NoiseMap - Real-time participatory noise maps

Immanuel Schweizer
TU Darmstadt
Hochschulstrasse 10
64289 Darmstadt
schweizer@tk.informatik.tu-darmstadt.de

Roman Bärtl
TU Darmstadt
Hochschulstrasse 10
64289 Darmstadt
r_baertl@rbg.informatik.tu-darmstadt.de

Axel Schulz
TU Darmstadt
Hochschulstrasse 10
64289 Darmstadt
aschulz@tk.informatik.tu-darmstadt.de

Florian Probst
SAP Research
Bleichstrasse 9
64283 Darmstadt
f.probst@sap.com

Max Mühlhäuser
TU Darmstadt
Hochschulstrasse 10
64289 Darmstadt
max@tk.informatik.tu-darmstadt.de

Abstract

Noise pollution is a problem increasingly acknowledged by authorities and governments around the globe. However, creating noise maps with conventional methods is either inaccurate or very expensive. To increase the spatial and temporal data resolution a high number of sensors must be deployed.

In this paper we present results and prototypes based on participatory sensing leading to accurate, real-time noise maps. First we present NoiseMap, a application currently released for Android phones. NoiseMap gathers data on loudness and transfers it to the open urban sensing platform da_sense. da_sense allows users to access and control their data, generate real-time noise maps and data graphs. Public data is made available using either a web service or a JavaScript API.

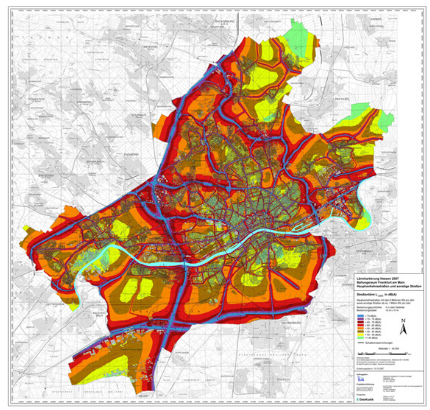


Figure 1. Noise Map Frankfurt, Germany

1 Introduction

Noise pollution is a ever increasing problem acknowledged by governments around the globe. To tackle noise pollution, the European Union has regulated that all member states have to gather noise data and create noise maps to efficiently develop and maintain measurements against noise pollution [4]. The Environmental Noise Directive requires noise levels to be assessed from road traffic, railways, major airports and industry. Since gathering data is expensive and the area that needs to be covered huge, the data is captured only at a small number of locations and the noise map is calculated using geographic models and simulation. Also, data is only gathered twice; the first run was conducted in 2007 and another run is planned for 2012 featuring more data points. Other countries are using comparable plans to create noise maps.

An example for a noise map for Frankfurt, Germany is shown in Figure 1[1]. While the data density seems to be high due to the simulation, the captured data is rather sparse. Noise emission is only measured at intersections next to the roads with traffic higher than six million cars per year, highlighted in blue. Additionally noise is measured next to three large industrial parks and railways. Other sources of noise pollution like small roads, road work etc. are not considered.

Over the past years, smartphones have seen high adoption rates. Smartphones are personal devices carried all day, equipped with different sensors, connected to the Internet and, most importantly, charged by their user. They are the perfect platform for sensing the environment using a participatory sensing approach [2]. While the sensor hardware differs between manufacturers and is limited by cost and space restrictions all smartphones, indeed almost all phones today, are equipped with a microphone and GPS transceiver. This enables there use as noise meter. Combining the sheer number of potential noise sensors to a participatory network allows for the creation of high density real-time noise maps. Furthermore, the ubiquity and the mobility make them perfect candidates for complementing the coverage of traditional sensor networks [7].

In this paper we present *NoiseMap*, a participatory sensing application for measuring noise. The measurements are

[1] Picture Source: http://www.hlug.de/?id=525

send to a urban sensing platform named *da_sense* where they are processed. Noise maps, graphs and public data sets are available at *www.da-sense.de*. Participatory noise sampling has already been considered, most notable in the Brussense project [8], but many challenges remain before these approaches are brought to their full potential.

Among other aspects we address in the future work section, we will put emphasis on the involvement of humans for delivering predictions, ratings and evaluations for enriching the collected sensor information, to deal with the increasing amount of data provided by sensor sources and participatory sensing applications. We believe that approaches relying on human cognition for collecting additional information will complement the existing participatory sensing approaches.

The paper is organized as follows. Section 2 will introduce the NoiseMap application. Section 3 will focus on da_sense. Future work is categorized and possible solutions are given in Section 4. We will be looking at related work in Section 5 and a short wrap-up is given in Section 6.

