On the Use of Sensor Nodes and Mobile Phones for the Assessment of Noise Pollution Levels in Urban Environments

Silvia Santini, Benedikt Ostermaier, Robert Adelmann
Institute for Pervasive Computing
ETH Zurich
CH-8092 Zurich, Switzerland
{santinis,ostermaier,adelmann}@inf.ethz.ch

***Abstract*—Recent work in the field of wireless sensing networks shows that cheap sensing platforms like wireless sensor nodes or mobile phones can be used, or "misued", for monitoring environmental pollution levels. In particular, several authors dedicated their attention to the problem of the assessment of environmental noise levels in urban environments, a complex and expensive task that public administrations and often also private entities are willing or forced to take over. In this paper, we review the approaches that have been presented within the wireless sensing research community and report on our own experiences in assessing noise pollution levels using both sensor nodes and mobile phones. Drawing form other authors' and our own work, we outline common pitfalls and open issues in the implementation of wireless noise monitoring systems and provide practical considerations that can speed up and advance the development of such systems.**

I. Introduction

The negative effects of environmental pollutants on human health and quality of life are nowadays unquestioned [1]. The definition of adequate strategies for the abatement of pollutants levels is therefore becoming a prime political issue in many developed and developing countries. The first step towards the identification of effective abetment strategies typically consists in the acquisition of data describing sources and distributions of the pollutants under consideration. Performing the thereby required data collection campaigns, however, is an expensive and cumbersome task and technical solutions that can reduce their costs and increase their reliability are therefore highly in demand.

Focusing on the assessment of environmental noise pollution in urban areas, we already outlined the advantages as well as the technical challenges that the use of wireless sensor networks may bring in this context [2]. In particular, we tested the feasibility of off-the shelf wireless sensor network platforms, like the *Tmote invent* from Moteiv [3], to be used as environmental noise sensors and outlined the thereby arising problems. For instance, we showed that the built-in microphones of the nodes are poorly calibrated and thus nearby devices typically produce misaligned acoustic responses, even if they actually record the same acoustic signal. This lack of calibration prevents the possibility to use such readings for mapping noise levels across a neighborhood or city, since measurements collected by different nodes do not refer to a common and reliable reference. Solving this problem requires not only using alternative hardware platforms, but also the definition adequate procedures to perform frequent re-calibrations towards a reliable reference. In section II, we will wrap-up the available know-how on using sensor nodes for noise pollution monitoring, and report our latest experiences.

Moving beyond the traditional wireless sensor networks paradigm, several recently started research projects aim at leveraging the computational, communication, and sensing capabilities of mobile phones to capture data from the environment. For instance, it has been postulated that microphones of mobile phones may be used as cheap and ubiquitously present noise pollution sensors [4], [5]. The availability of noise pollution data may then allow to infer information about the traffic situation or the crowdedness of a public place, which in turn can be used to feed other applications like a restaurant locator [5]. In section III, we will provide an overview on projects that adopted mobile phones to capture noise pollution data and we will outline the benefits and limits of such approaches. Supporting our assertions with experimental results, we will show that noise data collected with mobile phones is often not accurate enough to support the above mentioned application scenarios. Furthermore, we will summarize common pitfalls in the design and implementation of noise level metering applications and provide practical guidelines to ease their workaround.

The contribution of this paper is therefore threefold. First, we provide an extensive overview on related work concerning the gathering of noise levels data using commercially available sensor nodes and mobile phone platforms. Second, we report a detailed analysis of experimentally gathered data to assess the feasibility of mobile phones to be used as noise pollution sensors. Third, we provide a summary of the main technical issues influencing the design and implementation of reliable systems for the assessment of noise pollution levels using mobile phones.

II. Assessment of noise pollution levels using wireless sensor nodes

In recent years, the European Commission dedicated increasing attention to the problem of environmental noise pollution [1], [6], as also the large number of funded research projects demonstrates [7]. The implementation of reliable noise monitoring and management systems indeed still requires research on several issues, spanning from the refinement of noise propagation models to the development of an adequate standard infrastructure for noise management. Along this line, the European Community recently issued a Directive that requires all agglomerations within the European Union with more than $250'000$ inhabitants to provide accurate mappings of noise pollution levels across their territory [6, p.1], [2]. Such *noise maps* must be provided on a yearly basis and made available to the public through adequate web interfaces. However, since currently available noise maps are mainly drawn upon synthetic data, they often provide only rough estimations of the actual perceived noise levels. Therefore, the European Commission emphasizes that, although synthetic data may still be used, "*every effort should be made to obtain accurate real data on noise sources*" [8, p.6].

