feasibility of extracted features.
- We evaluate SoundWrite through micro-benchmarks. The results demonstrate that SoundWrite could achieve considerable recognition accuracy of above 90%.

The rest of this paper is organized as follows. Section 2 presents the system overview of SoundWrite. We describe the system components, stroke input detection and stroke recognition, in Section 3 and Section 4, respectively. Implementation and evaluations are presented in Section 5 and Section 6, respectively. We also investigate the limitations of the proposed system in Section 7, and discuss the related work in Section 8. Finally, we conclude this work in Section 9.

SmartObjects'15, September 7, 2015, Paris, France.
© 2015 ACM. ISBN 978-1-4503-3535-5/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2797044.2797045.

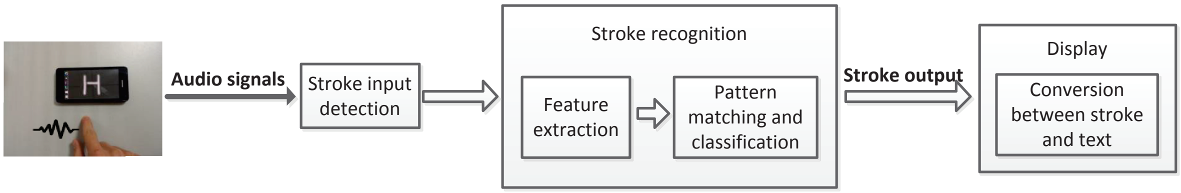

Figure 1: SoundWrite system overview.

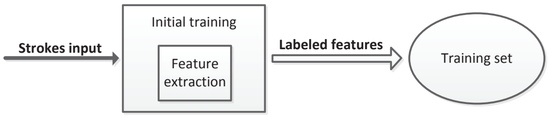

Figure 2: SoundWrite training procedure.

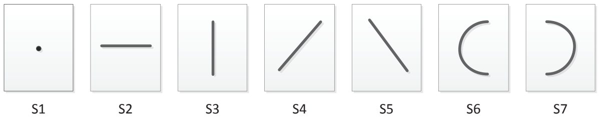

Figure 3: Basic strokes used for written text entry.

2. SOUNDWRITE SYSTEM OVERVIEW

SoundWrite is a mobile acoustic sensing system that enables text input on commercial mobile devices. It can be used on common surfaces as long as they can generate audible sounds when the user inputs text with fingertip and nail margin. As a usage condition, the mobile device must be static when performing SoundWrite. And, whenever the mobile device is put to a new surface or location, SoundWrite needs repeating the initial setup.

Figure 1 illustrates the architecture and work flow of SoundWrite. The architecture of SoundWrite consists of the following three key components.

- *Stroke input detection.* SoundWrite performs an energy detection algorithm that leverages the time domain patterns to detect the input text and further segments them into pieces covering different strokes. It can also suppress ambient noises.
- *Stroke recognition.* The frequency domain information is also leveraged to extract acoustic features of strokes. We explore a simple but efficient feature used inside SoundWrite. Then, SoundWrite runs a computational efficiently algorithm that matches the features with the training set (as shown in Figure 2), and classifies the strokes.
- *Conversion between stroke and text.* Characters (*e.g.*, English or Chinese character) can be viewed as a combination of multiple primitive strokes [4]. For instance, the alphabet “A" is composed of three strokes, *i.e.*, “╱", “╲" and “—". As Figure 3 shows, basically, the variety of combinations of strokes S2 - S7 is able to form all the 26 English letter. However, for other language, take a Chinese character “太" for example, the stroke “•" (S1 in Figure 3) is required. As a result, these seven strokes S1 - S7 requires to be recognized. SoundWrite uses the similar method presented in [4] to transform strokes into characters or text, namely, it treats each character as a combination of multiple strokes.

