

Ear-Phone: A context-aware End-to-End Participatory Urban Noise Mapping System

Rajib Rana¹ Chun Tug Chou¹ Wen Hu² Nirupama Bulusu³ Salil Kanhere¹

¹ University of New South Wales, Australia
{rajibr, ctchou, salilk}@cse.unsw.edu.au

² CSIRO ICT Centre, Australia
wen.hu@csiro.au

³ Portland State University, USA
nbulusu@cs.pdx.edu

Technical Report
UNSW-CSE-TR-1015
June 2010

THE UNIVERSITY OF
NEW SOUTH WALES



School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia

Abstract

A noise map facilitates monitoring of environmental noise pollution in urban areas. It can raise citizen awareness of noise pollution levels, and aid in the development of mitigation strategies to cope with the adverse effects. However, state-of-the-art techniques for rendering noise maps in urban areas are expensive and rarely updated (months or even years), as they rely on population and traffic models rather than on real data. Participatory urban sensing can be leveraged to create an open and inexpensive platform for rendering up-to-date noise maps.

In this paper, we present the design, implementation and performance evaluation of an *end-to-end* participatory urban noise mapping system called Ear-Phone. Ear-Phone, for the first time, leverages *Compressive Sensing* to address the fundamental problem of recovering the noise map from incomplete and random samples obtained by crowdsourcing data collection. Ear-Phone, implemented on Nokia N95, N97 and HP iPAQ mobile devices, also addresses the challenge of collecting accurate noise pollution readings at a mobile device. Ear-Phone also leverages context aware sensing and we study the impact of using data from different contexts upon noise map reconstruction. Extensive simulations and outdoor experiments demonstrate that Ear-Phone is a feasible platform to assess noise pollution, incurring reasonable system resource consumption at mobile devices and providing high reconstruction accuracy of the noise map.

1 Introduction

At present, a large number of people around the world are exposed to high levels of noise pollution, which can cause serious illnesses ranging from hearing impairment to negatively influencing productivity and social behavior [14]. As an abatement strategy, a number of countries, such as the United Kingdom [10] and Germany [11], have started monitoring noise pollution. They typically use a noise map (a visual representation of the noise level of an area) to assess noise pollution levels. The noise map is computed using simulations based on inputs such as traffic flow data, road or rail type, and vehicle type. Since the collection of such input data is very expensive, these maps can be updated only after a long period of time (e.g. 5 years for UK [10]). To alleviate this problem, a recent study [22] proposes the deployment of wireless sensor networks to monitor noise pollution. Wireless sensor networks can certainly eliminate the requirements of sending acoustic engineers for taking real measurements, but the deployment cost of a dedicated sensor network in a large urban space will also be prohibitively expensive.

In this paper, we instead propose an urban sensing approach (also known in the literature as participatory sensing [6], people-centric sensing [13] or community sensing [17]) for monitoring environmental noise, especially roadside ambient noise. The key idea in participatory sensing is to “crowdsource” the collection of environmental data in urban spaces to people, who carry smart phones equipped with sensors and location-providing Global Positioning System (GPS) receivers. The vision of participatory sensing is inspired by the success of other online participatory systems, such as Wikipedia, online reputation systems, and human computation systems such as the Google Image Labeler. Due to the ubiquity of mobile phones, the proposed approach can offer a large spatial-temporal sensing coverage at a small cost. Therefore, a noise map based on participatory data collection can be updated with a very small latency such as hours or days compared to months or years, making information provided by such a noise map significantly more current than that provided by traditional approaches.

It is non-trivial to build a noise pollution monitoring system based on mobile phones. Mobile phones are intended for communication, rather than for acoustic signal processing.¹ To be credible, noise pollution data collected on mobile phones should be comparable in accuracy to commercial sound level meters used to measure noise pollution. Since a participatory noise monitoring system relies on volunteers contributing noise pollution measurements, these measurements can only come from the place and time where the volunteers are present. Furthermore, volunteers may prioritize the use of the microphone on their mobile phones for conversation. Or they may choose to collect data only when the phone has sufficient energy. Consequently, samples collected from mobile phones are typically randomly distributed in space and time, and are incomplete. To develop a useful noise pollution monitoring application, we need to recover the noise map from *random* and *incomplete* samples obtained via crowdsourcing.

It is unrealistic to expect that volunteers will always carry the phones in their hand, with the microphones correctly exposed for sampling ambient noise. Research conducted by Nokia [9] suggests that other popular choices include trouser’s pockets, bags and belt case. Since, volunteers may contribute samples

This work is supported by ARC Discovery grant DP0770523. N. Bulusu is also supported by NSF grant 0747442.

¹For example, devices such as the Nokia N95 or HP iPAQ do not support floating-point arithmetic, which must be emulated with fixed point operations.

when the phone is in any of these (or other) positions, it is necessary to investigate if the phone location has a noticeable impact of the equivalent noise level recorded by the phone. In this paper, we address these challenges. Our main contributions are:

1. We present the design and implementation of an *end-to-end* noise mapping system, called Ear-Phone, to generate the noise map of an area using participatory urban sensing. Ear-Phone consists of mobile phones and a central server. It encompasses signal processing software to measure noise pollution at the mobile phone, as well as signal reconstruction software at the central server. This new noise mapping system is expected to cost significantly less than traditional noise monitoring systems.
2. We address the problem of incomplete samples that are obtained via crowdsourcing by using *compressive sensing*, focusing on roadside noise pollution.² To the best of our knowledge, this is the first application of compressive sensing to environmental noise data collection.
3. We evaluate Ear-Phone with extensive simulations and real-world outdoor experiments. The results show that Ear-Phone has reasonable accuracy, and resource requirements in terms of CPU load and energy consumption.
4. We demonstrate that the deviation in equivalent noise level is quite high when the phone is in belt case or in bag. On the other hand when it is carried inside trouser’s pocket, the deviation is insignificant. We therefore want to stop recording noise level when the phone is inside belt case or inside bag. We develop a classification algorithm which using input from proximity, rotation and orientation sensor and from GPS receiver can classify the location of the phone. This classification algorithm runs on the phone and ceases recording noise level when the phone is in either belt case or in bag. We show that the mean accuracy of our classification algorithm is approximately 92%..

We also determine the impact of fraction of time the phone is carried inside pocket. Experimental results show that when using small number of data points, a large fraction of time inside pocket causes significant reconstruction error. However, with the increase of data points (>50%), impact of pocket enclosure becomes negligible.

The rest of the paper is organized as follows. In the next section, we describe the Ear-Phone architecture followed by the system design in Section 3. Then, we evaluate Ear-Phone with both outdoor experiments (Section 4) and extensive simulations (Section 5). We present related work in Section 6 and conclude in Section 7.

2 Ear-Phone Architecture

In this section, we provide an overview of Ear-Phone. A detailed description of the system components is presented in Section 3.

The overall Ear-Phone architecture, depicted in Fig. 2.1 consists of a mobile phone component and a central server component. Noise levels are assessed on the mobile phones before being transmitted to the central server. The central server reconstructs the noise map based on the partial noise measurements. Note

²We focus on roads because typically noise pollution is most severe on busy roads.

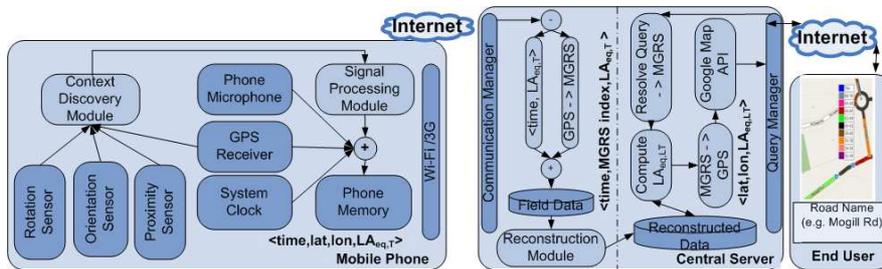


Figure 2.1: Ear-Phone Architecture.

that reconstruction is required because the urban sensing framework cannot guarantee that noise measurements are available at all times and locations.

Let us begin with a mobile phone user who is walking along a street. We call a mobile phone with the Ear-Phone application a MobSLM, where SLM stands for “sound level meter” which is the instrument used by acoustic engineers to measure environmental noise level. When the mobile phone is not used for conversation the MobSLM on the phone is turned on.¹ When turned on, the context discovery module first discovers the context of the phone and decides to trigger the signal processing module. Note that we record noise level from only a subset of contexts (here we refer hand, pocket, bag and belt case as context) due to higher noise added from other contexts. When triggered the signal processing module starts computing a loudness characteristic known as the equivalent noise level ($LA_{eq,T}$) over a time interval T from the raw acoustic samples collected by the microphone over the corresponding time interval. The computed noise level is further tagged with the GPS coordinates (which will be denoted by (lat,lon) and system time before being stored in the phone memory. The stored records $\langle time,lat,lon,LA_{eq,T} \rangle$ are uploaded to the central server when the mobile phone detects an open WiFi access point. Of course, 3G services on mobile phones can also be used to upload data.

The communication manager at the central server waits for user transmissions. When it receives user data, it converts the GPS coordinates of a record to a Military Grid Reference System (MGRS, see Section 3.3 for the detailed description) grid index and stores the information $\langle time, grid index, LA_{eq,T} \rangle$ in a data repository. Reconstruction is conducted at (predefined) periodic intervals²; when triggered, the reconstruction module is invoked to reconstruct the missing data. The reconstructed data is then stored in the data repository.

A query from an end user (e.g., what is the noise level on Oxford Street at 5pm on 28 October 2009?) is processed by a query manager at the central server. The location information (e.g., Oxford Street) of the query is first resolved into grid indices and the reconstructed data associated with those grid indices is fetched from the data repository. Then, the grid indices are converted back to GPS coordinates and the corresponding noise levels are overlaid on a geo-centric Internet map before being displayed to the end user.

¹Note that in the current prototype deployment we have not implemented this feature. During our experiments we did not use the phone for conversation.