In our own previous work we outlined how wireless sensor networks could help in satisfying this need for real noise pollution data with relatively cheap and marketable solutions [2], [9]. In particular, we tested three different different hardware platforms: the *Tmote invent* from Moteiv [3] and the *Tmote Sky*, also from Moteiv, equipped with either the $SBT80$ multimodality sensor board available from EasySen [10] or with a custom-made noise level meter. Although the Tmote invent features significantly better audio circuitry with respect to the EasySen board, both platforms provided quite unsatisfactory results. In particular, readings from different sensor nodes exhibit significant discrepancies, even if the measurements were taken concurrently in the same experimental setting. Additionally, acquiring and processing audio data on Tmote platforms heavily overloads the processor and quickly exhausts nodes' batteries.

To bypass the problems related to an inadequate audio sensor while still exploiting the potential of state-of-the-art networked sensing systems, we very recently investigated the possibility of interfacing the Tmote Sky platform with off-the-shelf sound level meters. Devices providing adequate output channels can be connected almost effortlessly to one of Tmote Sky's extension connectors. For instance, the Extech 407740 class 2 sound level meter makes the measured noise levels available on an analog output channel. We can thus easily gather the signal on this channel through a standard 3.5 mm TRS-connector (audio jack) and report it to one of the Tmote Sky's ADC input channels. A simple tinyOS[1] application can then read the measured noise levels, expressed in deciBel (dB), and log them on a back-end terminal through the serial connector of the Tmote. This simple prototype, although useful for collecting preliminary experimental data, does not represent a viable solution for a large scale deployment, due to the bulky form factor of the Extech 407740. As an inviting alternative, the SL328 hand-held sound level meter provides an agreeable compromise in terms of form-factor, accuracy and costs. However, it does not provide any standard interface to capture the measured noise levels and we are thus considering the possibility to develop a customized connection board.

Besides our own work, several other authors analyzed the possibility of using wireless sensor networks for noise pollution monitoring. For instance, in a well-known seminal paper back into 1999, Deborah Estrin and her co-authors mentioned fine-grained collection of noise data as one of the potential application scenarios for wireless sensor networks [11]. In the following years, other authors referred to noise monitoring as a possible application for wireless sensor networks, without however providing closer investigations [12]. Most recently, the *BikeNet* project [13] showed how average noise levels can be used to influence daily decisions like the choice of the cycling route to work. The prototype developed for the *BikeNet* project uses the Tmote invent platform to derive estimations of the actual noise levels in the immediate neighborhood of a cyclist. However, as reported in [2] and as we briefly recalled above, the Tmote invent platform doesn't lend itself well for accurate measurements of noise pollution levels.

Before moving on with our analysis, we would like to recall that a consistent body of literature deals with applications for wireless sensor networks that exploit acoustic measurements, like target or event detection and classification (e.g., shooters, birds or volcanic eruptions) or acoustic-based localization or communication [14]–[17]. For these applications, however, specific features of the audio signal, like its frequency spectrum (e.g., for birds classification) or the relative loudness or time shifting between two signals (for detection and localization) are of interest, and not the absolute loudness, like for the assessment of noise levels.

III. Assessment of noise pollution levels using mobile phones

As we already mentioned in the introduction, several research projects or even commercial applications[2] aim at using the microphone of commonly available mobile phones as ubiquitous environmental noise sensors. Measured noise levels can then contribute to the estimation of traffic loads or of the "healthiness" of a bike trail or to the preparation of noise maps [2], [5]. However, the reports on the usage of mobile phones for measuring noise levels presented so far do not mention important technical considerations and let the actual accuracy of the measured noise levels remain unascertained.

For instance, the MobGeoSens system [4] used built-in microphones and other sensors attached to mobile phones to collect pollutant levels in an urban environment. Examples of noise levels, expressed in dB, are presented, but the accuracy of this data as well as its coherence with measurements taken from nearby phones is not discussed at all.

[1] www.tinyos.net

[2] See, for instance, the *WideNoise* project: www.widetag.com/widenoise.

Within Microsoft's Nericell project [18], audio recordings from the built-in microphone of a smartphone constitutes the input of a honk-detection algorithm, which in turn feeds an estimator of the current traffic conditions. The authors investigate the influence of background noise and the sensitivity of the microphones on the performances of the honk-detector, but their approach does not support the assessment of the absolute of the actual noise levels. Similarly, the *CenceMe* system [19] uses simple Python scripts to capture audio signals on Nokia $N80$ and $N95$ mobile phones, but the data is processed on the mobile phone to determine whether it contains voice or just background noise.