The work flow of SoundWrite is as follows. The user writes characters or text on surfaces, such as the top of a wood table. SoundWrite then detects these text input and further segment them into pieces covering different strokes. After that, SoundWrite extracts acoustic features and run pattern matching and classification compared with the training set. Figure 2 shows the initial training procedure. The different combinations of recognized strokes are then transformed into characters or text. Finally, the written text display on the screen.

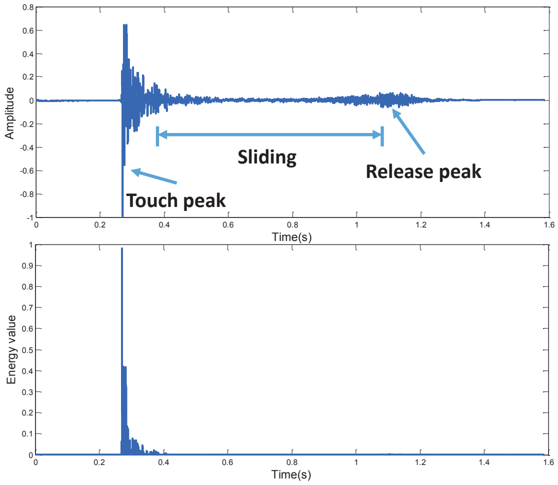

Figure 4: Example of a stroke's acoustic signal and its energy.

3. INPUT DETECTION ON SURFACES

In this part, we show how to detect the text input, and segment the received audio signals into pieces so that each piece represents a stroke.

3.1 Energy Detection

The amount of audio energy emitted from the interaction between the finger and surfaces is associated with both the speed of finger motion and the physical properties of the surface [7]. To understand the amount of available audio energy on commonly found surfaces, we perform several stroke input tests (*e.g.*, a “—") with moderate speed nearby the smartphone in a quiet noise environment.

The audio signals produced by touching and sliding the surfaces form a cluster of energy burst. In the beginning, the energy burst is enough to detect the start of input event, as Figure 4 shows. Besides, since the position of inputting text is close to the mobile device, the input sounds are much stronger than ambient noises. SoundWrite leverages such unique profiles to detect the text input, and single out audio signals.

As shown in Figure 4, the audio signal could be divided into three parts: *Touch peak*, *Sliding*, and *Release peak*. It is worth noting that the audio signal of a keystroke can find three peaks which are

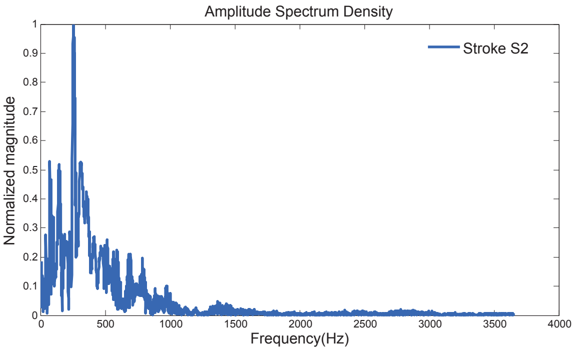

Figure 5: Amplitude spectrum density of a stroke.

touch peak, hit peak and release peak [8]. Formally, the energy of an audio signal $x(t)$ can be represented by $E(t) = kx(t)^2$, where k is a constant. We show the energy level of the audio signal in the lower part of Figure 4. To find *Touch peak*, we first conduct the moving average to suppress the ambient noise. $A(t)$ denotes the energy level through a moving average window with size W, and is given by:

$$A(t) = \frac{1}{W} \sum_{n=t-W-1}^{t} E(t)$$

The window size W is chosen to be $5ms$, *i.e.*, $44100Hz \times 5ms \approx 221$ sample points, according to our experiments. We use the similar approach presented in [8] to detect the timing of a *Touch peak* (denoted by t_{touch}. It can be acquired by the following definition:

$$\begin{aligned} t_{touch} &= \arg\max_{t} A(t) \\ s.t. \ &: \ A(t) \geq A(i) \text{ for } t - 50ms \leq i \leq t + 1150ms \\ &\quad \text{and } A(t) \geq A_\epsilon \end{aligned}$$

where the search range is empirically set to be $1200ms$ which is typically a maximum period of a stroke, and A_ϵ is set to 0.1.

