CyberGIS-Enabled Urban Sensing from Volunteered Citizen Participation Using Mobile Devices

Junjun Yin, Yizhao Gao, and Shaowen Wang

Abstract Environmental pollution has significant impact on citizens' health and wellbeing in urban settings. While a variety of sensors have been integrated into today's urban environments for measuring various pollution factors such as air quality and noise, to set up sensor networks or employ surveyors to collect urban pollution datasets remains costly and may involve legal implications. An alternative approach is based on the notion of volunteered citizens as sensors for collecting, updating and disseminating urban environmental measurements using mobile devices. A Big Data scenario emerges as large-scale crowdsourcing activities tend to generate sizable and unstructured datasets with near real-time updates. Conventional computational infrastructures are inadequate for handling such Big Data, for example, designing a "one-fits-all" database schema to accommodate

J. Yin • Y. Gao
CyberGIS Center for Advanced Digital and Spatial Studies, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

CyberInfrastructure and Geospatial Information Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Department of Geography and Geographic Information Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
e-mail: jyn@illinois.edu

S. Wang (✉)
CyberGIS Center for Advanced Digital and Spatial Studies, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

CyberInfrastructure and Geospatial Information Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Department of Geography and Geographic Information Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Department of Urban and Regional Planning, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
e-mail: shaowen@illinois.edu

P. Thakuriah et al. (eds.), *Seeing Cities Through Big Data*, Springer Geography,
DOI 10.1007/978-3-319-40902-3_5

diverse measurements, or dynamically generating pollution maps based on visual analytical workflows.

This paper describes a CyberGIS-enabled urban sensing framework to facilitate the volunteered participation of citizens in sensing environmental pollutions using mobile devices. Since CyberGIS is based on advanced cyberinfrastructure and characterized as high performance, distributed, and collaborative GIS, the framework enables interactive visual analytics for big urban data. Specifically, this framework integrates a MongoDB cluster for data management (without requiring a predefined schema), a MapReduce approach to extracting and aggregating sensor measurements, and a scalable kernel smoothing algorithm using a graphics processing unit (GPU) for rapid pollution map generation. We demonstrate the functionality of this framework though a use case scenario of mapping noise levels, where an implemented mobile application is used for capturing geo-tagged and time-stamped noise level measurements as engaged users move around in urban settings.

Keywords Volunteered Geographic Information • Urban sensing • Noise mapping • CyberGIS • Mobile devices

1 Introduction

In today's urban environments, various pollution problems have become significant concerns to people's health and well-being. Being able to monitor and measure the status of environmental pollution with high spatiotemporal resolution for producing accurate and informative pollution maps is crucial for citizens and urban planners to effectively contribute to decision making for improving living quality of urban environments. Traditionally, government agencies are responsible for measuring and collecting urban pollution data, which is done either by employing surveyors with specialized equipment or by setting up monitoring networks. For example, under the EU environmental noise directive (2002/49/EC) (Directive 2002), some cities commenced the installation of permanent ambient sound-monitoring networks. This approach is, however, subject to several limitations. For instance, it is often costly to build such sensor networks and hire surveyors. Furthermore, such sensors are statically placed and each can only cover an area or space of certain size. The sensor measurements themselves are usually sampled and aggregated for a period of time resulting in low update frequency.

Due to these limitations, alternative approaches have been investigated including the utilization of citizens as sensors to contribute to collecting, updating and disseminating information of urban environments, also known as crowdsourcing (Howe 2006; Goodchild 2007). In particular, some previous studies have explored the idea of encouraging participatory noise monitoring using mobile devices. For example, the NoiseTube mobile application utilizes the combination of microphone and embedded GPS receiver to monitor noise pollution at various sites of a city

(Maisonneuve et al. 2009, 2010). This effort also showed some promising results regarding the effectiveness of participatory noise mapping. Compared to the traditional noise monitoring approach that relies on centralized sensor networks, the mobile approach is less costly; and with collective efforts, this approach using humans as sensors can potentially reach a significantly larger coverage of the city.

With integrated environmental sensors,[1] the new generation mobile devices can instrument comprehensive environmental properties, such as ambient temperature, air pressure, humidity, and sound pressure level (i.e., noise level). However, when the involvement of a large number of participants engaging in crowdsourcing activities becomes a realization, a large volume of, near real-time updated, unstructured datasets are produced. Conventional end-to-end computational infrastructures will have difficulties in coping with managing, processing, and analyzing such datasets (Bryant 2009), requiring support from more advanced cyberinfrastructure regarding data storage and computational capabilities.