²Note that in this paper we primarily focus on the accuracy of the noise map obtained from participatory sensing. Determination of a suitable update interval is left for future work.

Table 3.1: Coefficient of the digital filter that approximates A-weighting.

l	0	1	2	3	4	5	6	7	8	9	10
b_l	0.9299	-2.1889	0.7541	1.3229	-0.7728	0.1025	-0.2398	-0.0098	0.1154	-0.0103	-0.0033
a_l		2.1856	-0.7403	-1.0831	0.6863	-0.2274	0.2507	-0.0058	-0.0821	0.0153	0.0004

3 System Components

In this section, we describe the major components of Ear-Phone in detail.

3.1 Mobile Phone Components

Signal Processing Module

The aim of the signal processing module is to quantitatively assess the environmental noise. Noise level or loudness is typically measured as the A-weighted equivalent continuous sound level or $LA_{eq,T}$. A-weighting is the commonly used frequency weighting that reflects the loudness perceived by human being [16]. Measured in decibel (dBA), $LA_{eq,T}$ captures the A-weighted sound pressure level of a constant noise source over the time interval T , which has the same acoustic energy as the actual varying sound pressure level over the same interval. Note that sound pressure level is captured by a microphone as an induced voltage. The A-weighted equivalent sound level $LA_{eq,T}$ in time interval T is thus given by

$$LA_{eq,T} = 10 \log_{10} \left(\underbrace{\frac{1}{T} \int_0^T (v_A(t))^2 dt}_{\bar{v}_A(T)} \right) + \Delta \quad (3.1)$$

where $v_A(t)$ is the result of passing the induced voltage $v(t)$ through an A-weighting filter and Δ is a constant offset determined by calibrating the microphone against a standard sound level meter.

In order to compute $\bar{v}_A(T)$, we design a tenth-order digital filter (whose coefficients are given in Table 3.1) whose frequency response matches with that of A weighting over the range 0–8kHz. This range is chosen because the acoustic standard, IEC651 Type 2 SLM [16], requires measurement of environmental noises between 0 and 8 kHz. Based on the coefficients of the digital filter (a_l, b_l where $l = 1..10$), we then calculate $\bar{v}_A(T)$ using Algorithm 1.

3.2 Context Identification Module

Note that the usability of the sound data collected from the mobile phone depends on the context (i.e., where is the phone carried) of the phone. Mobile phones can be in numerous contexts. An extensive survey [9] conducted by Nokia which queried people from over 11 different countries shows that a large percentage of people tend to carry phone in hands (palm) since it facilitates the interaction with the phone. The study also shows that vast majority of male tends to carry phone in their trouser’s pocket whereas carrying the phone inside the bag is a popular choice of female. Furthermore, carrying the phone in a belt case is also a popular choice, particularly among males. We therefore consider palm, pocket, bag and belt case as possible four contexts.

In Section 4, we present results from empirical measurements to highlight the impact of these 4 contexts on the measured noise samples. As expected, we find that the best quality data can be recorded when the phone is held in the

Initialize: $Q = F_s T - 1$, F_s = Sampling Frequency, Sampling Period
 $T_s = \frac{1}{F_s}$
Input: Voltage samples $v(kT_s)$ for $k = 0, 1, 2, \dots, Q - 1$ over duration $[0, T]$
Output: $\bar{v}_A(T)$
Based on $\{a_l, b_l\}$ and initial condition, $v_A(kT_s) = 0$ for $k = 0, \dots, 9$, recursively compute

$$v_A(kT_s) = \sum_{\ell=1}^{10} a_\ell v_A((k-\ell)T_s) + \sum_{\ell=0}^{10} b_\ell v((k-\ell)T_s) \text{ for } k \geq 10 \quad (3.2)$$

Compute

$$\bar{v}_A(T) = \frac{1}{Q} \sum_{k=0}^{Q-1} v_A(kT_s)^2 \quad (3.3)$$

Algorithm 1: Compute $\bar{v}_A(T)$.

hand. The equivalent noise level recorded when the phone is in the belt case and the bag has a significantly larger deviation from the ground truth and is therefore not useful. However, the noise samples recorded when the phone is in the pocket, is quite similar to the ground truth and thus can be useful.

We therefore focus on finding whether the phone is in one of the 3 following states - (i) palm (ii) pocket and (iii) bag or belt case. In the context identification module we use a combination of the proximity sensor, rotation sensor, orientation sensor and GPS receiver (all of which are embedded in a Nokia N97 phone) to identify the context. Fig. 3.1 shows the flow of the context identification process. Note that we introduce a two-step classification process that can conserve energy by not having to activate the other sensors (rotation, speed and orientation) when the phone is known to be in the hand.

Classifier 1

Using proximity sensor we primarily identify whether the phone is in open space or in closed space, i.e., whether the phone is held inside palm or in other three contexts. Note that the proximity sensor on N97 uses infrared proximity switches which work by sending out beams of invisible infrared light and then analyzing the reflection. Possible values of proximito state are EProximityUndefined, EProximityIndiscernible and EProximityDiscernible, where EProximityUndefined is returned when the proximity cannot be determined, EProximityDiscernible is returned when the sensor is triggered and EProximityIndiscernible is returned otherwise. Due to placement of the proximity sensor (see Fig. 3.3(b)) on the very top left corner of the phone, it is not typically tripped while holding the phone in hand, whereas for bag, pocket and belt case enclosures, the proximity sensor is tripped. Proximity states registered by the phone while kept in different contexts are summarized in Table 3.2.

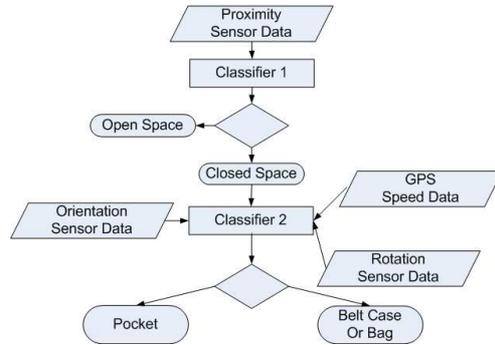


Figure 3.1: Process flow of context identification.

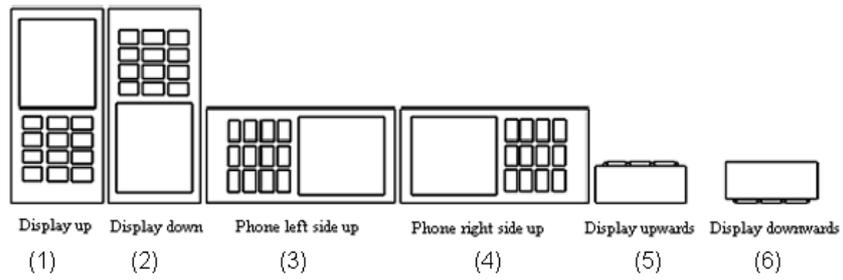
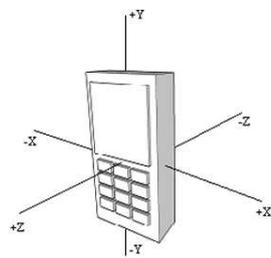
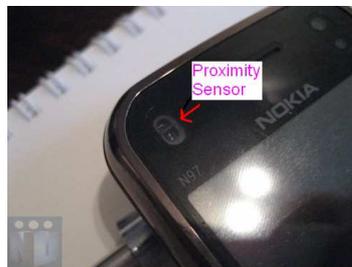


Figure 3.2: Phone orientation.



(a) Phone axes



(b) Proximity Sensor

Table 3.2: Classification table for Classifier 1.

Context	Proximito State
Palm	EProximityIndiscernible
Pocket	EProximityDiscernible
Bag	EProximityDiscernible
Belt Case	EProximityDiscernible

Classifier 2

If the phone is classified to be inside a closed space (i.e., either bag, belt case or pocket), we use the combination of orientation sensor and rotation sensor and GPS receiver (speed) to further classify whether the phone is inside pocket or in bag or belt case.

The rotation sensor registers rotation along the three different axes of the phone (see Fig. 3.3(a)). We observed that the variance of rotation is different when phone is attached to the different parts of the body i.e., shoulder (bag), waist (belt case) or thigh (trouser’s pocket) and it is not uniform along different axes.

The variance of rotation is also controlled by the orientation of the phone. Fig 3.2 shows the 6 possible orientations of the phone. Phones can be carried in any of these orientations. For example, belt cases may either hold the phone vertically or horizontally. In addition while carried vertically the phone can be either in display up (position 1) or display down (position 2) orientation and while carried horizontally it can be either in left side up (position 3) or right side up (position 4) orientation. The range of possibilities is similar when the phone is in the pocket. While carried inside bag the orientation can be quite complex if the phone is just thrown inside the bag. However, interviewing few female subjects we found that in order to make the phone easy accessible they typically keep the phone in small compartments inside the bag. Reviewing few ladies bags in the market it revealed that the compartments can hold the phones either vertically or horizontally. Therefore, we considered the first four orientations (positions 1,2,3 and 4) shown in Fig. 3.2 to be feasible for our application setting.

Our initial strategy was to find a <rotation-variance, Orientation > pair to detect if the phone is in the pocket or in bag or belt case). However, further investigation revealed that the rotation is also dependent on the speed of the person carrying the phones. Speed was registered by the GPS receiver on the phone.

In order to investigate the resultant impact of all three inputs (rotation, orientation, speed) we collected data from 10 subjects, where 7 subjects were male and 3 were female. Each of the subjects carried three phones in three different places: one in pocket (male carried phone in trouser’s pocket and female carried them in the skirt pocket), one in belt case and one in bag. They were asked to walk along a street while the phone was recording the rotation and speed, and storing it in phone memory. Each walk was of five minutes duration and after each walk the orientations were changed in all three places. In order to keep it simple, we used same orientation in all 3 places.

We plotted the variance of rotation against speed for different orientation of phone (Fig. 3.3 - 3.6). One can readily observe from these graphs that the data points along at least one axis are distinctly clustered for the two different positions (pocket and bag or belt case). For example, in Fig. 3.3-3.4 both y and z axis, in Fig. 3.5 both x and z axis and finally in Fig. 3.6, z axis.