2 NoiseMap

Today noise maps are created using high cost noise meters measuring high accuracy data at predefined locations. This data is extrapolated using landscape models and simulation tools. Even with the use of accurate landscape models and simulations this will lead to inaccurate maps due to the small number of data points and missing noise sources. Increasing the data points using more noise meters and more man power is too expensive and does not scale.

By applying participatory sensing, we want to enable real-time noise maps. Smartphones, in fact most mobile phones, come equipped with GPS and a microphone. NoiseMap is a mobile application, first released for Android[2], that transforms a smartphone into a mobile noise meter.

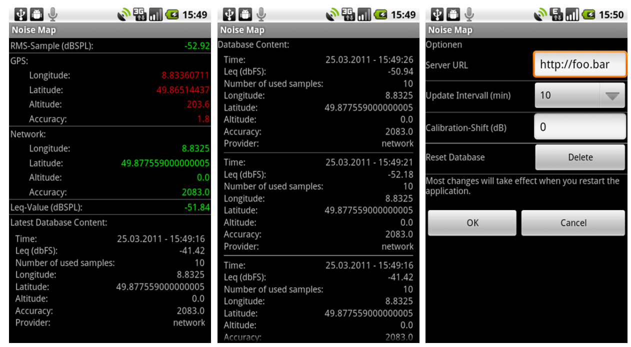
NoiseMap samples the incoming sound to translate the discrete digital signal to a dB full scale (dbFS) value. On this scale 0 dBFS stands for the maximum level the microphone can measure and all other dBFS values are negative.This value has to be translated to dBSPL were SPL stands for sound pressure level. SPL is a reference system relative to a given sound pressure value. This value is usually $20\mu PA$ which is considered as the threshold of human hearing.

To translate dBFS to dBSPL a calibrated value has to be added so $dBSPL = dBFS + x_{cal}$. NoiseMap has a build in calibration tool. Given a constant pink noise x_{cal} is calculated by the app. To increase participation calibration must be as easy as possible.

Currently not implemented in NoiseMap is the frequency-dependent hearing of the human ear. A signal of the same pressure is interpreted different for changing frequency levels. To normalize the value, a so called A-weighting has to be applied to the signal. This is not yet implemented but will be in the short future.

Since sound is dynamic, noise measurements are averaged over time using a root mean square (RMS) over 11025 samples, which corresponds to half a second at $22050Hz$. These RMS samples are then used to calculate the L_{eq}. L_{eq} stands for long term equivalent sound level and is commonly used for long time measurements. It represents the constant sound pressure level equivalent to the samples in the given time range. In NoiseMap this time range depends on the current speed of the smartphone as a higher temporal resolution is needed at higher speeds.

NoiseMap also requests the current location (latitude, longitude, altitude, accuracy). Android is able to provide location using cell tower or WLAN triangulation or GPS denoted by the provider string. GPS is always used if available. As it might not always be available, especially indoors, the triangulation is used as an indicator to allow for measurements. NoiseMap will tag the data such that triangulated locations can easily be filtered. All data is shown in the main screen of NoiseMap as shown in Figure 2(a).



(a) Main screen (b) Database view (c) Options

Figure 2. NoiseMap GUI examples

The resulting samples are transmitted to the web service using JavaScript Object Notation (JSON) a lightweight data-interchange format. All current measurements not yet transmitted to the server can be viewed as shown in Figure 2(b). The interval of the sample transmission as well as the endpoint of the web service can be changed in the options view (cmp. Fig. 2(c)). Using the default settings, the data is transmitted to the da_sense platform. To date approximately $27,000$ data points have been gathered using 7 different phones covering an area of about $1km^2$. The real-time noise maps are available at www.da-sense.de. We will introduce da_sense and show an example map in the next section.

