Ear-Phone: An End-to-End Participatory Urban Noise Mapping System

Rajib Kumar Rana[1] Chun Tung Chou[1] Salill Kanhere[1]
Nirupama Bulusu[2]
Wen Hu[3]

[1] University of New South Wales, Australia
{rajibr,ctchou,salilk}@cse.unsw.edu.au
[2] Portland State University, USA
nbulusu@cs.pdx.edu
[3] CSIRO ICT Centre, Australia
wen.hu@csiro.au

Technical Report
UNSW-CSE-TR-0920
October 2009

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 and HP iPAQ mobile devices, also addresses the challenge of collecting accurate noise pollution readings at a mobile device. We evaluate Ear-Phone with extensive simulations and outdoor experiments, that demonstrate that it is a feasible platform to assess noise pollution with reasonable system resource consumption at mobile devices and high reconstruction accuracy of the noise map.

1 Introduction

At present, a large number of people around the world are exposed to high level of noise pollution, which can cause serious illness ranging from hearing impairment to negatively influencing productivity and social behavior [12]. As an abatement strategy, a number of countries, such as the United Kingdom [9] and Germany [10], 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 data. Since the collection of such input data is highly expensive, these maps can be updated only after a long period of time (e.g. 5 years for UK [9]). To alleviate this problem, a recent study [19] 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 as participatory sensing [6], people-centric sensing [11] or community sensing [15] in the literature) 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 urban sensing can be updated with a very small latency (hours or days compared to months or years), which makes information provided by noise map significantly closer to current noise status 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. They may also 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. In order to develop a useful noise pollution monitoring application, we need to recover the noise map from the *random and incomplete* samples obtained via crowdsourcing. In this paper, we address these challenges. Our main contributions can be summarized as:

1. We present the design and implementation of an *end-to-end* noise mapping

[1]For example, devices such as the Nokia N96 or HP iPAQ do not support floating-point arithmetic, which must be emulated with fixed point operations.

system, called Ear-Phone, to generate the noise map of an area using participatory urban sensing. EarPhone 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 (or missing) 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 evaluated Ear-Phone with extensive simulations and real-world outdoor experiments. The results show that Ear-Phone has reasonable accuracy, and resource requirement in terms of CPU load and energy consumption.

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 with future directions in Section 7.

2 Ear-Phone Architecture

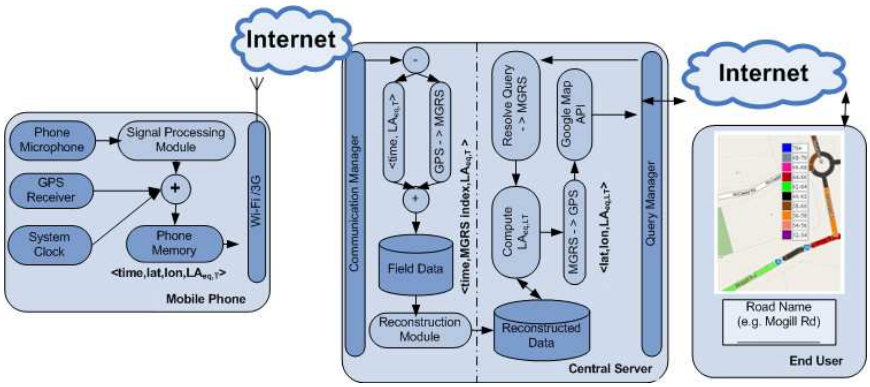


Figure 2.1: Ear-Phone Architecture

In this section, we describe the high-level view of how Ear-Phone works and a detailed description of the system components will be presented in Section 3. Fig. 2.1 presents the overall architecture of Ear-Phone. The Ear-Phone architecture consists of a mobile phone component and a central server component. Noise level is assessed on the mobile phones before being transmitted to the central server. The central server reconstructs the noise map based on the noise measurements. Note that reconstruction is required because the urban sensing

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

framework cannot guarantee noise measurements are available at all the time and locations.

Let us begin with a mobile phone user who is walking along a street. We call a mobile phone with 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. The signal processing module on the MobSLM computes the equivalent noise level ($\mathrm{LA}_{\mathrm{eq,T}}$) over a time interval T from the raw acoustic samples collected by the microphone over the same time interval. The computed noise level is further attested 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,$\mathrm{LA}_{\mathrm{eq,T}}$ $\rangle$ are uploaded to the central server when the mobile phone detects an open access point (3G services on mobile phones can also be used to upload data.).

The communication manager at the central server waits for transmissions from the users. When there is a transmission, it converts the GPS coordinates of a record to a Military Grid Reference System (MGRS, see Section 3.2 for the detailed description) grid index and stores the information $\langle$ time, grid index, $\mathrm{LA}_{\mathrm{eq,T}}$ $\rangle$ in a data repository. Reconstruction is conducted periodically at predefined intervals (e.g., minutes or hours); 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 are fetched from the data repository. Then, the grid indices are converted back to GPS coordinates and the related noise levels are overlaid on an Internet map (e.g., Google map) before being displayed to the end user.