Based on the observations we employed k-NN [12] to cluster the data set.

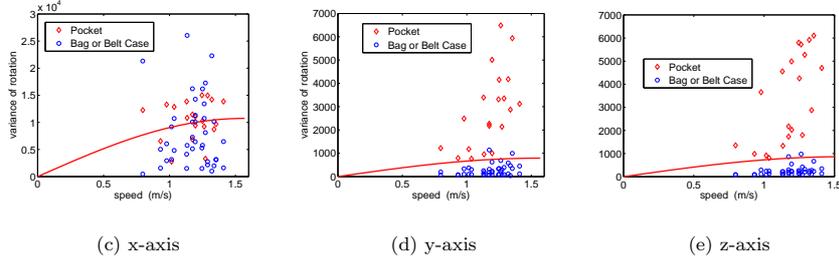


Figure 3.3: Right side up in pocket, belt case and bag.

Note that in k-NN a query or test point is classified by assigning the label which is most frequent among the k training samples nearest to that query point. The red curve shown in Fig. 3.3 to Fig. 3.6 is computed by the k-NN algorithm which indicates that if a query or test point lies to the right of the red line will be classified as bag or belt case enclosure and if lies to the left of the red line will be classified as pocket enclosure.

Window Size

Note that we compute variance of rotation over a time window. We also compute the average speed on that time window. We have investigated the impact of different window size on the classification accuracy. For every window size we determined the clusters using our training data set, then used the same data set to determine classification accuracy. Table 3.3 to 3.6 summarizes the result of impact of different window size on the classification accuracy for different orientation of the phone. We observed that a window size of 180 seconds would be sufficient, since it could produce more than 80% classification accuracy for all possible orientations of the phone.

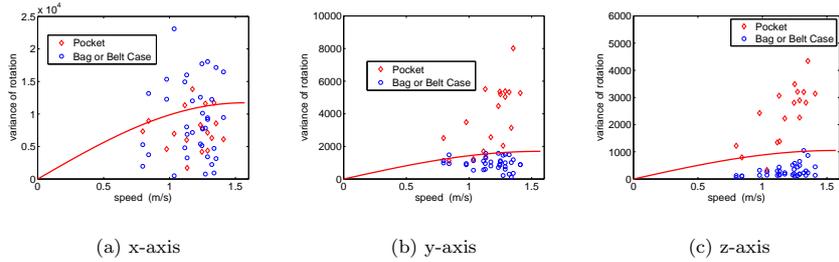


Figure 3.4: Left side up in pocket, belt case and bag.

3.3 Central Server Components

Computing Long-term Equivalent Noise Level, $LA_{eq,LT}$

In order to compute the long-term equivalent noise level $LA_{eq,LT}$ over the duration NT (where $N > 1$ and N is an integer) from the equivalent noise

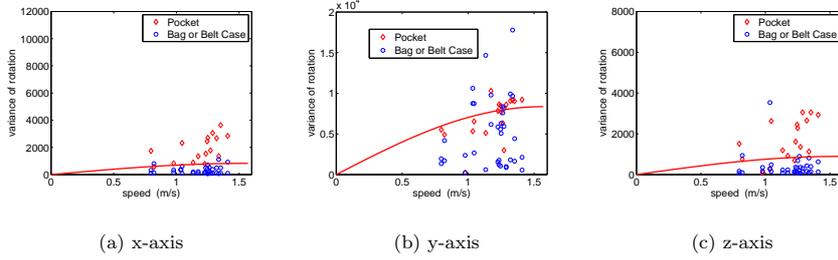


Figure 3.5: Display down in pocket, belt case and bag.

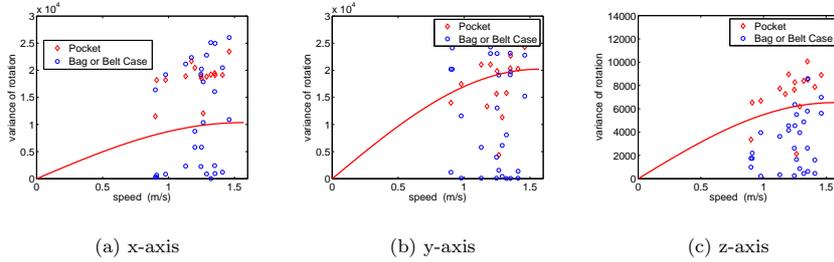


Figure 3.6: Display up in pocket, belt case and bag.

levels $LA_{eq,T}$ measured over shorter time durations T , we use the following standard formula:

$$LA_{eq,LT} = 10 \log_{10} \left[\frac{1}{N} \sum_{i=1}^N 10^{0.1 LA_{eq,T_i}} \right] \quad (3.4)$$

where N is the number of reference time intervals and LA_{eq,T_i} is the time average A-weighted sound pressure level in the i -th reference time interval. The above formula can be readily derived by noting that the equivalent noise level is defined as the logarithm of average noise power (see equation (3.1)).

GPS, MGRS conversions

The reasons for approximating GPS by square areas are two fold. First, computing the $LA_{eq,T}$ for every possible GPS coordinate is impractical because there are infinite GPS coordinates. Secondly, the acoustic standards for monitoring noise pollution recommend measuring the pollution in square areas (Section 5.3.1(a) in [1]) assuming that the noise level is constant over that area. In order to approximate GPS by grids, we use MGRS, which can divide the earth surface into squares of $100 \text{ m} \times 100 \text{ m}$, $10 \text{ m} \times 10 \text{ m}$ or $1 \text{ m} \times 1 \text{ m}$ etc.

We followed the Australian acoustic standard to determine an appropriate grid size. We assume that the volunteers walk along the pavement (or sidewalk) and measure ambient noise on the street level which is the aggregate of the noise generated by multiple moving vehicles. The Australian acoustic standard restricts the noise level difference between two adjacent grids to be no more than 5 dB (Section 5.3.2 in [1]). Therefore, we conducted a number of experiments where we put a MobSLM at a static position and put another MobSLM at difference distances from the first MobSLM and recorded the difference of $LA_{eq,1s}$

Table 3.3: Impact of Time window: Right side up orientation.

Time Window Size (Seconds)	Pocket		Bag or belt Case	
	Correct Classification (%)	False Negative (%)	Correct Classification (%)	False Negative (%)
10	87.5	12.5	87.5	12.5
20	93.75	6.25	90.625	9.375
30	87.5	12.5	90.625	9.375
60	93.75	6.25	90.625	9.375
90	93.75	6.25	93.75	6.25
120	93.75	6.25	93.75	6.25
180	93.75	6.25	93.75	6.25
240	93.75	6.25	93.75	6.25

Table 3.4: Impact of Time window: Left side up orientation.

Time Window Size (Seconds)	Pocket		Bag or belt Case	
	Correct Classification (%)	False Negative (%)	Correct Classification (%)	False Negative (%)
10	80	20	83.33	16.66
20	86.66	13.33	90	10
30	93.33	6.66	93.33	6.66
60	100	0	96.66	3.33
90	100	0	96.66	3.33
120	100	0	96.66	3.33
180	100	0	96.66	3.33
240	100	0	93.33	6.66

readings for each distance. For grid sizes of 10×10 , 20×20 , 30×30 , 40×40 and 50×50 square meters, the corresponding noise level differences between adjacent grids were found to be $2.26 \pm .06$, $3.82 \pm .05$, $3.86 \pm .03$, $4.11 \pm .02$ and $4.97 \pm .03$ dB, respectively. We could therefore use square grids which are less than or equal to 50 meters in each dimension. We chose to use a grid size of $30\text{m} \times 30\text{m}$ because it takes approximately 30 seconds for a Nokia N95 to acquire a GPS position. In that time, a person can travel 30 meters at normal walking speed (1 m/s). We use formulations in [20] to convert between GPS and MGRS.

Signal Reconstruction Module

To study the sampling requirements, communication overhead and reconstruction accuracy trade-offs within Ear-Phone, we developed two sensing strategies. In this section, we will describe the two sensing strategies, namely the *projection method* and the *raw-data method*, and also describe how the central server performs reconstruction using the information collected by these two different sensing strategies. For ease of explanation, we will explain the two sensing strategies with an example.

Consider the trajectory of two volunteers, A and B , along a section SG of a one dimensional street (see Fig. 3.7). Section SG contains three MGRS grid references: ℓ_1, ℓ_2 and ℓ_3 . Suppose at times t_1 and t_2 , volunteer A collects noise samples in grids ℓ_1 and ℓ_2 , and B collects samples in grids ℓ_3 and ℓ_1 respectively. Note that the noise sample in a grid refers to the equivalent noise level $LA_{\text{eq},1s}$ in that grid. The complete noise samples in section SG , during time t_1 and t_2 can be represented as a vector $x = [d(\ell_1, t_1), d(\ell_2, t_1), d(\ell_3, t_1), d(\ell_1, t_2), d(\ell_2, t_2), d(\ell_3, t_2)]^T$, where $d(\ell, t)$ is the noise level at locations $\ell = \{\ell_1, \ell_2, \ell_3\}$ and time $t = \{t_1, t_2\}$. We refer to the vector x as a *noise profile*. Similarly, samples collected by A and B can be represented as vectors

$$x_A = [d(\ell_1, t_1), 0, 0, 0, d(\ell_2, t_2), 0]^T \text{ and}$$

$$x_B = [0, 0, d(\ell_3, t_1), d(\ell_1, t_2), 0, 0]^T \text{ respectively.}$$

In the projection method, A multiplies his measurement vector x_A with a projection vector

$$\phi_A = [\phi_A^1, 0, 0, 0, \phi_A^5, 0]^T, \text{ where } \phi_A^1, \phi_A^5 \text{ are Gaussian distributed random numbers with zero mean and unit variance, and sends the projected value, } y_A =$$

Table 3.5: Impact of Time window: Display down orientation.