Other efforts investigate the challenges and possibilities related to the use of mobile phones as "complex" sensors. For instance, Misra et al. [20], present several examples of how the microphone of mobile phones can be used for music applications. Furthermore, they underline that, being the development of mobile operating systems still ongoing, writing applications for mobile platforms that rely on such systems may be cumbersome and time consuming. Also other authors dedicated their attention to the actual ease of programming of mobile devices, and some focused specifically on the audio subsystem [19], [21].

Using mobile devices for assessing environmental noise levels poses several technical challenges that are only partially addressed in the above described efforts. In the remainder of this section we provide a list of properties an application for capturing noise levels should be endowed with, along with suggestions and ideas on how to enforce such properties.

Context-Awareness. Information about the context of the user should be collected and used to trigger data collection. Indeed, the phone should measure noise levels only when this makes sense in the actual location and status of the user, e.g., if she is visiting a neighborhood whose "loudness" could be of interest for her or other users or applications. The particular context of the mobile device also influences the decision about wether to start a measurement or not. For instance, it would be of little value to perform a measurement while the phone is in a pocket or in a bag. In these cases, however, using microphones of Bluetooth- or cable-based headsets may represent an alternative solution. Context recognition techniques able to classify the status of both the user and her mobile phone are therefore needed and could ideally involve several local sensors [22].

Unobtrusiveness. Access to the resources of the mobile phone should occur in the background, possibly without requiring the user to perform any action to participate in the sensing task. An unobtrusive usage of the sensing resources of the mobile phones includes a thrifty access to hardware resources (computation, communication, batteries) and the availability of adequate primitives that allow the user to set her privacy settings. In the case of audio measurements, for instance, we often hit upon users mistrusting the application and fearing for their conversations to be recorded without their consensus. Besides supporting privacy issues from the technical side (e.g., by computing noise levels on the phone and transmitting only the averaged dB values to the back-end server), providing adequate incentives for participation in the sensing task may help reducing or redirecting users concerns.

Correctness. The accuracy of the measured value should be estimated and logged along with the value itself. Audio input channels of mobile phones typically feature noise canceling or low-pass filters and/or dynamic input level adjustment, which partially hamper the possibility of measuring the actual absolute loudness of a received acoustic signal. Exploiting available programming primitives to bypass filtering of the audio signal is the first, fundamental step towards more precise mobile phone-based noise level measurements. Indeed, although it is to expect that common mobile phones won't be able to reach the accuracy of dedicated sound level meters, it is important to ascertain if the obtainable accuracy is sufficient to allow the envisioned applications to work reliably. In many cases, for instance for the identification of quiet bike trails [5], it would be sufficient to infer discrete states from the raw measurements (e.g., *quiet*, *moderately loud* or *very loud*). However, inadequacy of the hardware or not disengageable processing stages may hamper even this possibility, as we will also demonstrate trough our experimental study in the following section IV. Last but not least, since microphones of mobile phones are obviously not intended for noise measurements, calibration issues arise, as in the case of wireless sensor nodes. We will investigate programming and audio processing issues in more details in the following section IV.

Energy-Awareness. To preserve the battery of the mobile phone, the amount of energy used to measure noise levels should be kept as low as possible. Clearly, the energy spent for sensing heavily depends on the duration of the measurements. Since acoustic signals usually exhibit quick and wide fluctuations, noise levels are computed as long-term averages of the (opportunely time- and frequency-weighted) mean square acoustic pressure [2], [23], [24]. Although short-term averages may be valuable too, an averaging period of at least few seconds should always be guaranteed. Additionally, within this period the measuring devices should be let in a stable position. Energy awareness should be enforced in the system by, for instance, make the duration of a measurement depend upon the available battery charge.

IV. Experimental results

To understand the feasibility of mobile phones to be used as noise pollution sensors, we performed a series of experiments using several different test signals, and carefully analyzed the collected data. Our experiments aim at investigating two main issues, namely, the comparableness of the acoustic measurements of nearby located phones and the accuracy of such measurements against those taken by a reference sound level meter.

As test devices, we used three Nokia N95 8GB mobile phones[3], to which we will refer as *Phone1*, *Phone2* and *Phone3*. The Nokia N95 8GB well represents state-of-the-art mobile devices, featuring remarkable processing power, good permanent storage capability and several built-in sensors, like an accelerometer and a GPS-receiver. We emphasize here that the availability of a GPS-module is an essential feature for devices serving mobile sensing applications, since it allows to associate a sensor value with the correspondent location at which the measurement took place. Furthermore, recent work demonstrated that accurate indoor-localization is doable by opportunely processing signals received by standard Bluetooth and WLAN modules [25], both being available on the Nokia N95 8GB devices.