3.2 Segment

We further segment the received audio signal to several pieces, and the time period of each piece covers a stroke. After detecting the *Touch peak*, for each t_{touch}, we take $50ms$ before t_{touch} and $1150ms$ after t_{touch} to form a segment of an audio signal covering a stroke. According to experimental observations, we set the width of each piece to $1200ms$, containing $44100Hz \times 1200ms = 52920$ sample points.

4. STROKE RECOGNITION

In this section, we describe how SoundWrite recognizes different strokes. We first extract the feasible feature from experimental observations. Then, the strokes can be classified by pattern matching using extracted feature.

4.1 Feature Extraction

SoundWrite leverages frequency information to extract the feasible feature to distinguish the strokes. We first perform a set of experiments to answer the following question: Do different strokes generate distinct frequency profiles that can be utilized as recognition feature?

In the experiments, a HUAWEI U9508 smartphone is placed on the top of a wood table in an office room, and the user writes the strokes nearby (as shown in the left photo of Figure 1). The upper part of Figure 4 shows stroke S2's audio signal captured by the front-microphone of the listening smartphone. Further, we observe the frequency domain profiles of stokes from the received signals' ASD. ASD of discrete audio signal $x(t)$ is given by: $FFT(x(t))$. Figure 5 shows a stroke's ASD, which is normalized with respect to the maximum across all frequency bins of each. The majority of spectrum concentrates within $1Hz$ and $3000Hz$. Furthermore, we plot ASD of two different strokes within $50Hz$ and $350Hz$, as Figure 6 shows. It is concluded that the ASD of two different strokes exhibit distinct values across frequencies, and their peaks are at different frequency bins. As a result, we are able to leverage this feature to recognize and distinguish different strokes.

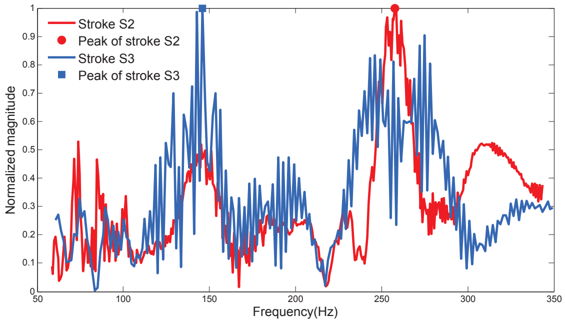

Figure 6: ASD of two different strokes, S2 and S3.

4.2 Pattern Matching and Classification

We employ a pattern matching and classification method to recognize and distinguish strokes. We first note that for stroke S1 "●", it could be recognized by energy detection with a very short time period compared the other six strokes. For strokes S2 to S7, the training set contains the ASD features of these six strokes. SoundWrite extracts the ASD features from the audio signals after detecting a stroke input event. These ASD features form a vector, and then perform pattern matching with corresponding vectors in the training set.

The pattern matching algorithm utilizes a simple but efficient metric, Euclidean distance, for comparison. In similar with a method presented in [3], SoundWrite runs a nearest-neighbor based pattern matching that compares the extracted feature vectors with those in the training set. The current stroke is recognized as long as it gets the minimum distance with one of the training strokes.

5. IMPLEMENTATION

We implemented SoundWrite as an application on smartphones running Android platform. All components described in Figure 1 were implemented. In particular, taking the software design into consideration, these components are defined as `InputDetection`, `FeatureExtraction`, `Classification`, `Conversion` and the GUI. These software components could be incorporated into different mobile devices conveniently.

We only use one of two microphones in the smartphone. The sampling frequency of audio is set to be $4.41kHz$ by default. The samples are put into an audio buffer, and then the stroke recognition algorithm runs on these audio instances.