Time Window Size (Seconds)	Pocket		Bag or belt Case	
	Correct Classification (%)	False Negative (%)	Correct Classification (%)	False Negative (%)
10	83.33	16.67	80.56	19.44
20	77.78	22.22	83.33	16.67
30	83.33	16.67	86.11	13.89
60	88.89	11.11	83.33	16.67
90	88.89	11.11	86.11	13.89
120	88.89	11.11	88.89	11.11
180	94.44	5.56	88.89	11.11
240	94.44	5.56	88.89	11.11

Table 3.6: Impact of Time window: Display up orientation.

Time Window Size (Seconds)	Pocket		Bag or belt Case	
	Correct Classification (%)	False Negative (%)	Correct Classification (%)	False Negative (%)
10	57.14	42.86	75	25
20	57.14	42.86	75	25
30	64.29	35.71	75	25
60	81.43	18.57	78.57	21.43
120	81.43	18.57	85.71	14.29
150	81.43	18.57	82.14	17.86
180	81.43	18.57	89.29	10.71
240	81.43	18.57	85.71	14.29

$\phi_A^T x_A$ to the central server. Note that the inner product $\phi_A^T x_A$ is known as a projection in compressive sensing.

In the raw-data method, A directly sends his noise samples to the central server. Then, at the central server the projection vector for A 's data is regenerated as

$\phi_A = [\phi_A^1, 0, 0, 0, 0, 0; 0, 0, 0, 0, \phi_A^5, 0]^T$, where $\phi_A^1 = \phi_A^5 = 1$. Note that the projected value is again given by $y_A = \phi_A^T x_A$. In fact, in this case, y_A is a vector consisting of A 's measurements $d(\ell_1, t_1)$ and $d(\ell_2, t_2)$.

At the central server the reconstruction module accumulates the projected values from all volunteers in a vector $y = [y_A, y_B]^T$ and forms the projection matrix, $\Phi = [\phi_A^T, \phi_B^T]$. The reconstruction proceeds in two steps. In the first step, the central server solves the following optimization problem:

$$\hat{g} = \arg \min_{g \in \mathbb{R}^N} \|g\|_1 \text{ such that } y = \Phi \Psi g \quad (3.5)$$

where Ψ is a transform basis in which the noise profile x is compressible. In the appendix we report that the noise profile x is compressible in the Discrete Cosine Transform (DCT) basis. In the second step, an estimate of the noise profile x is given by $\Psi \hat{g}$. Note that the optimization problem (3.5) is a convex optimization and there exist efficient numerical routines for this class of problems.

In our current prototype implementation we used a simplified ‘‘query to grid resolver’’, which is essentially a look up table, to store the grid indices of the road segments. We only stored the grid indices of the road segments where our experiments were conducted. We used widely available open-source software for query manager and communication manager, therefore we do not describe these components in further detail.

4 Implementation and Evaluation

In this section, we first describe the Ear-Phone implementation. Then, we evaluate the system performance in terms of noise-level measurement accuracy, classification performance, resource (CPU, RAM and energy) usage and noise-map generation, which demonstrates that Ear-Phone is an effective end-to-end

mean of the difference in readings between the commercial meter (we refer it by RefSLM) and our mobile based SLM, as the offset. After adding the computed offset, we repeat the experiment and plot the responses in Fig. 4.1(b). We observe that our mobile phone based SLMs have a precision of ± 2.7 dB. Note that a difference of 3 dBA is *imperceptible* to the human ear. Note also that we found that phones from the same model could have different calibration offsets. This essentially means that a calibration technique needs to be developed to automatically calibrate the mobile phones of volunteers.

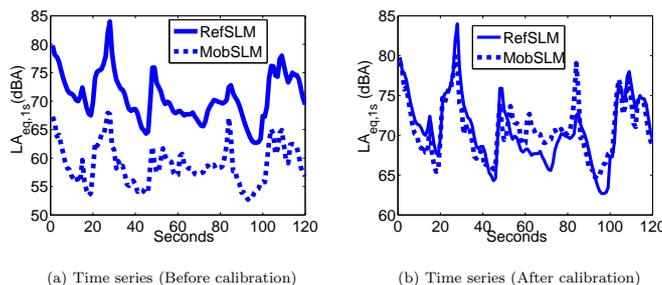


Figure 4.1: Measurement Accuracy of Ear-Phone.

In the above experiments, we assume that phones are carried in the volunteer’s palm or in a manner such that the microphone is not obstructed. However, we have also conducted experiments to investigate how the positioning of the phone affect the measurement accuracy. In these experiments we kept a MobSLM in three different positions: inside a volunteer’s trouser’s pocket, on his belt case, and in his bag, and recorded roadside equivalent noise levels. In order to compare the noise measurement, the volunteer also carried a MobSLM in one hand and the RefSLM in the other hand. Fig. 4.2 summarizes the experimental results. If the MobSLM is being held in our palm, then its accuracy is within 2.7dB of the RefSLM. If the phone is in a trouser’s pocket, the accuracy is within 3.1 dB of the RefSLM. The additional error of 0.7dB is small compared with the actual noise level of about 67dB. If the phone is carried in a belt case or in a bag, then the accuracy is within 4.1dB (3.7 dB for pocket and 4.1 dB for belt case) of the RefSLM. This measurement error is quite high since an increase in 5dBA would be perceived as doubled the loudness. We therefore want to exclude the data if it comes from bag or belt case.

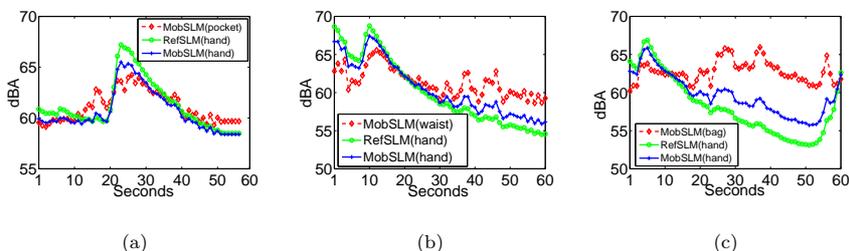


Figure 4.2: Measurement accuracy when the mobile phone is carried (a) inside a trouser’s pocket (b) in waist belt case (c) inside a bag.

Table 4.1: Classification record of Classifier 1.

		Subject 1	S2	S3	Mean
Palm	Correct Classification(%)	98.88	100	98.89	99.25
	False Negative(%)	1.12	0	1.11	0.75
Pocket, bag or Belt Case	Correct Classification(%)	97.5	98.61	99.16	98.33
	False Negative(%)	2.5	1.39	0.84	1.67

4.3 Classification Performance

We used following two metrics with parameters TP = True Positive, TN = True Negative, FP = False Positive and FN = False Negative to evaluate both of our classifiers.

- Sensitivity = $\frac{\#TP}{\#FP + \#FN}$: Corresponds to proportion of correctly detected events.
- Positive Predictive Value (PPV) = $\frac{\#TP}{\#TP + \#FP}$: Probability of correctly detecting an event when the system is exposed to a matching event.

We first compute the parameters (such as TP, FN etc) for each of the classifiers and then evaluate the metrics.

Classifier 1

Recall that, classifier 1 performs a simple binary classification using the proximity sensor. It can only tell whether a phone is held in the hand or in one of the other non-handheld positions (pocket, bag or belt case). To evaluate the performance of the classifier an experiment was conducted with 3 subjects. For each subject, 3 experimental runs were recorded. Each run was for 12 min and the phone was kept in each of the contexts for 3 minutes. The classifier executes every second on the phone. Result summarized in Table 4.1 shows that the mean accuracy is greater than 97%. With the mean values computed in Table 4.1 we compute the metric (see Figure 4.3) for classifier 1. The average value of sensitivity and PPV are greater than 98%.

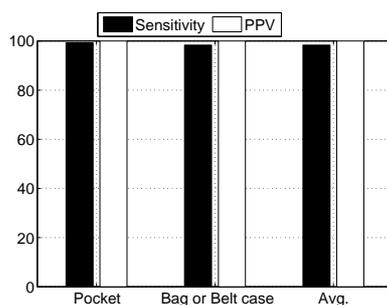


Figure 4.3: Performance metric of classifier 1.

Classifier 2

Recall that Classifier 2 can differentiate between phone held in pocket with bag or belt case enclosure. Also recall that we use k-NN to cluster out pocket data

Table 4.2: Classification record of Classifier 2.

Orientation	Pocket		Bag or belt Case	
	Correct Classification (%)	False Negative (%)	Correct Classification (%)	False Negative (%)
Left Side Up	100	0	100	0
Right Side Up	100	0	91.67	8.33
Display Up	81.33	18.67	86	14
Display Down	88	12	87.11	12.99

form bag or belt. Note that K-NN needs test data set, therefore, in order to evaluate the performance of classifier 2 we formed a test data set using data from 10 additional subjects where 6 subjects were male and 4 subjects were female. We stored the k-NN cluster mapping (resulting from the training data set) in the phone memory for different orientations. In Section 3 we have shown that for all orientations, z axis was common to most accurately cluster the data set into two groups. Therefore, rotation axis was chosen to be z axis. Time window was chosen to be 180 seconds. Each subject carried three phones at a time with a given orientation in all three places and walked along a street for 5 minutes. After finishing one walk, orientation of the phones were changed in all three places and the process was repeated for all four orientations. Time window was used 180 seconds. After completing the time window the classifier on the phone computed the variance of rotation and average speed and comparing with the stored cluster mapping computed the context.

Table 4.2 summarizes the experimental result from the above experiments. We observe that over different orientations and positioning, accuracy of Classifier 2 is above 80%. Using the value of the parameter determined in Table 4.2 the average value of sensitivity and PPV for classifier 2 are greater than 80%.