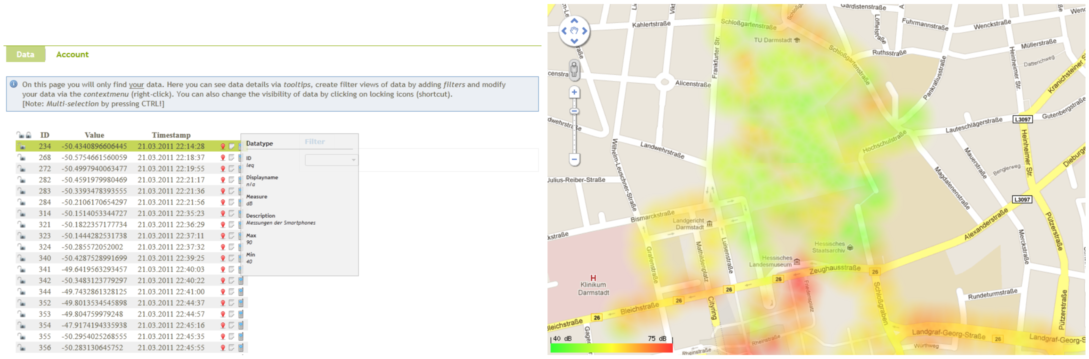
3 da_sense

da_sense is a open urban sensing platform collecting data from participatory as well as wireless sensor networks. While NoiseMap provides noise pollution, CO_2, CO, and other environmental gases are collected using wireless sensors. The data from NoiseMap is transmitted to a web service and written to the da_sense database. A overview of the da_sense architecture is given in Figure 3.

What makes da_sense important for NoiseMap are three key aspects:

- Control over collected data
- Incentive through information

[2] The current beta version can be downloaded at `http://www.tk.informatik.tu-darmstadt.de/de/research/smart-civil-security/noisemap/`



(a) User management (b) Map visualization

Figure 4. da_sense portal

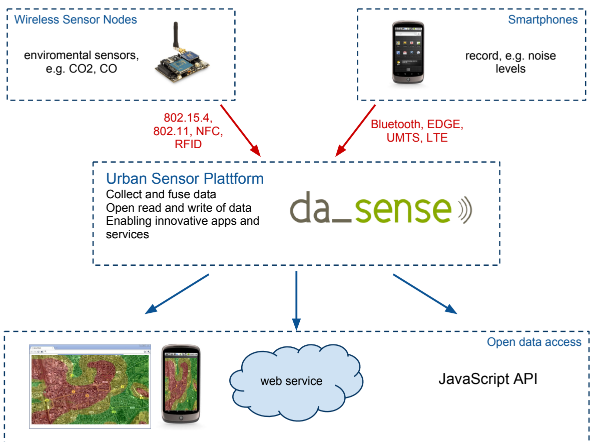


Figure 3. Overview da_sense

- Open data access

To enable user control over all data collected da_sense provides user management that is working closely with the NoiseMap application. All data provided by NoiseMap is linked to a user account and can be made private. Private data is only visible to the user after login. We believe that giving the user the right to control their data is important for data privacy but is also important for data quality. A user might want to use NoiseMap to measure noise pollution in different scenarios, e.g. loudness in their car. Such measurements are not useful for a global noise map and can be filtered by the user. Private measurements ensure that NoiseMap is not only a research tool but a versatile noise meter. Figure 4(a) shows an excerpt of the user management of da_sense. A small lock indicates private data.

Even though the data is private a user might be concerned as data is stored on our servers. NoiseMap can be easily configured using the in-app options dialog to report to any web service capable of parsing the JSON string. There is no need to change the application. This allows full control over personally collected data.

Another important aspect of da_sense is that all data is visualized using maps and graphs. Information as incentive means that users are given direct feedback about all their data. There is the map that can be filtered to show any information for the user. It is also available for the public without registration to get an idea what da_sense is about and increase the incentive to become a part of the participatory network. The map is also important for authorities to show real-time noise maps using thousands of data points. The visualization is calculated as a heatmap overlay (cmp 4(b)) for Google Maps even though other APIs, e.g. OpenStreetMaps, Bing Maps, Ovi Maps, will be supported in the future. Since participatory sensing needs to scale to be able to generate critical mass data points are clustered depending on the zoom level. The clustering algorithm used is DBSCAN [6] as it provides a good and fast geographic clustering. Every data point is selectable and the tool tip gives additional information to the sensor node and average noise level. The graph for each sensor node is also accessible from the tool tip. It allows for a quick overview of all the data collected, average loudness and a graph showing the data over time. It is also possible to select a single location to show a 24hour noise graph of the area.

NoiseMap is used on the mobile phone and might want to access all of his measurements. Therefore we also made a da_sense Android application available (cmp. Fig. 5). The next step is to combine the da_sense application with the NoiseMap application to allow one-click access to all the user data and visualization.

Open data access is really important as it is in our opinion essential for the success of any participatory project. It must be easy to read and write to the urban sensing platform. da_sense allows data access on different levels. The easiest access is using the web page. Raw data access is provided by a web service to access all public and the own private data set to the members. Based on the web service a JavaScript API is available for fast mashups. Using the JavaScript API it will be possible to generate map mashups where the heatmap is calculated and rendered at the client-side using the newest HTML5 standard to decrease the load on the server.