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 $\mathrm{LA}_{\mathrm{eq,T}}$. A-weighting is the commonly used frequency weighting that reflects the loudness perceived by human being [14]. Measured in decibel (dBA), $\mathrm{LA}_{\mathrm{eq,T}}$ 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 $\mathrm{LA}_{\mathrm{eq,T}}$ in time interval T is

thus given by

$$\begin{aligned} \mathrm{LA}_{\mathrm{eq,T}} &= 10\log_{10}(\underbrace{\frac{1}{T}\int_0^T (v_A(t))^2 dt}_{\bar{v}_A(T)}) \\ &+ \underbrace{\text{constant offset}}_{\Delta} \end{aligned} \tag{3.1}$$

where $v_A(t)$ is the result of passing induced voltage $v(t)$ through an A-weighting filter and the constant offset is 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 frequency response matches with that of A-weighting over 0–8kHz, since the acoustic standard, IEC651 Type 2 SLM [14], requires to measure the 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 the following algorithm.

Algorithm *Compute* $\bar{v}_A(T)$

1. 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 $k = 0, 1, 2, \ldots, Q-1$ over duration $[0, T]$;
Output: $\bar{v}_A(T)$

2. Based on $\{a_l, b_l\}$ and initial condition, $v_A(kT_s) = 0$ for $k = 0, ..., 9$, recursively compute

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

3. Compute

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

3.2 Central Server Components

Computing Long-term Equivalent Noise Level, $\mathrm{LA}_{\mathrm{eq,LT}}$

In order to compute the long-term equivalent noise level $\mathrm{LA}_{\mathrm{eq,LT}}$ over duration LT (where L is an integer bigger than 1) from the equivalent noise levels $\mathrm{LA}_{\mathrm{eq,T}}$ measured over shorter time duration T, we use the following standard formula:

$$\mathrm{LA}_{\mathrm{eq,LT}} = 10\log_{10}[\frac{1}{N}\Sigma_{i=1}^{N} 10^{0.1\mathrm{LA}_{\mathrm{eq,T_i}}}] \tag{3.4}$$

where N is the number reference time intervals and $\mathrm{LA}_{\mathrm{eq,T_i}}$ 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 equivalent noise level is defined as the logarithm of average noise power, see equation (3.1).

GPS, MGRS conversions

Reasons for approximating GPS to square areas are two fold. Firstly, computing the $\mathrm{LA}_{\mathrm{eq,T}}$ for every possible GPS coordinates is impractical because there are infinitive GPS coordinates. Secondly, the acoustic standards for monitoring noise pollution suggest to measure the pollution in square areas (Section 5.3.1(a) in [1]) assuming the noise level is constant over that area. In order to approximate GPS into grids, we use MGRS, which can divide the earth surface into a square area of such as 100 m × 100 m, 10 m × 10 m or 1 m × 1 m etc.

We followed the Australian acoustic standard to determine an appropriate grid size. This 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 $\mathrm{LA}_{\mathrm{eq,1s}}$ readings for each distance. We found that for the grid size of 10 × 10, 20 × 20, 30 × 30, 40 × 40 and 50 × 50 square meters, the corresponding noise level differences between adjacent grids are $2.26 \pm .06$, $3.82 \pm .05$, $3.86 \pm .03$, $4.11 \pm .02$ 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 choose to use grid size of 30m ×30m because it takes approximately 30 seconds for a Nokia N95 to acquire a GPS position and a person can travel 30 meters in 30 seconds in normal walking speed (1 m/s). Furthermore, GPS has an accuracy of 10 meters in outdoor environment, therefore a 30 × 30 grid could help us to cope with the GPS accuracy. We use formulations in [17] to convert between GPS and MGRS.

Signal Reconstruction Module

In this section, we will describe two sensing strategies (namely *projection method* and *raw-data method*) and how the central server performs reconstruction using the information collected by these two different sensing strategies. For reconstruction, we use the recently developed theory of compressive sensing [7]. For ease of explanation, we will explain the two sensing strategy with an example.

Let us consider the trajectory of two volunteers, A and B, along a section SG of a one dimensional street (see Fig. 3.1). Section SG contains three MGRS grid references: ℓ_1, ℓ_2 and ℓ_3. Suppose at time t_1 and t_2, volunteer A collects noise sample 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 is referring to the equivalent noise level $\mathrm{LA}_{\mathrm{eq,1s}}$ in that grid. The complete noise samples in section SG, during time t_1 and t_2 can be represented as a vector $x = [d(\ell_1, t_1), d(\ell_2, t_1), d(\ell_3, t_1), d(\ell_1, t_2), d(\ell_2, t_2), d(\ell_3, t_2)]^{\mathrm{T}}$, 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
$x_A = [d(\ell_1, t_1), 0, 0, 0, d(\ell_2, t_2), 0]^{\mathrm{T}}$ and
$x_B = [0, 0, d(\ell_3, t_1), d(\ell_1, t_2), 0, 0]^{\mathrm{T}}$ respectively.