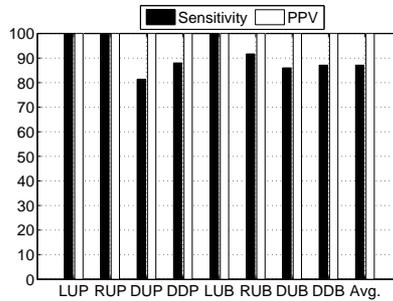


Figure 4.4: Performance metric of classifier 2. DUP/B=Display up Pocket/Bag or belt Case, DDP/B=Display down Pocket/Bag or belt Case, LUP/B=Left up Pocket/Bag or belt Case, RUP=Right up Pocket/Bag or belt Case.

4.4 Resource Usage

Power Benchmarks

We measure the power consumption of Ear-Phone using the Nokia Energy Profiler, a standard software tool provided by Nokia specifically for measuring energy usage of applications running on Nokia hardware. The profiler measures battery voltage, current, and temperature approximately every fourth of a second and stores the results in the RAM. Power consumption of different classifiers and the signal processing thread is shown in Fig. 4.5. We observe that in order

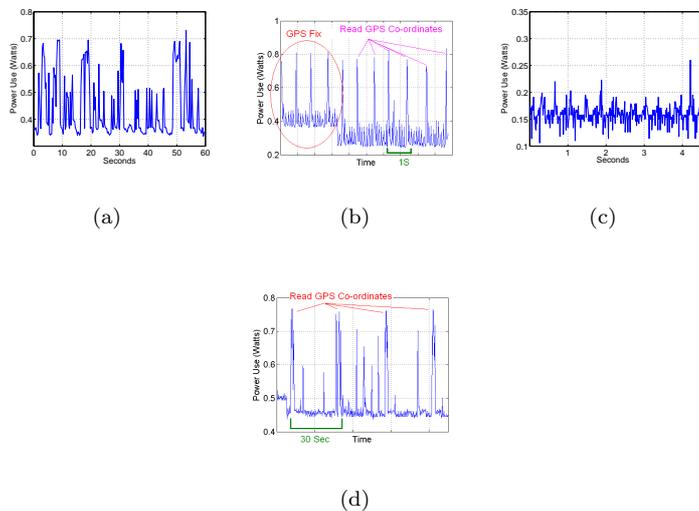


Figure 4.5: Power Consumption of (a) Classifier 1: reading proximity (b) Classifier 2: reading speed from GPS Receiver (c) Classifier 2: reading rotation and orientation (d) Signal processing thread.

to acquire reading from rotation and orientation sensors, classifier 2 requires least amount of power, however, power consumption for reading speed from the GPS receiver is high. Such as every time reading a GPS coordinates requires 0.8 watts. Similar consumption is observed for signal processing thread, as it also consumes approximately 0.8 watt to acquire a reading from GPS receiver. Surprisingly, the proximity sensor also consumes significant power, average power consumption is 0.45 watt.

Note that, our classifiers do not need to run continuously on the phone. Such as in Section 3 we have shown that a window size of 180 seconds is sufficient to determine the position of the phone. Therefore, after running continuously for say 180 seconds we could stop it for a while (determining a suitable interval is included in our future study) since it is likely that people do not change phone location very frequently. Furthermore when classifier 1 determines that phone is in hand classifier 2 is not triggered at all.

Memory and CPU Benchmarks

We also carried out benchmark experiments to quantify the RAM and CPU usage of Ear-Phone running on the N95 using the Nokia Energy Profiler tool. To precisely measure the resource consumption, we enable the screen saver to disassociate the resource occupation of the N95 LCD. We first measure the amount of RAM and CPU usage when the phone is idle. Then, we repeat the measurement to determine the worst case power consumption i.e., the signal processing thread is running with all classifiers. We find the CPU usage is approximately 73% and memory usage is approximately 97 MB. In order to find the significant contributor we determine the CPU load due to individual classifiers and threads and find that signal processing thread consumes about 35% CPU load followed by proximity and rotation sensor. Detail report of memory and CPU consumption by different classifiers and threads are summarized in Table 4.3.

Table 4.3: CPU and RAM usage.

	CPU Load (%)	RAM (MB)
Phone Idle	2±0.79	32.86
Ear-Phone all classifiers+threads	73.21±20.03	97.10
Rotation	21.08±5.95	87.73
Orientation Sensor		
Speed (GPS receiver)	8.10±10.65	87.47
Proximity Sensor	20.75±27.39	80.45
Signal Processing Thread	34.89±26.32	90.71

The current Ear-Phone implementation is not optimized for CPU utilization or power consumption since our main concern at this stage is the accuracy of the noise map. Proper techniques can be designed to minimize usage of these resources.

4.5 Performance Evaluation

To evaluate the performance of Ear-Phone as an end-to-end system, we conducted several outdoor experiments. Our primary goal is to investigate the impact of data availability on reconstruction performance. In the experiments, we reconstructed the noise map along a major road intersection in Brisbane, Australia. This intersection includes Mogill Road, a major artery that carries significant traffic, and Bainbridge Drive, which is a branch road that leads to a residential neighborhood. Consequently Mogill road is much noisier compared to Bainbridge Drive. We reconstructed an hourly noise map during peak (8:00-9:00) and off-peak (14:00-15:00) hours. To collect noise samples, we walked along these segments several times within the one hour period with Ear-Phone running on the Nokia N95. The path used is marked with arrows in Fig 4.6. The travel time was approximately 5 minutes for each walk (from start to end of the segment) and we traveled 8 times during a one hour period. Each walk represents a different person walking along the segment and contributing data.

To investigate the impact of data availability on the reconstruction, we reconstructed the noise profile by varying the number of contributing persons, and including the data contributed by the corresponding persons. For each person, we reconstructed the noise profile during his 5-minute travel. We reconstructed separately for Mogill Rd and Bainbridge Drive. Using the reconstructed $LA_{eq,T}$, for each person we computed $LA_{eq,LT=1hr}$ using Eq.(3.4). We repeated this process to compute $LA_{eq,1hr}$ using measurements from multiple people. Figs. 4.7 and 5.1 show the impact of measurements included from a varying number of persons on the reconstruction accuracy during off-peak and peak hours respectively.

When we use data from only one person, the reconstruction does not reveal any distinct patterns along the noisy and quiet streets. In fact, the reconstruction appears to be random (in our experiments, a single person collects only a small amount of information of the temporal-spatial noise profile, which is not sufficient for the Compressive sensing based reconstruction algorithm to succeed. This is why the reconstruction is random.). However, when we include data from multiple persons, the reconstruction gradually reveals the contrast between the noisy and quiet streets. Furthermore, after a certain threshold, increasing data contributors does not improve the reconstruction accuracy significantly. For example, comparing Fig. 4.7(c) and Fig. 4.7(d), it is evident that the reconstruction achieved by data from 4 person is similar to that from 6 person. A similar behavior can be seen in Fig. 5.1(c) and Fig. 5.1(d).

During these experiments, we simultaneously measured the $LA_{eq,LT}$ using our commercial sound level meters placed midway along Mogill Rd and Bain-

bridge drive. Comparing the reconstructed noise map with the commercial sound level meter readings, we find that we need measurements from at least 5 person during peak hour and from a minimum of 4 person during off-peak hour, for a reconstruction comparable to the commercial sound level meter. Note that data from 4/5 person was sufficient for the noise profile we considered in our experiments. It may change for different noise profiles. The amount of data needed depends on both the noise profile and the percentage of missing data. This topic is studied in the following section.



Figure 4.6: Data collection route.

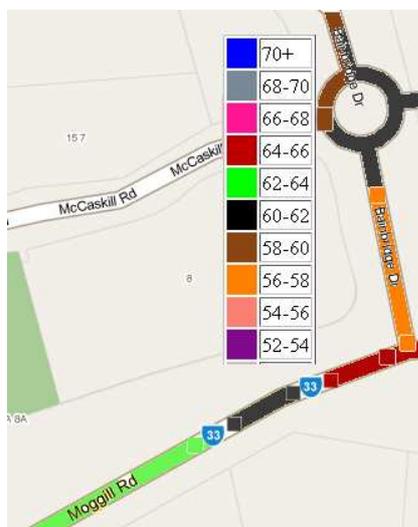
5 Simulation

Real experiments certainly provide valuable information. However, real experiments are not repeatable. Conducting a real experiment on a large scale is expensive and time consuming. We therefore conducted simulation experiments where factors such as the number and mobility patterns of volunteers, sensing strategies (see Section 3.3) etc. can be varied easily. In this section, we will first describe how we perform measurement campaigns to collect noise profiles which will be fed into the simulation as ground truth. Next, we will describe the simulation itself and performance evaluation in terms of reconstruction accuracies.

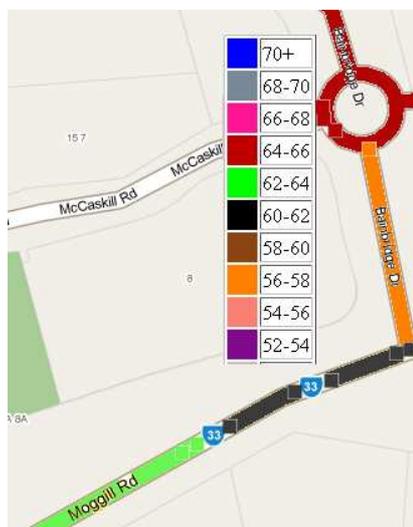
5.1 Simulation Design

As in Section 4, we limit our consideration to noise measurements along a road, which can be modeled as a scalar field over a uniform 2-dimensional grid of cells with one spatial and one temporal dimension. We assume that each cell has a spatial width of D meters and a temporal width of T seconds. We use the ordered pair (i, j) to refer to the cell bounded by the spatial interval $[(i-1)D, iD]$ and temporal interval $[(j-1)T, jT]$. Assuming that $i \in N_s = \{1, 2, \dots, n_s\}$ and $j \in N_t = \{1, 2, \dots, n_t\}$, the reference grid covers a length of $n_s D$ meters and a duration of $n_t T$ seconds. We assume that the equivalent noise level $LA_{eq,T}$ measured over each cell is almost constant. Now let $d(i, j)$ denote the equivalent noise level $LA_{eq,T}$ measured in cell (i, j) , then a *noise profile* S is defined as the set of all $LA_{eq,T}$ measured over the defined grid, i.e. $S = \{d(i, j)\}_{(i,j) \in N_s \times N_t}$.