A small example for a possible API call is:
www.da-sense.de/api.php?fields[]=ID,data&fields[]=value,

data&fields[]=ID,locations&fields[]=longitude,locations &limit=2&filter[]=ID,data,70,100&order[]=value,data.
The corresponding JSON response is given below.

```
{
 querytime: 0.0013
  result: [
    {
      ID: 76
      value: −50.5794334411621
      locations: {
        ID: 3
        longitude: 8.6544732
      }
    }
  ]
}
```

Listing 1. Example Response

Open access to data is important as it will spur participation and allows others to make use of the data. It also goes the other way as we want to use external data to enhance the platform.

Summarized, da_sense is the default platform collecting all the data from NoiseMap. It adds important elements, such as: Control over collected data, incentive through information and open data access.

4 Future Work

NoiseMap and da_sense do provide a stable set of tools for real-time noise maps. Extending the given functionality we will now highlight areas of planned or already ongoing future work.

NoiseMap Application. The next step for NoiseMap is to use frequency weighted sampling (A-weighted) and generate the L_{Aeq} (A-weighted equivalent sound levels) that represents the characteristics of the human ear. We are currently also investigating different other microphone properties, e.g. linearity and noise reduction techniques. It is also important to think about the measurement context of the smartphone. For example if the phone is located in the pocket the measurements are skewed. This should be detected and represented in the meta information of the sensor. Approaches for this have already been discussed [9] and might be integrated with NoiseMap.

Official Noise Map. To calculate noise maps that are compliant with governmental regulations noise samples have to be aggregated in specific intervals. There are different norms defined, e.g. L_{den} (day-evening-night equivalent level) and L_{night} (night equivalent level) [4]. These aggregates will be implemented to deliver real-time noise maps that comply with government regulations.

Incentive systems. While incentive through information is an interesting concept we will take a close look at different incentive systems. Checkins are commonly used in location-aware online social networks to spur participation. We will also take a look at ranking systems and how points can be assigned to different actions, e.g. area covered, time, data points, etc. to build daily, weekly, or monthly rankings.

Data quality. Microphones built into smartphones are not meant to be noise meters. There quality is inferior to expensive sensors. Some of the issues that are directly related to the microphone are already covered above. To ensure data quality erroneous measurements must be filtered. Algorithms to detect data outliers cannot easily be applied since spikes in noise might easily be mistaken for erroneous measurements. Detecting outliers might identify changes happening over time or interesting short term phenomena. A thorough data analysis algorithm is needed to guarantee accurate results. A important idea is to use the data quantity to increase data quality.

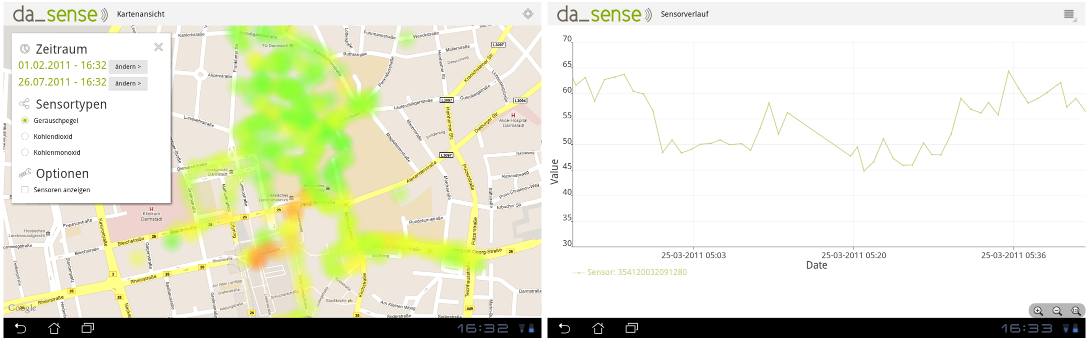
Calibration. An important aspect of any sensor application is calibration. Microphones differ between phone models but also between phones of the same model. Calibration can, therefore, be done per phone model or per phone. We favor an approach were calibration data is provided per phone model, but each phone can be calibrated later to deliver even more accurate results.

Putting human minds in the center of data fusion and analysis. The often sub-optimal coverage of static sensor networks calls for approaches, based on participatory sensing. Humans carrying sensors can create a dynamic network gathering additional information. However, participatory sensing leads to new problems like duplicates and high variance in data quality. While in many participatory sensing approaches, the human user only carries the sensors (built into his mobile device), approaches focusing on the human cognition, put the focus on the human senses themselves. This opens a wide range of possibilities for enriching the collected sensor information with predictions, ratings and evaluations.