We used three devices of the same type to avoid as far as possible discrepancies in the responses of the mobile phones being due to differences in the built-in hardware. Indeed, even devices of the same type may be built using chips from different manufactures, but their overall performance remain congruent still in this case. Although we experimented on devices of a specific manufacturer (Nokia) and type (N95), our conclusions qualitatively apply also to most commercially available mobile phone platforms, since these usually exhibit comparable characteristics with respect to their audio circuitry.

In our experiments, we stimulated the mobile devices with a series of different acoustic signals and recorded the responses of the phones for offline analysis. To capture such responses, we implemented two recording applications, one in Python (PyS60) and the other in Java (J2ME), which simply capture audio signals and store them in a *wave* file. Both the Python and the Java API expose simple methods for recording audio data but while the first only allows to capture signals at a rate of 8kHz, the latter enables sampling rates of up to 41kHz. Lower level primitives granting direct access to the unprocessed raw audio data are in both cases still unavailable. This implies that before we can record them, the audio signals likely undergo the processing steps typical for voice communication applications, like band-pass filtering, noise canceling, or input level adjustment. This "polishing" of the signal, if not properly bypassed, constitutes a serious burden to the use of mobile phones for the assessment of environmental noise levels, as we will show later in this section. The third programming option for the Nokia mobile devices under consideration is C++ Symbian. The correspondent API offers methods for setting low-level audio parameters like the microphone gain, as well as for defining custom codecs or selecting the specific audio source (built-in microphone, line-in, phone call or radio). To the best of our knowledge, however, C++ Symbian does not expose methods for immediate access to the raw audio data, as also underlined in [19]. Considering the complexity related to writing application in C++ Symbian, we limited our investigations to Python and Java, and let additional implementations to future work.

In all our experiments, the Extech 407740 class 2 professional sound level meter, connected as sketched in section II, has been co-located with the three mobile devices and collected ground-truth noise level measurements. We set the phonometer to gather data using F-weighting in time and A-weighting in frequency, as recommended in [24]. F-weighting basically runs a moving average on the squared acoustic pressure using a time constant of 125 ms, while the A-weighting attenuates very low and very high frequencies to resemble the natural filtering behavior of the human ear [2], [23]. We correspondingly applied F- and A-weighting filters also to the responses of the mobile phones and indicated the resulting A-weighted noise levels in dBA (A-weighted decibels). We should notice at this point that due to mechanical or thermal shocks even professional sound level meters may get out of calibration and therefore return erroneous measurements. During our experiments, we let the phonometer rigourously untouched across and during measurements, so as to reduce errors due to calibration drifts.

To reproduce the test signals we used a common laptop supporting up to 192kHz audio in-/output that we opportunely connected to high quality external speakers. In the following, we will refer to this system as the "audio source".

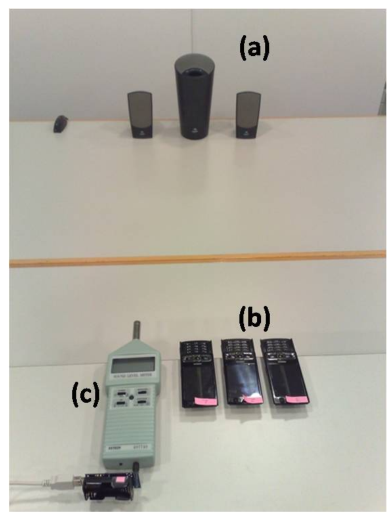


Fig. 1. Experimental setup: **(a)** the audio source; **(b)** the three mobile phones; **(c)** the phonometer.

A. *Response to the* **tones** *test signal*

Our first experiment aimed at studying the response of the mobile phones to synthetic test signals produced by an audio source in a controlled environment. To this scope, we aligned the three mobile devices and the phonometer on a surface in a silent room[4], all at approximately the same distance from the audio source, as shown in figure 1. In this setting, we

[3]www.forum.nokia.com/devices/N95_8GB

[4]When the audio source is off, the average noise level in this room, measured with our reference phonometer, is slightly above 30dBA.

reproduced three times a one minute long test signal and recorded responses of the phonometer and the three mobile phones. The test signal starts with a five seconds long white noise snippet, followed by five seconds of *silence* (i.e., the audio sources is on but outputs a zero-amplitude signal). After this first phase, the test signal repeats five times a five seconds long 1 kHz sinusoidal tone, whose amplitude is regularly increased at each repetition (varying from 20% to 100% of the total available output dynamic). The tones are interleaved with five seconds long silence cuts. We use a pure tone as test signal since standard calibrators calibrate sound level meters at one single frequency, namely 1 kHz. Since the signal is a repetition of sine tones, we will refer to it as the *tones* test signal.