In the projection method, A multiplies his measurement vector x_A with a projection vector
$\phi_A = [\phi_A^1, 0, 0, 0, \phi_A^5, 0]^{\mathrm{T}}$ (where ϕ_A^1, ϕ_A^5 are Gaussian distributed random numbers with mean zero and unit variance) and sends the projected value, $y_A = \phi_A^T * x_A$ to the central server (Note that the inner product $\phi_A^T x_A$ is known as a

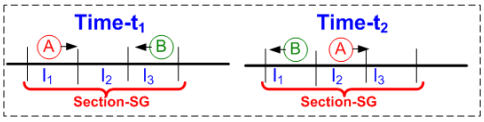


Figure 3.1: Illustration of urban sensing

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 vectors for A's data is regenerated as
$\phi_A = [\phi_A^1, 0, 0, 0, 0, 0; 0, 0, 0, 0, \phi_A^5, 0]^{\mathrm{T}}$, 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 $y = [y_A, y_B]^{\mathrm{T}}$ and forms the projection matrix, $\Phi = [\phi_A^{\mathrm{T}}, \phi_B^{\mathrm{T}}]$. The reconstruction proceeds in two steps. In the first step, the central server solves the following optimization problem:

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

where Ψ is a transform basis in which the noise profile x is compressible (We will give some evidence to show that the noise profile x is compressible in the DCT transform basis in the Appendix). 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 implementation we used a simplified "query to grid resolver", which is essentially a look up table, where we store the grid indices of the road segments (In our prototype implementation we only stored the grid indices of the road segments, where we conducted the experiments.). 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, resource (CPU, RAM and energy) consumption and noise-map generation, which demonstrates that Ear-Phone is an effective end-to-end system for measuring noise pollution from incomplete and random samples inherent in participatory sensing.

4.1 System Implementation

We have implemented the mobile phone components on two hardware platforms - the Nokia N95 and the HP iPAQ (Fig. 4.1). We choose Java as the programming language because it is platform independent. The various mobile components are implemented as separate application threads (e.g., GPS thread

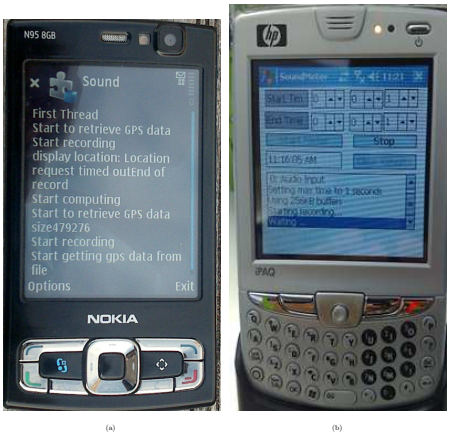


Figure 4.1: Screenshots of (a) Ear-Phone application running on Nokia N95 (b) Signal processing module running on HP iPAQ 6965.

and signal processing thread) in Java. We use the raw-data method (see Section 3.2) as the sensing strategy for the current Ear-Phone prototype. The server component consists of a MySQL database and PHP server-side scripting. We use MySQL database to store both the collected noise level data and the reconstructed noise level data. We used a PHP script to implement the server-side modules such as communication manager, GPS MGRS converter, noise signal reconstruction module, and query manager (see Section 2 for the description of these modules).

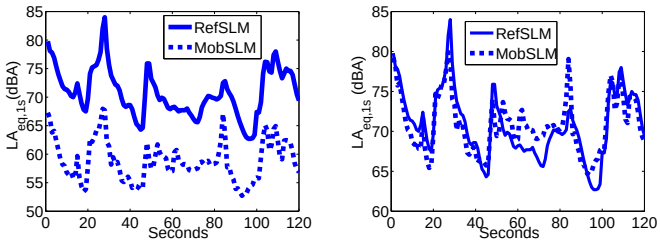
4.2 Measurement Accuracy

Recall from Section 3 (Eq.(3.4)) that we need to know the calibration offset to measure $\mathrm{LA}_{\mathrm{eq,T}}$. We determine this offset by conducting a simple calibration experiment. We use the freely available Audacity tool [4] to produce a chain of one second wide pulses of varying amplitudes and compare the responses of our algorithm (when computing $\mathrm{LA}_{\mathrm{eq,1s}}$) on a Nokia N95 or a HP iPAQ with the responses of a commercial sound level meter, Center-322 SLM [8] (see Fig. 4.2(a)). We use the mean of differences between the commercial meter (we refer it by RefSLM) and our mobile based SLM readings, as the offset. After adding the offset, we repeat the experiment and plot the responses in Fig. 4.2(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.