We also implemented a sever side implementation in Matlab, which helped us perform signal processing and statistical analysis. At the early stage, we obtained audio signals from the smartphone,

	Office	Library	Cafe
Noise level	46.8dB	40.5dB	62.3dB
P_{mis}	0.3%	0.0%	0.6%
P_{fls}	0.0%	0.0%	0.1%

Table 1: Stroke input detection in different environments

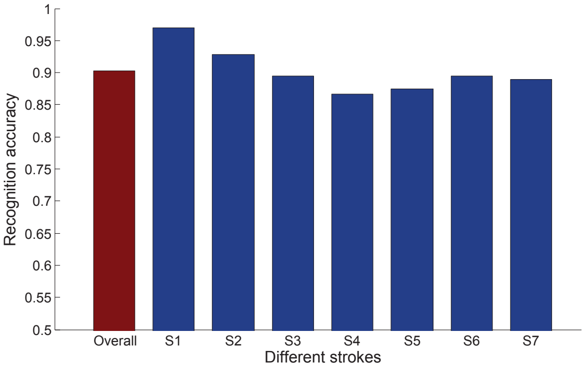

Figure 7: Accuracy of stroke recognition.

and processed them in Matlab using programmed components. We then prototyped these codes on Java for on-phone processing.

6. EVALUATION

In this section, we evaluate the performance of SoundWrite in terms of accuracy, robustness, as well as several impact factors in a variety test scenarios.

6.1 Experimental Setting

In the experiments, we use three different kinds of smartphones to run SoundWrite, and those mobile devices are placed on the top of different common found surfaces. In a default setting scenario, a HUAWEI U9508 smartphone is placed on the top of a wood table in a quiet office room, and the user writes the text or strokes beneath it.

6.2 Micro-benchmarks

We present the evaluation of SoundWrite through micro-benchmark tests, including the overall accuracy of recognition and the impact of the underlying factors, such as the input speed and the location of writing, on the detection accuracy.

6.2.1 *Accuracy of input detection in different environments*

At first, we evaluate the stroke input detection in different daily environments, namely, an office room, library, and cafe. We randomly choose a surface (typically, a wood table) in each environment. In each test, for simplicity, the user writes 20 instances per stroke.

Table. 1 shows the mis-detection (P_{mis}) and false-alarm (P_{fls}) rates. In both three scenarios, both P_{mis} and P_{fls} are negligible. The test results reveal the stroke input detection of SoundWrite is accurate and robust, when the noise level is lower than 70dB. We note that if there are bursty noises, the detection error rates may slightly increase. We will address this challenge in our future work.

6.2.2 *Accuracy of stroke recognition*

We perform an overall test of stroke recognition in a quiet office room. Each strokes are written below the smartphone sequentially and repeats 20 times. Figure 7 illustrates both the overall accuracy and accuracy of each stroke achieved by running SoundWrite. The overall recognition accuracy is 90.3%. While the accuracy of stroke S1 is highest among all strokes, that of others are not much different. The reason is that the time period of stroke S1 is very short so that it could be directly recognized by energy detection algorithm. This test and its results validate the effectiveness of SoundWrite' stoke recognition.

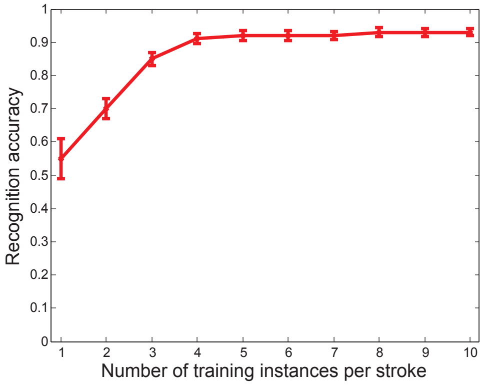

Figure 8: Impact of initial training set size on recognition accuracy.

	HUAWEI	MOTOROLA	SAMSUNG
Accuracy	90.3%	88.5%	90.1%

Table 2: Impact of different kinds of mobile devices on recognition accuracy.