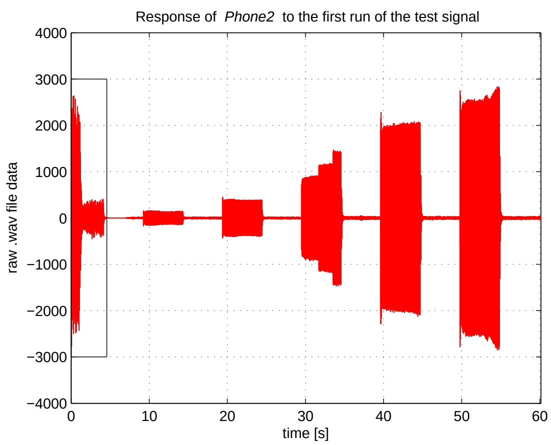


Fig. 2. Response of *Phone1* to the first run of the *tones* test signal.

Figure 2 displays an example of the recorded instantaneous acoustic pressure levels correspondent to a single run of the *tones* test signal. As we can see, the mobile phone properly mirrors the increasing amplitude of the five sinusoidal tones. Instead, the response of the mobile phone to the white noise signal, framed in a black rectangle in figure 2, clearly shows the effect of a noise canceling filter. After recording the white noise signal for about one-third of the five seconds snippet, the filter classifies the signal as background noise and suppresses it, causing a reduction in the amplitude of the recorded signal, and, consequently, a diminution of the "perceived" loudness associated with the signal. Noise canceling is a standard signal processing technique used in several audio applications and it is therefore not a surprise to ascertain its effect in this measurement. However, we aimed at outlining that the action of noise canceling can hamper the possibility to use mobile phones for the measurement of environmental noise levels, since it may partially suppress the signal one is actually trying to capture. As mentioned above, the availability of adequate programming primitives granting access to the raw audio data would allow to bypass this problem, but this is still not possible with the currently available APIs.

The noise levels corresponding to the recorded acoustic pressure levels, properly A- and F-weighted and averaged over

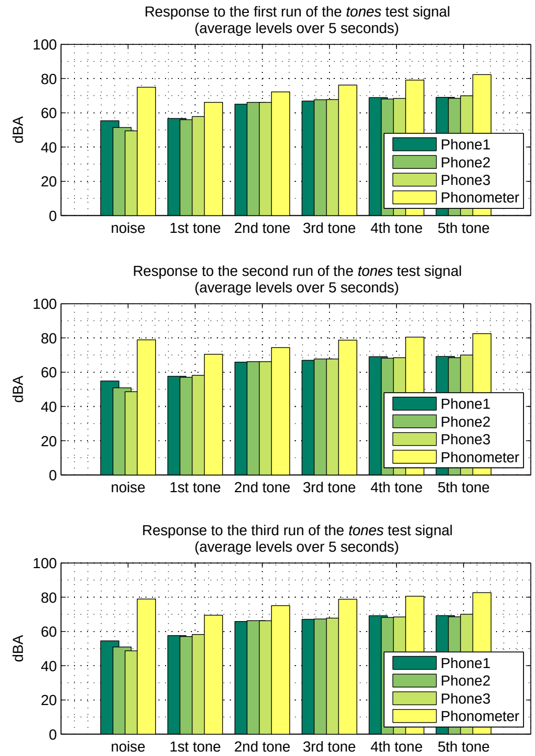


Fig. 3. Response to the three runs of the *tones* test signal, for the three devices under test *Phone1*, *Phone2* and *Phone3* and the reference phonometer.

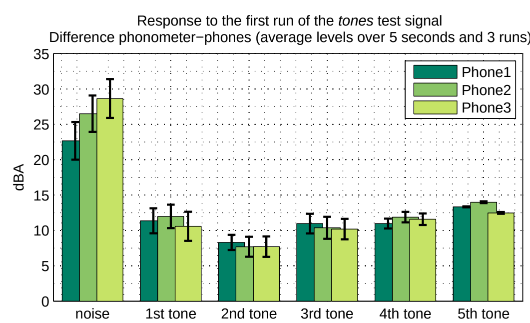


Fig. 4. Response to the first run of the *tones* test signal. Comparison of the difference of the levels measured by the phonometer and the correspondent levels captured by *Phone1*, *Phone2* and *Phone3*. The difference are averaged over the three runs of the test signal.