4.3 Resource Usage

Power Benchmarks

Power consumption of Ear-Phone is measured 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



(a) Time series (Before calibration) (b) Time series (After calibration)

Figure 4.2: Measurement Accuracy of Ear-Phone

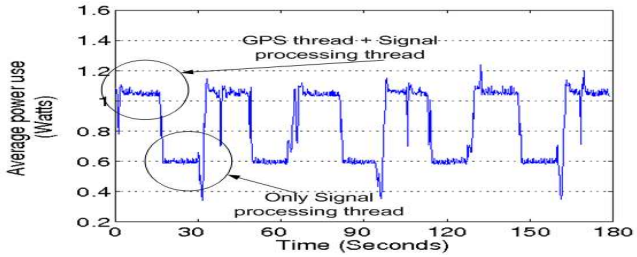


Figure 4.3: Power consumption of Ear-Phone on Nokia N95 for a three-minute period.

stores the results in the RAM. Fig. 4.3 shows the typical contribution of Ear-Phone to the overall energy budget during a three minute period with power consumption in Watts on the y axis. A 30-second cycle is evident from this plot, where the high power consumption during the first half of this cycle is due to concurrent execution of the GPS and signal processing threads, and in the second half, power consumption is due to the standalone execution of the signal processing thread. Note that due to resource limitations we can only get one GPS coordinate every 30 seconds on the Nokia N95 platform.

Running Ear-Phone with no other activities (such as net browsing, conversation etc) consumes approximately 0.667 Watt-hour energy on Nokia N95. If Ear-Phone runs for one hour each day, a user contributes only 15% of the phone battery lifetime to Ear-Phone application, which is quite a small percentage compared to the full battery lifetime (Nokia N95 comes with a 3.7V rechargeable battery of 1200 mA-Hour.). Also note that talktime on Nokia N95 is only 4 hours, whereas within the battery lifetime, Ear-Phone application can run approximately 7 hours continuously, therefore energy consumption overheaed of Ear-Phone is approximately half compared to the conversation [1].

[1]We believe the lifetime performance of Ear-phone on the HP iPAQ platform is similar to that of Nokia N95 platform because both platforms have similar hardware configuration. For example, both come with a 3.7V rechargeable battery of 1200 mA-Hour; Nokia N95 comes with 100 MB RAM while HP iPAQ comes with 64MB RAM; the processor of Nokia N95 runs at 330 MHz while that of iPAQ runs at 416 MHz.

Table 4.1: CPU and RAM usage

	CPU Load (mean) (%)	CPU Load (stdev) (%)	RAM (MB)
Phone Idle	2	0.79	32.86
Ear-phone (Signal processing thread)	5.22	3.03	38.06
Ear-phone (Sound processing & GPS thread)	98.15	11.40	38.28

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 power consumption of Ear-Phone with only the signal processing thread running. Finally, we repeat with both the signal processing and GPS threads running concurrently. The results in Table 4.1 show that Ear-Phone uses less than 40% of system RAM. Furthermore, Table 4.1 shows that the GPS thread is the dominating factor of Ear-Phone CPU consumption. We believe the Ear-Phone performance can be further improved because the current implementation has not been optimized.

4.4 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 is thus noisy, and Bainbridge Drive, which is a branch road that leads to a residential neighborhood and is hence much quieter. We reconstructed the hourly noise map for time periods (off peak: 14:00 - 15:00 and peak: 8:00 - 9:00) along these road segments. 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 4.4. 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 reconstruct 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 Bainbbridge Drive. Using the reconstructed $LA_{eq,T}$, 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. 4.5 and 4.6 show the impact of measurements included from a varying number of persons on the reconstruction accuracy during off-peak and peak hour 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. However, when we include data from multiple

persons, the reconstruction gradually reveals the contrast between the noisy and quiet street. Furthermore, after a certain threshold, increasing data contributors does not improve the reconstruction accuracy significantly. For example, comparing Fig. 4.5(c) and Fig. 4.5(d), it is evident that the reconstruction achieved by data from 4 people is similar to that from 6 people. A similar behavior can be seen in Fig. 4.6(c) and Fig. 4.6(d).

During these experiments, we simultaneously measured the $\mathrm{LA}_{\mathrm{eq,LT}}$ using our commercial sound level meters placed midway along Mogill Rd and Bainbridge drive. Comparing the reconstructed noise map with the commercial sound level meter readings, we find that we need measurements from at least 5 people during peak hour and from a minimum of 4 people during off-peak hour, for a reconstruction comparable to the commercial sound level meter. The result implies that the off-peak hour has a lower sampling requirements.