Our first task is to conduct a number of measurement campaigns to obtain *reference noise profiles* which we can feed into the simulation as ground truth.



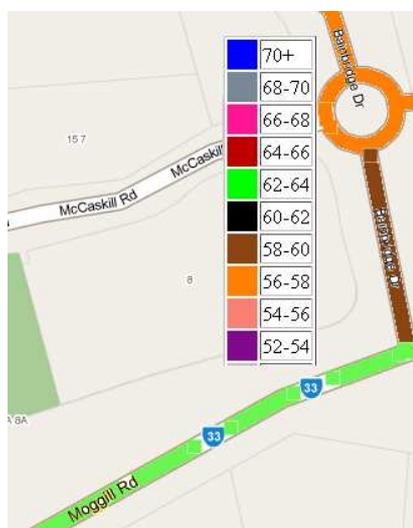
(a)



(b)



(c)



(d)

Figure 4.7: Noise map reconstruction during an off peak hour (2:00pm-3:00pm) using data from (a) 1 person, (b) 2 persons, (c) 4 persons and (d) 6 persons.

We conducted four experiments to collect $LA_{\text{eq},1s}$ under a variety of noise conditions and settings. The experimental conditions and parameters used are summarized in Table 5.1. During each of these experiments, we measured $LA_{\text{eq},1s}$ along Anzac Parade, which is a major artery road in Sydney. This road has two-way traffic with 3 lanes in each direction. The traffic flow was reasonably high as indicated by the mean noise level in Table 5.1. We used 6 MobSLMs (HP iPAQ) to capture the reference noise profile and placed them in 6 equidistant locations along the road with the microphone pointed towards the road. Different spatial separations are used in the experiments, see Table 5.1. The clocks on the phones were synchronized to ensure that all phones start and stop sampling at the same time. The MobSLMs measured $LA_{\text{eq},1s}$ during the experiment and stored the data in a text file which was downloaded to a computer at the end of the experiment. From each experiment, we created a reference noise profile, where $|N_s| = 6$ and $|N_t|$ is the experimental duration in seconds. We deliberately conducted one experiment (see Table 5.1) with a side road between the mobiles to create a reference profile with high noise variation (side road divides the traffic flow, therefore noise levels on either side of the road typically have high difference.).

Our simulation considers only discrete agent (we refer to simulated volunteers as agents) movements. Let $d_i \in [0, n_s D]$ denote the position of the agent at time iT seconds. The location of this agent at time $(i+1)T$ is given by $d_{i+1} = d_i + V_i T$ where V_i is the average speed (in ms^{-1}) of the agent in the time interval $[iT, (i+1)T]$. The value of V_i is assumed to be uniformly distributed in $[0, 1.11]$ where $1.11 \text{ ms}^{-1} = 4 \text{ km/hr}$ is the typical walking speed [3]. The sign of V_i determines the direction of movement. In our setting, the agent is in cell $(\lceil \frac{d_i}{D} \rceil, i) \in N_s \times N_t$ at time iT , where $\lceil u \rceil$ denotes the smallest integer that is greater than or equal to u . We consider a particular agent and let $W \subset N_s \times N_t$ denote all the cells visited by this particular agent. To simulate urban sensing, we assume that an agent does not take samples at all visited cells (Due to privacy concerns, volunteers may not contribute samples near their home or office. The microphone may be in use for conversation). Let $\tilde{W} \subset W$ denote the set of all cells whose data is contributed by this agent.

Simulating Sensing Strategies

In the projection method, an agent uses the $LA_{\text{eq},1s}$ samples collected in the cells in \tilde{W} to form a projection. Recall from Section 3 that a projection is essentially a linear combination of the data. The agent computes

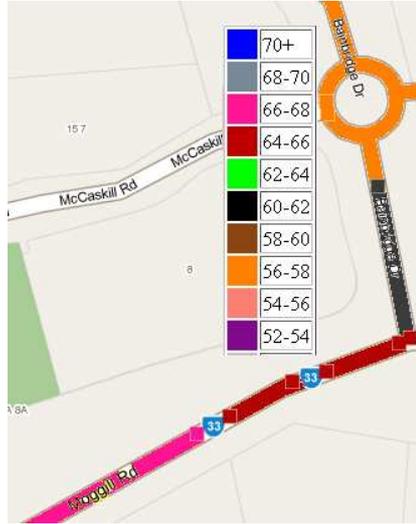
$$\tilde{y} = \sum_{(i,j) \in \tilde{W}} d(i,j) \eta(i,j) \quad (5.1)$$

where $d(i,j)$ is the $LA_{\text{eq},1s}$ sample collected at cell (i,j) and $\eta(i,j)$'s (with $(i,j) \in \tilde{W}$) are $|\tilde{W}|$ random numbers drawn from the standard Gaussian distribution. The agent transmits the projected value \tilde{y} to the central server, along with the seed used to generate the random coefficients of the projection vector. In the raw-data method, the agent sends $d(i,j)$ values and $(i,j) \in \tilde{W}$ (note that i and j represents location and time respectively) to the central server.

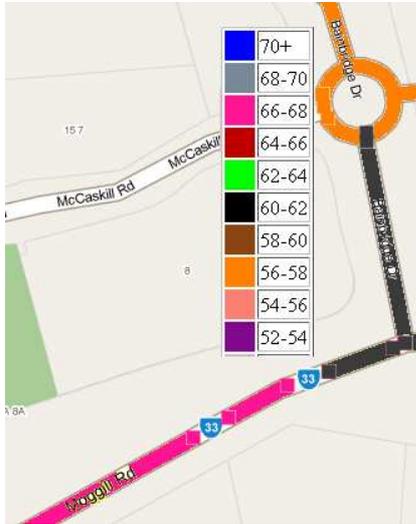
Let $\tilde{S} = \{d(i,j)\}_{(i,j) \in \tilde{W}} \subset S$ be the $LA_{\text{eq},1s}$ samples collected by volunteers. The reconstruction operation can be viewed as the estimation of the missing samples in the noise profile S from the information in \tilde{S} . Let $\hat{S} = \{\hat{d}(i,j)\}_{(i,j) \in N_s \times N_t}$ be a reconstruction of S . Then we compute root mean



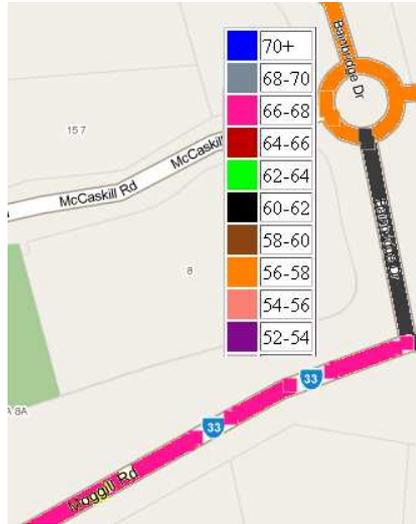
(a)



(b)



(c)



(d)

Figure 5.1: Noise map reconstruction during a peak hour (8:00am-9:00am) using data from (a) 1 person, (b) 3 persons, (c) 5 persons and (d) 7 persons.

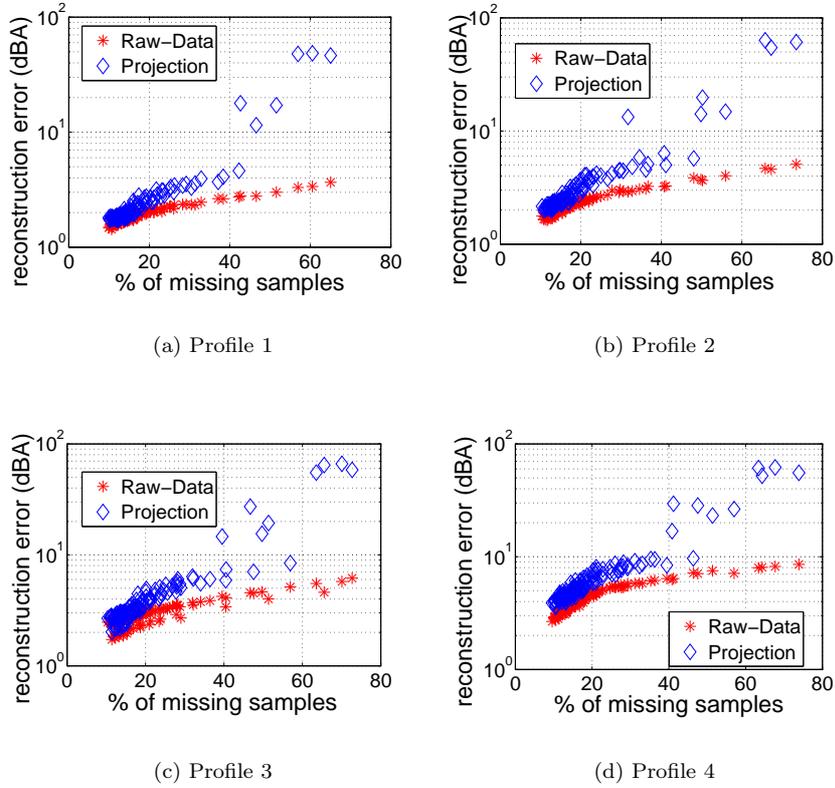


Figure 5.2: Percentage of missing data (x -axis) and its impact on reconstruction accuracy expressed in RMS error (y -axis).