6.2.3 *Impact of initial training*

Recalling the description of stroke recognition in Section 4.2, pattern matching and classification of strokes relies on the feature vectors in the training set. That is the initial training set influences on the accuracy of SoundWrite. We show the accuracy achieved by SoundWrite with the increase of the initial training size. With one initial training, the recognition accuracy is around 60%. the accuracy escalates to above 90% averagely as the number of training instances increases to 4. It is suggested to input 4 instances per stroke to achieve considerable accuracy according to the test results.

6.2.4 *Accuracy of different kinds of devices*

We also use the other two mobile devices, SAMSUNG G3568V and MOTOROLA MT887, to evaluate the performance of SoundWrite. Table 2 presents the evaluation results. The accuracy varies slightly among different mobile devices. Varying microphone quality of these three devices may bring about the small difference of achieved accuracy.

7. LIMITATIONS AND DISCUSSION

SoundWrite has made some advances towards interacting with mobile devices. Still, it bears several limitations that needs further improvement.

Disturbed by bursty noises. The near-field noise sources, *e.g.*, a sequence of sudden clicks, disturbs the energy detection of text input. The emerging noises may override audio signal of the text. We are investigating the benefits of multi-dimensional sensing using multiple sensors. The motion sensors, such as an accelerometer and gyroscope, can provide complementary information of text recognition.

Requiring initial training. So far, SoundWrite requires initial training at fixed positions, which limits its usability. According to the results in SubSection 6.2.3, 4 training instances per stroke is enough to achieve considerable accuracy. This still is a burden for the mobile user. We are exploring new scheme to get rid of the bondage of initial training. The physical distinction of each stroke is investigated for further enhancement.

Quick writing and ligatures. Writing quite fast ($< 1.5s$ per character) and ligatures degrade recognition accuracy. The reason lies in that the features of fast written text makes a big difference with that in training set. It is suggested that the user keeps consistent input speed, and clearly writes single alphabet. We try to employ an association input method to improve the usability.

8. RELATED WORK

Text input system. Text input is one of fundamental function for mobile devices, such as smartphone and tablet. Except from the existing hardware accessory (*e.g.*, an Apple wireless keyboard) and software systems (*e.g.*, a projection keyboard [9]), recently, several text input systems have been developed, leveraging multiple sensing methods and various sensors embedded into mobile devices. Gummeson *et al.*, designed an wearable ring for gestures and text input, using both acoustic sensing and accelerometer information [1]. TypingRing [2] is also a wearable ring that is composed of multiple sensors and builds an invisible standard keyboard. However, these system requires additional and expensive hardware modules. As for using the COTS devices, PhonePoint Pen [4] uses inbuilt accelerometer in mobile phones to recognize human writing in air. However, this method demands writing large character sizes. Recently, Wang *et al.*, have proposed UbiK, an approach to mobile text input that recognizes keystrokes through acoustic localization. This method, however, requires an extra accessory, *e.g.*, a piece of paper, and careful typing in this virtual keyboard.

Acoustic sensing. Acoustic sensing has enabled varied innovative applications in many fields, such as health monitoring [10] [11] and mobile games [6]. For example, Spartacus [12] enables spatially-aware neighboring device interaction leveraging acoustic sensing and Doppler effect. Zhu *et al.*, presented the context-free attacks using keyboard acoustic emanations [8]. Acoustic sensing could also be used for recommendation systems [13] [14], ranging and localization systems [15] [5] and meeting systems [16] [17]. A variety of applications based on acoustic sensing and the new processing methods inspire us to develop SoundWrite.

9. CONCLUSION

In this paper, we have presented SoundWrite, a mobile acoustic sensing system that enables effective text-entry on common found surfaces for COTS devices. SoundWrite leverages time and frequency information of received audio signals to extract feasible features. Based on ASD features, SoundWrite runs efficient pattern matching and classification algorithm to recognize and distinguish strokes. We have implemented SoundWrite in Android platform. And, evaluation results validate the effectiveness of the SoundWrite design, and demonstrate it achieves above accuracy of 90%. In the future, we would like to improve the accuracy of SoundWrite, and eliminate the initial training for better usage experience.