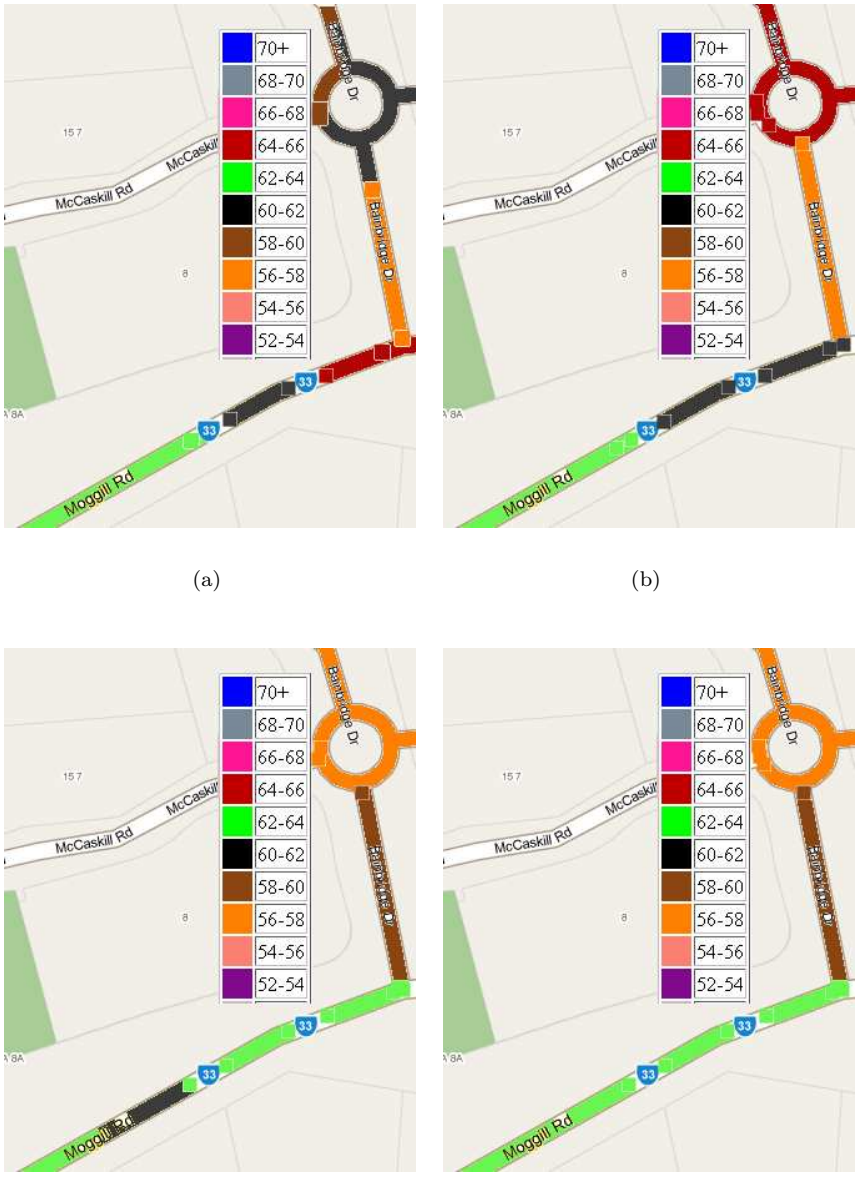
Figure 4.4: Data collection route

5 Simulation

Real experiments certainly provide valuable information. However, real experiments are not repeatable and conducting real experiments on a large scale is expensive and time consuming. We therefore conduct simulation experiments where factors such as the number and mobility patterns of volunteers, sensing strategies (see Section 3.2) etc. can be varied easily. In the following, we will first describe how we perform measurement campaigns to collect noise profiles which will be fed into the simulation as ground truth. This will be followed by a description on the simulation itself and performance evaluation in terms of reconstruction accuracies

5.1 Simulation Design

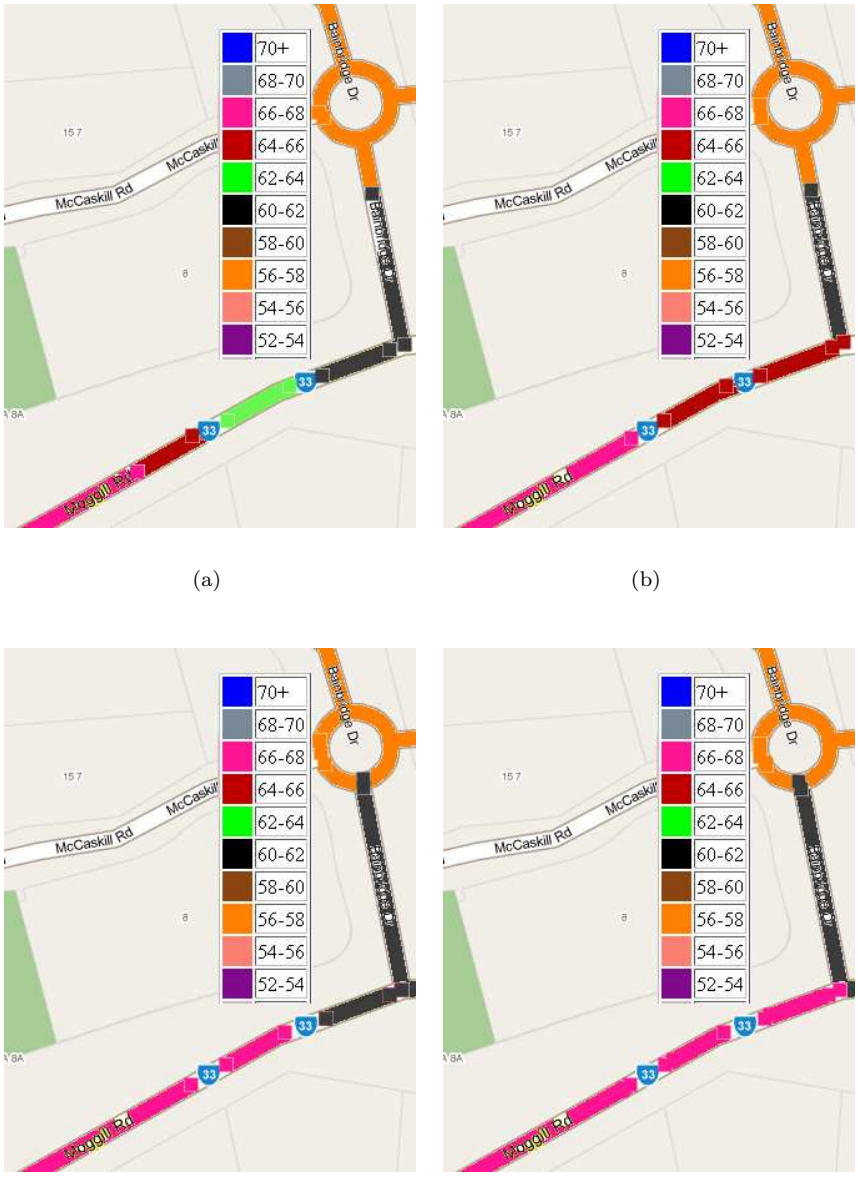
As in Section 4, we limit our consideration to noise measurements along a road, which can be modeled as a scalar field over a uniform 2-dimensional grid of cells with one spatial and one temporal dimension. We assume that each cell has a spatial width of D meters and a temporal width of T seconds. We use the ordered pair (i, j) to refer to the cell bounded by the spatial interval $[(i-1)D, iD]$