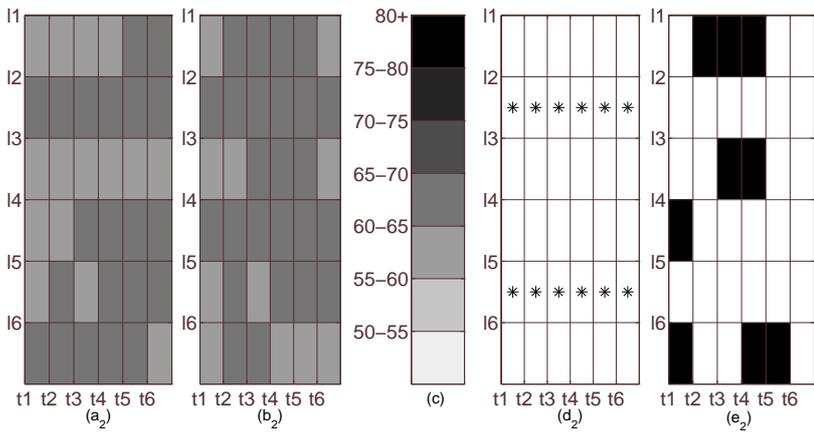
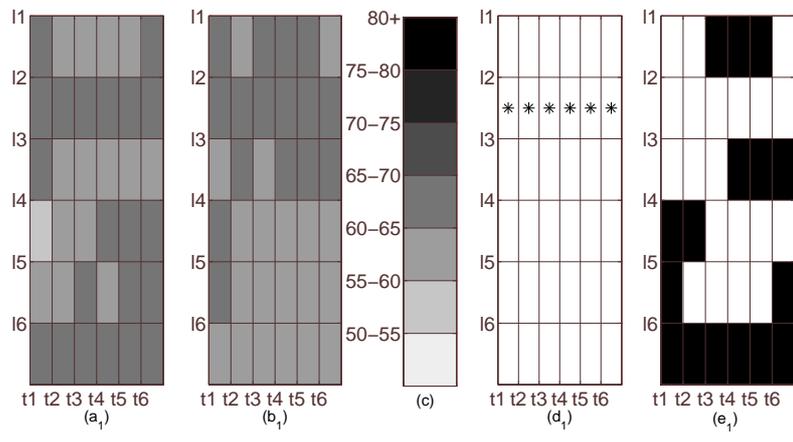
square (RMS) reconstruction error by:

$$S_{rms} = \sqrt{\frac{1}{n_s \times n_t} \sum_{1 \leq i < n_s, 1 \leq j < n_t} (d(i, j) - \hat{d}(i, j))^2} \quad (5.2)$$

5.2 Performance Evaluation

Noise map reconstruction

As discussed earlier, the key benefit of using compressive sensing is the ability to accurately reconstruct the spatio-temporal sensed field from incomplete and random samples. We now proceed to study the trade-off between the reconstruction accuracy, communication overhead and the percentage of missing data for the two sensing strategies discussed in the paper namely: (i) the raw-data method and (ii) the projection method. We used the 4 different noise profiles as a reference and evaluated the reconstruction performance under varied mobility patterns and number of agents. In Figs. 5.2(a) to 5.2(d) we plot the reconstruction accuracy as a function of sampling requirements for our reference noise



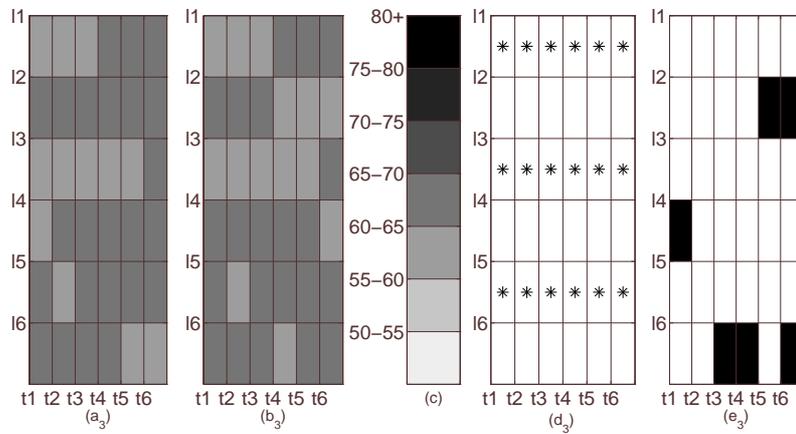


Figure 5.3: This figure shows the reconstruction performance at the cell level. Each row of this figure consists of 5 sub-figures (a_i) , (b_i) , ..., (e_i) where $i = 1, \dots, 3$. Each row ($i = 1, 2, 3$) shows the reconstruction of a section of the profile for a given percentage of missing data. The percentage of data used for rows 1, 2, 3 are, respectively, 18.42%, 34.73% and 45.03%. Sub-figure (a_i) shows a section of the reference profile. Note that each section consists of 6 locations (l1, ..., l6) over a duration of 6 seconds (t1, ..., t6). The same reference profile is used for all 3 rows. (c) The scale of noise levels (d_i) * in a cell means the $LA_{eq,1s}$ sample from that cell is used in the reconstruction. (e_i) Reconstruction error. A black-filled cell indicates that the error for that cell is more than 3 dBA. The more white cells the better reconstruction.

Table 5.1: Experimental settings for collecting the reference noise profiles.

Exp No.	Date and time	Mean, Standard Deviation of sound level (dBA)	Spatial separation (meters)	Duration (min)	Continuous road segment without side roads	% of DCT coefficients needed to approximate the profile to within 1 dBA RMS error
1	21/08/08 3:00 pm	73.05,2.95	10	20	yes	27.83
2	21/08/08 4:30 pm	70.09,4.43	10	15	yes	35.15
3	29/08/08 5:14 pm	70.43,5.16	50	15	yes	39.94
4	01/09/08 6:24 pm	71.22,5.55	50	10	no	44.14

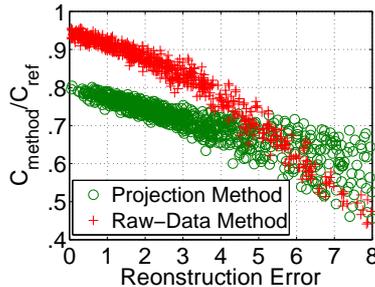


Figure 5.4: Reconstruction accuracy VS communication overhead.

profiles. We observe that the raw-data method has better reconstruction accuracy for all 4 reference profiles, specifically when the amount of missing samples is large. We observe that due to the aggregation of data, reconstruction becomes difficult in the projection method (Note that the aggregation inevitably leads to loss of information, but the projection method can reduce the communication requirements, see the next paragraph.). Except for profile 4, Ear-Phone can reconstruct the profiles to within 3dBA (3 dBA difference is not perceptible by human being) error with 40% or fewer missing samples. Profile 4 can accept slightly less missing samples. The increase in sampling requirements for profile 4 can be explained in terms of the profile compressibility. One way to determine the compressibility of a profile is to study the percentage of transform coefficients needed to approximate a profile to a given level of accuracy. The last column of Table 5.1 shows that profile 1 is the most compressible while profile 4 is the least compressible.

To demonstrate the reconstruction quality, we plot a section of the reconstructed profile in Fig. 5.3. A total of 3 sections are shown in Fig. 5.3 for different percentages of missing samples for the raw-data method. Note that the reconstruction is pretty accurate at the cell level.

We now discuss the communication requirements of the raw-data and projection methods as a function of their reconstruction accuracy. Let C_{ref} denote the number of bytes returned, if $LA_{eq,1s}$ samples from all the cells of our profile are returned and let C_{method} denote the corresponding number of bytes returned by either raw-data or projection method. Fig. 5.4 shows a typical plot of (we plot only the result from experiment 4 due to space restrictions) C_{method}/C_{ref} as a function of the reconstruction error. We observe that, to limit the reconstruction error within 3dBA, the projection method and the raw data method reduce the ratio C_{method}/C_{ref} is 0.7 and 0.85 for projection and raw-data method respectively, i.e., the reduction in communication cost is 30% and 15% by projection and raw data method respectively. However, when a high reconstruction error is acceptable, the raw-data method is more communication efficient than the projection method.

Impact of positioning the phone in the pocket

In all of the above simulations, we assumed that the phone was always held in the hand by volunteers with the microphone correctly exposed for sampling the ambient environment. However, recall that in Section 4, we have demonstrated that the noise samples collected when the phone is placed in the pocket, though suffering from some deviation, are still usable. In this section, we evaluate the impact of positioning the phone in the pocket, on the reconstruction accuracy. Note that, the amount of time that the phone is placed in the pocket can vary from one individual to another. We therefore designed an experiment to study the impact of different amount of time the phone is in the pocket, on the noise map reconstruction. In this experiment we reused our four noise profiles. As earlier we used simulated user (agent) movement to collect $LA_{eq,T}$ samples from each of the profiles. In addition, we introduce a variable ρ which represents the percentage of the time phone was kept inside pocket. For example, if $\rho = 0.1$, for each agent, 10% of the data were recorded within pocket and 90% of data is recorded within palm. In Section 4 we have shown that the deviation of $LA_{eq,T}$ is ± 3.1 for pocket enclosure, therefore, we added a noise uniformly from $[3.1, -3.1]$ with the agent’s data to simulate its pocket enclosure. Fig. 5.5(a) to Fig. 5.5(d) shows the impact of pocket enclosure on the noise map reconstruction for profile 1 to profile 4 respectively. We use ρ in the range $[0 - 0.8]$, where $\rho = 0$ means each agent held the phone only in palm. We observe that when the percentage of data points used for reconstruction is approximately 50%, the impact of pocket enclosure (even for $\rho = 0.8$) is negligible, i.e., if we collect $LA_{eq,T}$ from 50% cells of a grid, the impact of amount of time phone is kept inside pocket becomes negligible.

6 Related Work

There are a number of efforts in the deployment of urban sensing applications, on the study of incentives to improve participation in human computation systems, and on improving the trustworthiness of participatory sensing. However, we focus our attention on the following.

In [23], the authors survey technical issues influencing the design and implementation of systems that use mobile phones to assess noise pollution. However, they do not provide an end-to-end system, and they do not study the problem of reconstructing the noise map from incomplete and random samples.

Noisetube [18] is a recently developed platform to generate a collective noise map by aggregating measurements collected by the public. As the authors do not provide any details on how they perform data aggregation, we cannot contrast EarPhone with this work.