five seconds, are reported in figure 3, along with the noise levels measured by the phonometer. Figure 4 reports the differences between the measurements of the phonometer and those of the three phones for the first run of the tones *test* signal. Both figures 2 and 4 make evident that, when the test signal is white noise, there is a the high discrepancy ($>$ 20dBA) between the noise level measured by the phonometer and those measured by the phones. This discrepancy is consistently lower, although still high ($\sim$ 10dB), for all the five subsequent measurements and for all the three test devices. A constant divergence from the reference measurement would represent a correction factor that could be used for calibrating the mobile phones against the reference itself. However, our experiments showed that this value may change of several (> 5) dBAs, when the measurement is repeated in the same setting but at a different time.

B. *Response to the* **chirp** *test signal*

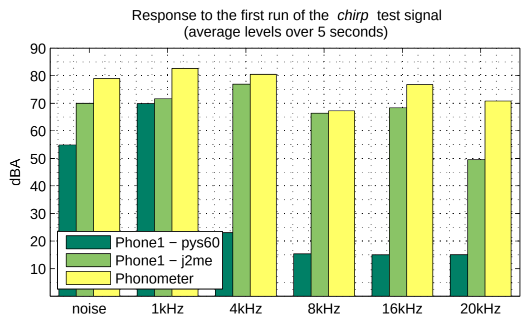


Fig. 5. Equivalent noise levels measured by the phonometer and by *Phone1* (using both the PyS60 and J2ME applications). The levels are averaged over the 5 seconds duration of each tone of the *chirp* test signal.

To investigate the ability of the mobile phones to capture signals at different frequencies, we performed a second experiment using a test signal consisting of five sine waves of equal amplitude whose frequencies increments from 1 to 4, 8, 16 and 20kHz. Since a signal whose frequency content increases linearly with time is typically called a *chirp*, we refer to this sequence of sine tones as the *chirp* test signal. Letting unchanged the experimental setting, we played the sixty seconds long test signal three times in a row, and recorded the responses for offline analysis. Figure 5 shows the average noise levels measured during the first run of the *chirp* test signal, using both the PyS60 and the J2ME applications and along with the correspondent values recorded by the phonometer. These levels result from averaging the F- and A-weighted squared instantaneous acoustic pressure over the five seconds long snippets of the *chirp* test signal. As we can see, the audio signal captured by the PyS60 application is clearly low-passed, since already the response to the 4kHz tone is so feeble that it approaches the value measured when no test signal is present. This results doesn't come as a surprise, since for voice transmission applications a low-pass filtering at a frequency around 4 kHz is a standard figure. However, since the human hear can actually hear frequencies far above 4 kHz [23], bypassing this low-pass filter is mandatory if the mobile device is intended to be used as noise level meter. The J2ME programming primitives clearly allow to access richer audio input data, since our J2ME application manages to capture, although with low accuracy, also the tone at 20 kHz, as shown again in figure 5. This superior recording quality, however, comes at the cost of higher processing and storage loads.

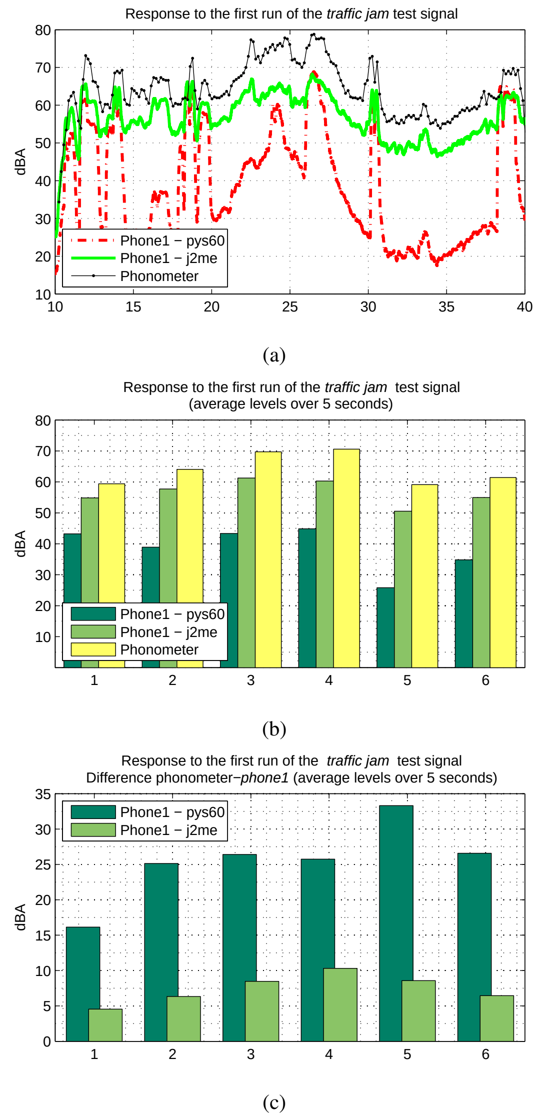
C. *Response to the* **traffic jam** *test signal*