(a) (b)

(c) (d)

Figure 4.5: Noise map reconstruction during off peak hour (2:00pm-3:00pm) using data from (a) 1 person, (b) 2 persons, (c) 4 persons and (d) 6 persons.



(a) (b)

(c) (d)

Figure 4.6: Noise map reconstruction during peak hour (8:00am-9:00am) using data from (a) 1 person, (b) 3 persons, (c) 5 persons and (d) 7 persons.

and temporal interval $[(j-1)T, jT]$. Assuming that $i \in N_s = \{1, 2, ..., n_s\}$ and $j \in N_t = \{1, 2, ..., n_t\}$, 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 $\mathrm{LA}_{\mathrm{eq,T}}$ measured over each cell is almost constant. Now let $d(i, j)$ denote the equivalent noise level $\mathrm{LA}_{\mathrm{eq,T}}$ measured in cell (i, j), then a *noise profile* S is defined as the set of all $\mathrm{LA}_{\mathrm{eq,T}}$ 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. We conducted four experiments to collect $\mathrm{LA}_{\mathrm{eq,1s}}$ 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 $\mathrm{LA}_{\mathrm{eq,1s}}$ 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 equally spaced 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 all phones start and stop sampling at the same time. The MobSLMs measured $\mathrm{LA}_{\mathrm{eq,1s}}$ 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 $1.11\ \mathrm{ms}^{-1} = 4$ 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 $\mathrm{LA}_{\mathrm{eq,1s}}$ samples collected in the cells in $\tilde{W}$ to form a projection. Recall from Sect. 3 that a projection is essentially a linear combination of the data. The agent computes

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

where $d(i, j)$ is the $\mathrm{LA}_{\mathrm{eq,1s}}$ sample collected at cell (i, j) and $\eta(i, j)$'s (with

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

Table 5.1: Experimental settings for collecting the reference noise profiles

$(i,j) \in \tilde{W}$) 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 that it 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 $\mathrm{LA}_{\mathrm{eq,1s}}$ 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 square (RMS) reconstruction error by:

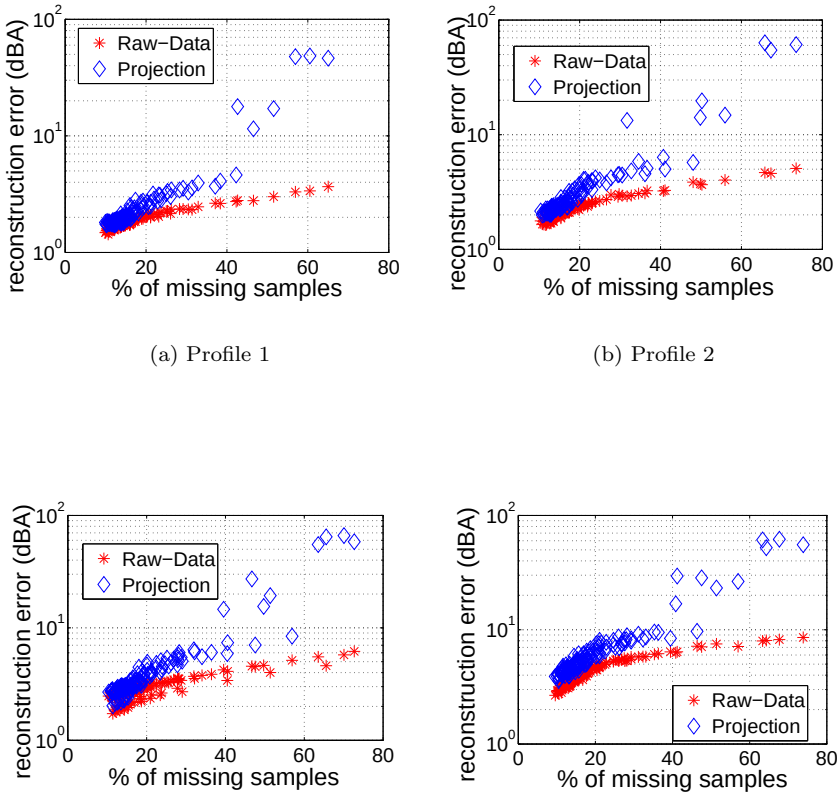
$$S_{rms} = \sqrt{\frac{1}{n_s \times n_t} \sum_{1\le i<n_s, 1\le j<n_t} \left(d\left(i,j\right) - \hat{d}\left(i,j\right)\right)^2} \tag{5.2}$$