Recent research in plenacoustic functions [2] studies the sampling requirement of an acoustic field. While the work in [2] deals with a continuous signal, our work considers a discrete signal over time and space. Specifically, we consider the equivalent noise level over a physical area and time duration.

Work presented in [15] studies the compressibility of acoustic signals in both spatial and temporal dimensions. A limitation of their work is that it is based on a single acoustic source in a laboratory setting. In addition, they aim to reconstruct the pressure waveform. This is different from our focus on studying the compressibility of temporal-spatial field of noise levels in an outdoor environment, which are influenced by multiple acoustic sources.

Community Sensing [17] uses a traditional interpolation framework to estimate missing data, when data is obtained via crowdsourcing. In contrast, we

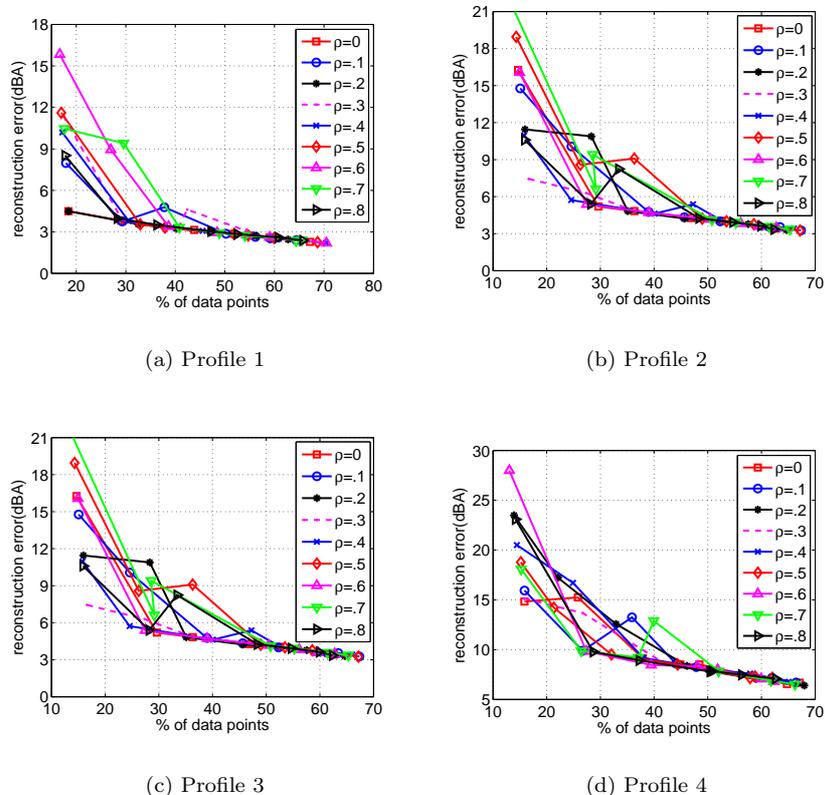


Figure 5.5: Impact of pocket enclosure on the reconstruction accuracy. ρ is the fraction of time phone is kept inside the pocket. Along x axis is the percentage of data points used for reconstruction and along y axis is the corresponding reconstruction error.

apply compressive sensing to show that temporal-spatial noise profiles are in fact compressible and clarify the sampling-accuracy trade-off.

Compressive sensing has so far been applied in traditional low-power wireless sensor networks [21, 8, 5]. For example, Compressive Wireless Sensing (CWS) [5] derives a method to compute the projection using the wireless channel. However, CWS cannot be applied to urban sensing because CWS requires the entire data set to form the projection. In this paper, we have proposed sensing strategies that are suitable for urban sensing.

Phone sensors have been used to detect the context of the person [19], however the context used in the paper is different and defined in terms of activity such as running, dancing, etc.

7 Conclusions and Discussion

In this paper, we presented the design, implementation and evaluation of Ear-Phone, an end-to-end noise pollution mapping system based on participatory urban sensing. Ear-Phone comprises signal processing software to measure noise pollution at the mobile phone, as well as signal reconstruction software and

query processing software at the central server. To address the problem of noise map reconstruction from incomplete data samples, a key issue in crowdsourced sensor data collection, we exploit the compressibility of the spatial-temporal noise profile and apply recently developed reconstruction methods from compressive sensing. We study the sensing and communication requirements of Ear-Phone. Using simulation experiments, we show that Ear-Phone can recover a noise map with high accuracy, allowing nearly 40% missing samples while reducing communication costs by 30%. Two different noise mapping experiments report that Ear-Phone can accurately characterize the noise levels along roads using incomplete samples.

Mobile phones are often carried inside bags or pockets. We propose a classification strategy that achieves approximately 92% accuracy to locally determine the context of the phone. We also show that when the phone is carried inside the pocket accuracy (3.1dB) of the MobSLM is better compared to when the phone is carried inside belt case or inside bag (4.7 dB).

We finally demonstrate that when the percentage of data points used for reconstruction is small, including large percentage of data from pocket introduces significant error. With the increase of percentage of data points for reconstruction, impact due to pocket enclosure becomes negligible. We finally demonstrate that enclosure in pocket has a negligible impact on the noise map reconstruction accuracy when we have 50% or less missing samples.

Bibliography

- [1] Australia/New Zealand Standards Committee AV/5. Australian Standard: acoustics description and measurement of environmental noise. AS 1055.3 1997, Part 3—Acquisition of data pertinent to land use.
- [2] Thibaut Ajdler and Martin Vetterli. The Plenacoustic function, sampling and reconstruction. In *WASPAA*, page 147, 2003.
- [3] Alberta Center for Active Living. Watch your steps: pedometers and physical activity. *WellSpring*, 14(2):489–509, 2003.
- [4] Audacity. Free, cross-platform sound editor and recorder. <http://audacity.sourceforge.net>.
- [5] Waheed Bajwa, Jarvis Haupt, Akbar Sayeed, and Robert Nowak. Compressive wireless sensing. In *IPSN*, pages 134–142, 2006.
- [6] J Burke et al. Participatory sensing. In *WSW06: Mobile Device Centric Sensor Networks and Applications*, 2006.
- [7] Center Technology Corp. Center c322.
- [8] Chun Tung Chou, Rajib Rana, and Wen Hu. Energy efficient information collection in wireless sensor networks using adaptive compressive sensing. In *In Proc. LCN 2009*, pages 443–450, 2009.
- [9] Yanqing Cui, Jan Chipchase, and Fumiko Ichikawa. A cross culture study on phone carrying and physical personalization. In *HCI (10)*, pages 483–492, 2007.
- [10] DEFRA. Noise mapping england. <http://www.noisemapping.org/>.
- [11] Department for Health and Environment of the City of Munich (Germany). Noise maps 2007. <http://tinyurl.com/>.

- [12] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.
- [13] S Eisenman et al. Metrosense project:people-centric sensing at scale. In *Workshop on World-Sensor-Web (WSW06): Mobile Device Centric Sensor Networks and Applications*, 2006.
- [14] European Union. Future noise policy, com (96) 540 final. European Commission Green Paper, Nov 1996.
- [15] A. Griffin and P. Tsakalides. Compressed sensing of audio signals using multiple sensors. In *EUSIPCO 2008*, August 2008.
- [16] International Electrotechnical Commission. Electroacoustics - sound level meters - part 2: Pattern evaluation tests, April 2003.
- [17] Andreas Krause, Eric Horvitz, Aman Kansal, and Feng Zhao. Toward community sensing. In *IPSN*, pages 481–492, 2008.
- [18] N. Maisonneuve et al. Noisetube: Measuring and mapping noise pollution with mobile phones. In *ITEE 2009*, pages 215–228, May 2009.
- [19] Emiliano Miluzzo, Nicholas D. Lane, Kristf Fodor, Ronald Peterson, Hong Lu, Mirco Musolesi, Shane B. Eisenman, Xiao Zheng, and Andrew T. Campbell. Sensing meets mobile social networks: The design, implementation and evaluation of the cenceme application. In *in Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 337–350. ACM Press, 2008.
- [20] National Geospatial-Intelligence Agency (NGA). Datums, ellipsoids, grids, and grid reference systems. *DMA TECHNICAL MANUAL*.
- [21] Rajib Rana, Chun Tung Chou, and Wen Hu. Energy-aware sparse approximation technique (east) for rechargeable wireless sensor networks. In *In Proc. EWSN 2010*, 2010.
- [22] Silvia Santini et al. First experiences using wireless sensor networks for noise pollution monitoring. In *Proc. of the REALWSN'08*, April 2008.
- [23] Silvia Santini, Benedikt Ostermaier, and Robert Adelman. On the use of sensor nodes and mobile phones for the assessment of noise pollution levels in urban environments. In *Proc. of the INSS 2009*.

APPENDIX

In order to study the compressibility of noise profile, we compute their representations in a number of transform bases, which include DCT, Fourier and different wavelets such as Haar, Daubechies, Symlets, Coiflets, and Splines etc. For each basis, we compute the root mean square (RMS) error between the original profile and its approximation by retaining only the largest k ($k = 1, 2, \dots$) coefficients in that basis. Fig. 7.1 is a representative plot that shows the compressibility of noise profile in DCT, Haar and Fourier basis (The results in Figure 7.1 is obtained from reference profile 4 mentioned in Section 5. We have carried out similar study using the other collected noise profiles, and they give similar results.). We observe that for same number of coefficients, the representation in DCT gives a lower error compared to other bases. In the last column of Table 5.1, we have summarized the percentage of DCT coefficients required to approximate the profiles collected in all experiments within 1 dBA RMS error.

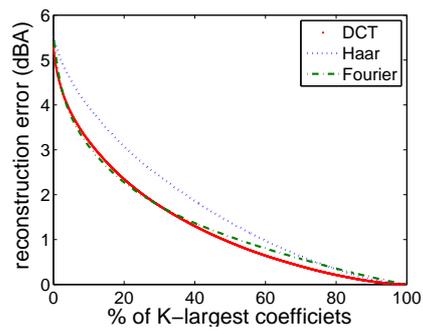


Figure 7.1: Compressibility of the noise profile