To evaluate the response of the mobile devices in more realistic conditions, we performed a third experiment using a thirty seconds long recording of a traffic jam. The signal features several different honks, breaks and engine noises, resulting in complex frequency and amplitude spectra. The experimental setting is the same as in the first two experiments and the test signal is again played three times in a row. Since the responses to the three subsequent runs of the test signal are very similar to each other, we report, in figure 6, only the results ascertained in the first run. Since the PyS60 and the J2ME applications cannot run concurrently on the mobile phone, we performed the test twice, recording once with the PyS60 and once with the J2ME application. As expected, since the test signal and the experimental setting did not change at all, the phonometer profiles collected during the two subsequent experiments overlap almost perfectly. For this reason, in commenting figure 6 we can indistinguishably refer to *the* phonometer profile. A visual inspection of figure 6(a), demonstrates the superior fidelity of the J2ME-profile to the actual ground-truth measurements collected by the phonometer. The PyS60-profile, on the contrary, diverges consistently from the phonometer profile, as also the average noise levels reported in figure 6(b) show. The difference between the average levels recorded by the phonometer and those captured by *Phone1* are also reported in figure 6(c). This graph evidences the higher accuracy reachable using the J2ME application but also shows that the difference between the noise levels measured by the phones and those measured by the phonometer varies significantly across different time sectors of the signal.

The experimental results presented in this section allow to make some general qualitative assertions on the feasibility of mobile phones to be used as noise pollution sensors. First, devices of the same type do return coherent audio responses so that a direct comparison of their readings is, at least qualitatively, possible. Second, the noise levels measured by mobile phones may diverge from those captured by a co-located professional sound level meter and the magnitude of this divergence may vary significantly over time, depending on the actual audio signals. Moreover, the accuracy of the measured noise levels is influenced by the processing steps the audio signal undergoes before being recorded and by the sampling rate at which the signal is captured. Further, the specific language used to program the devices may influence the accuracy of the measured noise levels, since different APIs may expose audio data at different "rawness" levels.

V. Conclusions

In this paper, we reported qualitative considerations, an overview of related work and a collection of experimental data that demonstrate the complexity hiding behind the intriguing idea of using mobile phones to unobtrusively and ubiquitously capture noise pollution data.



(a)

(b)

(c)

Fig. 6. Response to the first run of the *traffic jam* test signal.

The reported experimental evidences show that the assessment of noise pollution levels using mobile phones still poses several challenges. For instance, bypassing the audio signal processing modules of the mobile phones (noise canceling, low-pass filtering) is still unfeasible even on state-of-the art mobile devices. Furthermore, we outlined several additional factors that may hamper reliable collection of noise levels. Among these, we mentioned the need for determining the context of both the mobile phone and the user carrying it to ascertain weather a measurement should be taken or not and the problem of fixing an appropriate value for the duration of the measurements. Possible users' concerns about audio data being recorded "in the background" by their mobile phone should also be considered in the design of a system for noise monitoring.

ACKNOWLEDGMENTS

The work presented in this paper has been partially supported by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

REFERENCES

[1] European Commission, "Green Paper on Future Noise Policy," http://ec.europa.eu/environment/noise/greenpap.htm, COM (96) 540 final, November 1996.

[2] S. Santini, B. Ostermaier, and A. Vitaletti, "First Experiences Using Wireless Sensor Networks for Noise Pollution Monitoring," in *Proceedings of the 3rd ACM Workshop on Real-World Wireless Sensor Networks (REALWSN'08)*, Glasgow, United Kingdom, April 2008.

[3] Moteiv Corporation (now Sentilla), www.sentilla.com.

[4] E. Kanjo, S. Benford, M. Paxton, A. Chamberlain, D. S. Fraser, D. Woodgate, D. Crellin, and A. Woolard., "MobGeoSen: Facilitating Personal GeoSensor Data Collection and Visualization using Mobile Phones," *Personal Ubiquitous Computing Journal*, vol. 12, no. 8, pp. 599–607, November 2008, published online: 8. August 2007.