Acknowledgments

This research is partially supported by NSF China under Grants, No. 61232018, 61272487, 61373137.

10. REFERENCES

[1] J. Gummeson, B. Priyantha, and J. Liu, “An energy harvesting wearable ring platform for gesture input on surfaces,” in *MobiSys’14*. ACM, pp. 162–175.

[2] S. Nirjon, J. Gummeson, D. Gelb, and K.-H. Kim, “Typingring: A wearable ring platform for text input,” in *MobiSys’15*. ACM.

[3] J. Wang, K. Zhao, X. Zhang, and C. Peng, “Ubiquitous keyboard for small mobile devices: harnessing multipath fading for fine-grained keystroke localization,” in *MobiSys’14*. ACM, pp. 14–27.

[4] S. Agrawal, I. Constandache, S. Gaonkar, R. Roy Choudhury, K. Caves, and F. DeRuyter, “Using mobile phones to write in air,” in *MobiSys’11*. ACM, pp. 15–28.

[5] P. Lazik and A. Rowe, “Indoor pseudo-ranging of mobile devices using ultrasonic chirps,” in *SenSys’12*. ACM, pp. 99–112.

[6] Z. Zhang, D. Chu, X. Chen, and T. Moscibroda, “Swordfight: enabling a new class of phone-to-phone action games on commodity phones,” in *MobiSys’12*. ACM, pp. 1–14.

[7] A. Akay, “Acoustics of friction,” *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1525–1548, 2002.

[8] T. Zhu, Q. Ma, S. Zhang, and Y. Liu, “Context-free attacks using keyboard acoustic emanations,” in *CCS’14*. ACM, pp. 453–464.

[9] C. Harrison, H. Benko, and A. D. Wilson, “Omnitouch: wearable multitouch interaction everywhere,” in *UIST’11*. ACM, pp. 441–450.

[10] T. Hao, G. Xing, and G. Zhou, “isleep: unobtrusive sleep quality monitoring using smartphones,” in *SenSys’13*. ACM, p. 4.

[11] T. Rahman, A. T. Adams, M. Zhang, E. Cherry, B. Zhou, H. Peng, and T. Choudhury, “Bodybeat: a mobile system for sensing non-speech body sounds,” in *MobiSys’14*. ACM, pp. 2–13.

[12] Z. Sun, A. Purohit, R. Bose, and P. Zhang, “Spartacus: spatially-aware interaction for mobile devices through energy-efficient audio sensing,” in *MobiSys’13*. ACM, pp. 263–276.

[13] X. Bao, S. Fan, A. Varshavsky, K. Li, and R. Roy Choudhury, “Your reactions suggest you liked the movie: automatic content rating via reaction sensing,” in *UbiComp’13*. ACM, pp. 197–206.

[14] S. Nirjon, R. F. Dickerson, Q. Li, P. Asare, J. A. Stankovic, D. Hong, B. Zhang, X. Jiang, G. Shen, and F. Zhao, “Musicalheart: A hearty way of listening to music,” in *SenSys’12*. ACM, pp. 43–56.

[15] Z. Sun, A. Purohit, K. Chen, S. Pan, T. Pering, and P. Zhang, “Pandaa: physical arrangement detection of networked devices through ambient-sound awareness,” in *UbiComp’11*. ACM, pp. 425–434.

[16] C. Luo and M. C. Chan, “Socialweaver: collaborative inference of human conversation networks using smartphones,” in *SenSys’13*. ACM, p. 20.

[17] S. Sur, T. Wei, and X. Zhang, “Autodirective audio capturing through a synchronized smartphone array,” in *MobiSys’14*. ACM, pp. 28–41.