5.2 Performance Evaluation

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 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. 5.1(a) to 5.1(d) we plot the reconstruction accuracy as a function of sampling requirements for our reference noise 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. However, when the the percentage of missing samples is small, this loss of information is small). Except for profile 4, Ear-Phone can reconstruct the profiles to within 3dBA error with 40% or less missing samples. The increase in sampling requirements from profile 1 to 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. 5.2. A total of 3 sections are shown in Fig. 5.1 for

different percentages of missing samples for the raw-data method. Note that the reconstruction is pretty accurate at the cell level.



(a) Profile 1

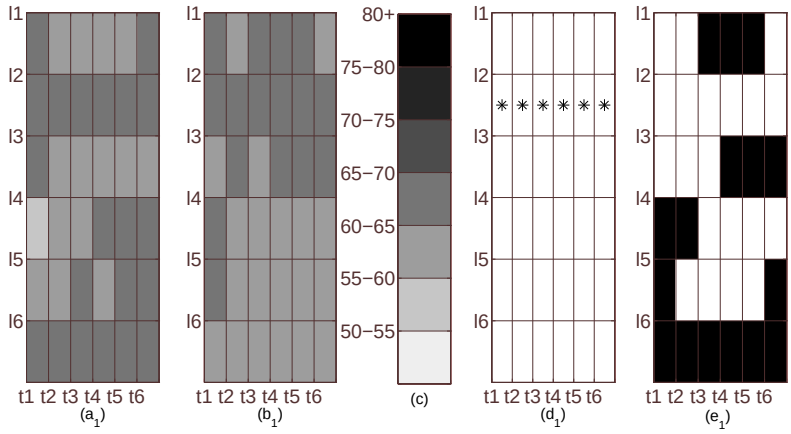
(b) Profile 2

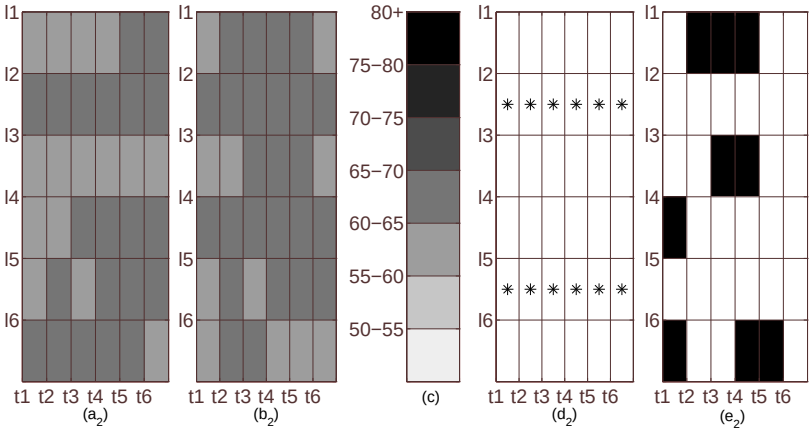
(c) Profile 3

(d) Profile 4

Figure 5.1: Percentage of missing data (x-axis) and its impact on reconstruction accuracy expressed in RMS error (y-axis).

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 $\mathrm{LA}_{\mathrm{eq,1s}}$ 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. 5.3 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 (what humans cannot perceive), the projection method and the raw data method reduce the communication costs by 30% and 20% respectively compared to the state-of-the-art sampling technique. However, for a high reconstruction error (an increased amount of missing information), the raw-data method is more communication efficient than the projection method.