[5] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson, "People-Centric Urban Sensing," in *Proceedings of the 2nd Annual International Wireless Internet Conference (WICON'06)*, Boston, MA, United States, August.

[6] "Directive 2002/49/EC of the European Parliament and of the Council of 25 June 2002 relating to the Assessment and Management of Environmental Noise," Official Journal of the European Communities, July 2002.

[7] Coordination for European Research for Advanced Transport Noise Mitigation (CALM) - A Coordination Action funded by the 6th Framework Programme of the European Community, "BlueBook - Research on Environmental Noise," Available online at http://www.calm-network.com/bluebook/content/index.htm (Accessed on January 13th, 2009), April 2006.

[8] European Commission Working Group Assessment of Exposure to Noise (WG-AEN), "Good Practice Guide for Strategic Noise Mapping and the Production of Associated Data on Noise Exposure," January 2006.

[9] L. Filipponi, S. Santini, and A. Vitaletti, "Data Collection in Wireless Sensor Networks for Noise Pollution Monitoring," in *Proceedings of the 4th IEEE Intl. Conference on Distributed Computing in Sensor Systems (DCOSS '08)*, Santorini Island, Greece, June 11–14 2008.

[10] EasySen LLC, www.easysen.com.

[11] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," in *Proceedings of the 5th annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom '99)*, Seattle, Washington, United States, 1999, pp. 263 – 270.

[12] Y. Yao and J. Gehrke, "The Cougar Approach to In-Network Query Processing in Sensor Networks," *SIGMOD Record*, vol. 31, no. 3, pp. 9 – 18, September 2002.

[13] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "The BikeNet Mobile Sensing System for Cyclist Experience Mapping," in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, Sydney, Australia, 2007, pp. 87 – 101.

[14] B. Greenstein, C. Mar, A. Pesterev, S. Farshchi, E. Kohler, J. Judy, and D. Estrin, "Capturing High-Frequency Phenomena Using a Bandwidth-Limited Sensor Network," in *Proceedings of the 4th ACM International Conference on Embedded Networked Sensor Systems (SenSys'06)*, Boulder, Colorado, USA, November 1–3 2006.

[15] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring Volcanic Eruptions with a Wireless Sensor Network," in *Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN05)*, Instanbul, Turkey, January 2005.

[16] V. Trifa, A. N. G. Kirschel, C. E. Taylor, and E. E. Vallejo, "Automated Species Recognition of Antbirds in a Mexican Rainforest using Hidden Markov Models," *Journal of the Acoustical Society of America*, vol. 123, no. 4, pp. 2424–2431, April 2008.

[17] J. J. Ding, S.-Y. Cheung, C.-W. Tan, and P. Varaiya, "Signal Processing of Sensor Node Data for Vehicle Detection," in *Proceedings of the IEEE Intelligent Transportation Systems Conference*, Washington DC, USA, October 2004.

[18] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: Rich Monitoring of Road and Traffic Conditions Using Mobile Smartphones," in *Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems (SenSys 2008)*. ACM, 2008, pp. 323–336.

[19] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing Meets Mobile Social Networks: the Design, Implementation and Evaluation of the CenceMe Application," in *Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems (SenSys 2008)*. New York, NY, USA: ACM, 2008, pp. 337–350.

[20] A. Misra, G. Essl, and M. Rohs, "Microphone as Sensor in Mobile Phone Performance," in *Proceedings of the 8th International Conference on New Interfaces for Musical Expression (NIME 2008)*, Genova, Italy, June 5-7 2008.

[21] K. Leichtenstern, A. D. Luca, and E. Rukzio, "Analysis of Built-In Mobile Phone Sensors for Supporting Interactions With the Real World," Whitepaper, University of Munich, Februar 2006.

[22] K. V. Laerhoven, A. Schmidt, and H. werner Gellersen, "Multi-Sensor Context Aware Clothing," in *in Proceedings of the Sixth International Symposium on Wearable Computers, 2002*. IEEE Press, 2002, pp. 49–56.

[23] D. A. Bies and C. H. Hansen, *Engineering Noise Control: Theory and Practice.*, 3rd ed. London and New York: Spon Press (Taylor &Francis Group), 2003.

[24] International Organization for Standardization, "Acoustics - Description, Measurement and Assessment of Environmental Noise - Part 1: Basic Quantities and Assessment Procedures," 2007.

[25] P. Bolliger, "Redpin - Adaptive, Zero-Configuration Indoor Localization through User Collaboration," in *Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environment Computing and Communication Systems*, San Francisco, USA, September 2008.