Combining heterogeneous information sources. While additional information provided by crowds adds a new layer of information, automatic feature extraction is needed to cope with the amounts of data available. To extract high level features or add more information we envision the composition of heterogeneous information sources, e.g. different open services. A main aspect of da_sense is open access to data and we want to use freely available data to enhance the platform. An example is the extraction of traffic data and the combination with the noise data we provide. If we can interfere that noise is not produced by an increase in traffic there might be a phenomenon that needs further analysis by human observers. This could then also be combined with data from location-aware social networks. All of this leads to a better feature extraction and high-level context on top of the raw noise data.

Dynamic Sensor Description Language. For acknowledging the change in quality of sensors and number of sensors involved in ad hoc networks a dynamic way of describing the changing accuracy or quality is needed. Furthermore, a description of the semantics about the sensor and its provided information is needed to allow context inferences. This is currently not reflected in sensor description languages like SensorML.

Applications. da_sense and NoiseMap as well as other participatory applications span across different application domains. Areas where spatial information generated via sensors is important is disaster response and urban management. Here dynamic situations must be addressed, making the us-



(a) Map visualization

(b) Graph visualization

Figure 5. da_sense Android app

age of static sensor networks difficult. Crowdsourcing can be used to identify possible hotspots quickly, hence shortening the reaction time significantly.

5 Related Work

Due to the increasing adoption of smartphones the concept of participatory sensing has become increasingly popular over the past years [2, 5, 1].

It has been used for different applications, e.g. characterizing people movement [11], health [10] and many others. An exhaustive overview over participatory sensing applications is given in [3]. Participatory sensing is also used to build noise maps. Most notable is NoiseTube [8] developed by Matthias Stevens and Ellie D'Hondt and embedded in the Brussense project. NoiseTube is very strong when it comes to sound processing as their output is already frequency weighted. The problem is that all their data is not freely available and there is no incentive for the user. He is not allowed to access any real-time information and noise maps are only static.

6 Conclusion

Accurate noise maps complying with governmental regulations are costly to create and suffer from low spatial and temporal data resolution. Participatory sensing is a solution to increase the data resolution. In this paper we presented NoiseMap a participatory application gathering noise pollution. We have also presented da_sense a open urban sensing platform. da_sense adds control over gathered data, incentive through information, and open data access to NoiseMap.

We have also presented different future problems identified for NoiseMap but applicable to participatory sensing applications in general.

7 References

[1] T. Abdelzaher, Y. Anokwa, P. Boda, J. Burke, D. Estrin, L. Guibas, A. Kansal, S. Madden, and J. Reich. Mobiscopes for Human Spaces. *IEEE Pervasive Computing*, 6:20–29, April 2007.

[2] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. Srivastava. Participatory sensing. In *World Sensor Web Workshop*, pages 1–5, 2006.

[3] D. Christin, A. Reinhardt, S. Kanhere, and M. Hollick. A Survey on Privacy in Mobile Participatory Sensing Applications. *Journal of Systems and Software*, 2011.

[4] E. Directive. The Environmental Noise Directive (2002/49/EG). *Official Journal of the European Communities*, 2002.

[5] P. Dutta, P. M. Aoki, N. Kumar, A. Mainwaring, C. Myers, W. Willett, and A. Woodruff. Common Sense: Participatory urban sensing using a network of handheld air quality monitors. In *Proceedings of the 7th ACM SenSys*, pages 349–350, 2009.

[6] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining*, pages 226–231, 1996.

[7] M. Jiang and W. L. McGill. Human-Centered Sensing for Crisis Response and Management Analysis Campaigns. In *Proceedings of the 7th International Conference on Information Systems for Crisis Response and Management*, 2010.

[8] N. Maisonneuve, M. Stevens, and B. Ochab. Participatory noise pollution monitoring using mobile phones. *Information Polity*, 15(1):51–71, 2010.

[9] E. Miluzzo, M. Papandrea, N. Lane, H. Lu, and A. Campbell. Pocket, Bag, Hand, etc.-Automatically Detecting Phone Context through Discovery. In *Proceedings of the 1st International Workshop on Sensing for App Phones*, 2010.

[10] S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin, and M. Hansen. Image browsing, processing, and clustering for participatory sensing: lessons from a DietSense prototype. In *Proceedings of the 4th Workshop on Embedded networked sensors*, pages 13–17, 2007.

[11] L. Vu, K. Nahrstedt, S. Retika, and I. Gupta. Joint bluetooth/wifi scanning framework for characterizing and leveraging people movement in university campus. In *Proceedings of the 13th ACM International conference on Modeling, analysis, and simulation of wireless and mobile systems*, pages 257–265, 2010.