I1
I2
I3
I4
I5
I6
t1 t2 t3 t4 t5 t6
(a1)
(b1)
80+
75–80
70–75
65–70
60–65
55–60
50–55
(c)
(d1)
(e1)


I1
I2
I3
I4
I5
I6
t1 t2 t3 t4 t5 t6
(a2)
(b2)
80+
75–80
70–75
65–70
60–65
55–60
50–55
(c)
(d2)
(e2)

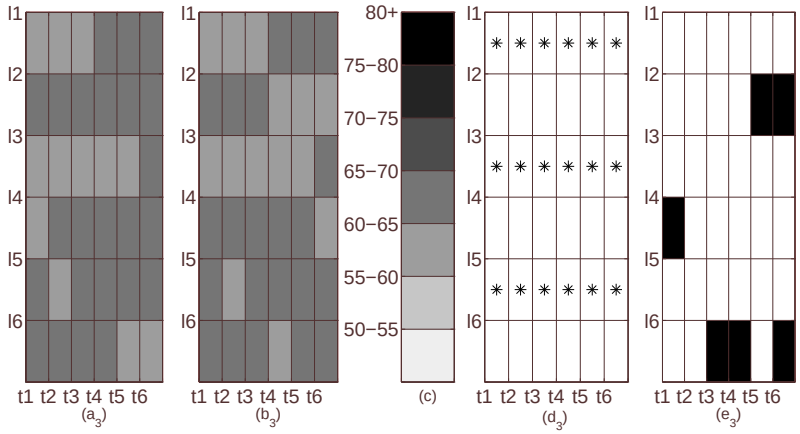


Figure 5.2: This figure shows the reconstruction performance at cell level. Each row of this figure consists of 5 sub-figures (a_i), (b_i), ..., (e_i) where $i = 1, .., 3$. Each row $(i = 1, 2, 3)$ shows the reconstruction of a section of the profile for a given percentage of missing data. The percentage of missing data 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 $\mathrm{LA}_{\mathrm{eq,1s}}$ 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.

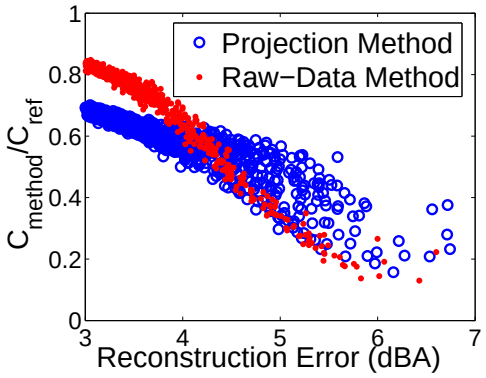


Figure 5.3: Reconstruction accuracy VS communication overhead

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 improve the trustworthiness of participatory sensing. However, we focus our attention on the following.

In [18], 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 [16] 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 [13] 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 [15] uses a traditional interpolation framework to estimate missing data, when data is obtained via crowdsourcing. In contrast, we 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. 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.

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 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% of 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, while our experiments were conducted with the phones held in the volunteer's palm. Our future work will examine the impact of the "carryin" position of mobile phones on the noise measurements. In addition, there is an opportunity to develop a context-aware application which will only sample ambient noise when the phone is in the right environment.

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] Ajdler et al. The Plenacoustic function, sampling and reconstruction. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, NY*, 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] Bajwa et al. Compressive wireless sensing. In *IPSN*, pages 134–142, 2006.

[6] J. Burke et al. Participatory sensing. In *Workshop on World-Sensor-Web (WSW06): Mobile Device Centric Sensor Networks and Applications*, 2006.

[7] E. Candés. Compressive sensing. In *Proc. of the Int. Congress of Mathematics*, 2006.

[8] Center Technology Corp. Center c322.

[9] DEFRA. Noise mapping england. `http://www.noisemapping.org/`.

[10] Department for Health and Environment of the City of Munich (Germany). Noise maps 2007. `http://tinyurl.com/`.

[11] 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.

[12] European Union. Future noice policy, com (96) 540 final. European Commission Green Paper, Nov 1996.

[13] A. Griffin and P. Tsakalides. Compressed sensing of audio signals using multiple sensors. In *EUSIPCO 2008*, Lausanne, Switzerland, Aug. 2008. to be published.

[14] International Electrotechnical Commission. Electroacoustics - sound level meters - part 2: Pattern evaluation tests, April 2003.

[15] A. Krause et al. Toward community sensing. In *IPSN*, pages 481–492, 2008.

[16] N. Maisonneuve, M. Stevens, M. E. Niessen, and L. Steels. Noisetube: Measuring and mapping noise pollution with mobile phones. In *ITEE 2009 - Information Technologies in Environmental Engineering*, pages 215–228. Springer Berlin Heidelberg, May 2009. Proceedings of the 4th International ICSC Symposium in Thessaloniki, Greece, May 28-29, 2009.

[17] National Geospatial-Intelligence Agency (NGA). Datums, ellipsoids, grids, and grid reference systems. *DMA TECHNICAL MANUAL.*

[18] S. Santini et al. 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.*

[19] S. Santini et al. First experiences using wireless sensor networks for noise pollution monitoring. In *Proc. of the REALWSN'08*, Apr. 2008.

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 ($k = 1, 2, ...$) 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.

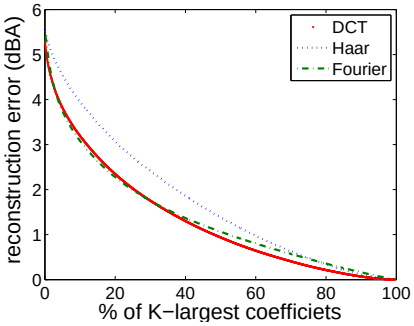


Figure 7.1: Compressibility of the noise profile