This paper describes a CyberGIS-enabled urban sensing framework to facilitate the participation of volunteered citizens in monitoring urban environmental pollution using mobile devices. CyberGIS represents a new-generation GIS (Geographic Information System) based on the synthesis of advanced cyberinfrastructure, GIS and spatial analysis (Wang 2010). It provides abundant cyberinfrastructure resources and toolkits to facilitate the development of applications that require access to, for example, high performance and distributed computing resources and massive data storage. This framework enables scalable data management, analysis, and visualization intended for massive spatial data collected by mobile devices. To demonstrate its functionality, we focus on the case of noise mapping. In general, this framework integrates a MongoDB[2] cluster for data storage, a MapReduce approach (Dean and Ghemawat 2008) to extracting and aggregating noise records collected and uploaded by mobile devices, and a parallel kernel smoothing algorithm using graphics processing unit (GPU) for efficiently creating noise pollution maps from massive collection of records. This framework also implements a mobile application for capturing geo-tagged and time-stamped noise level measurements as users move around in urban settings.

The remainder of this paper is organized as follows: Section "Participatory Urban Sensing and CyberGISParticipatory Urban Sensing and CyberGIS" describes the related work in the context of volunteered participation of citizens in sensing urban environment. We focus on the research challenges in terms of data management, processing, analysis, and visualization. In particular, CyberGIS is argued to be suitable for addressing these challenges. Section "System Design and Implementation" illustrates the details of the design and implementation of the CyberGIS-enabled urban sensing framework. Section "User Case Scenario" details a user case scenario for noise mapping using mobile devices. Section "Conclusions and Future Work" concludes the paper and discusses future work.

[1] http://developer.android.com/guide/topics/sensors/index.html

[2] http://www.mongodb.org/

2 Participatory Urban Sensing and CyberGIS

To monitor and study urban environmental pollution, data collection and processing are two major steps in our framework. In terms of data collection from citizens engaged in reporting noise levels around a city, researchers found a low cost solution of using the microphone of mobile device to record and calculate the sound levels, such as the SPL android application.[3] Combining the embedded GPS receiver on mobile devices, the noise-level measurements are geo-tagged with geographic locations, which allow researchers to generate heatmap like noise maps (Maisonneuve et al. 2009; Stevens and DHondt 2010). In addition to appending the geo-location as a tag to the measurement, other applications, such as NoiseTube also encourages the appending of environmental tags, such as the type of noise (e.g., cars and aircraft) as additional attributes to the records. To encourage participants to contribute to the sensing activity as much as possible, such measurement can even take place whenever a user posts a social media message using their mobile device. However, since the availability of sensors varies in different devices, the collection of users' measurements can seem to be "unstructured", which makes it difficult to design a "one-fits-all" database schema to accommodate all the user inputs. Furthermore, when a large number of citizens participate in sensing urban environments using mobile devices simultaneously, it poses challenges for efficient data management, processing and visualizations. A Big Data scenario emerges in large-scale crowdsourcing activities, which requires an innovative system to support scalable data handling, such as data storage with flexible data schema and efficient database querying. Many applications, such as NoiseTube, use a relational database for data storage and processing. Relational databases, with a rigidly defined, schema-based approach, make it difficult to incorporate new types of data (Stonebraker et al. 2007) and achieve dynamic scalability while maintaining the performance users demand (Han et al. 2011).

The large volume and dynamic nature of the datasets also causes visualization problems for noise map generation. Existing GIS libraries, such as heatmap.js[4] and map servers (e.g., GeoServer[5]) provide inadequate support for this type of data. In particular, the ability to perform visualization based on customized queries regarding, e.g., a specified time window or an individual user (or a particular group of users) from the accumulated large volume of data, is limited in the existing applications. To embrace the characteristics of Big Data from large-scale crowdsourcing activities and accommodate the geographic attributes of the user generated content, CyberGIS integrates high performance computing resources and scalable computing architecture to support data intensive processing, analysis and visualization (Ghosh et al. 2012; Wang et al. 2012). CyberGIS represents a new-generation of GIS based on the synthesis of advanced cyberinfrastructure

[3] http://play.google.com/store/apps/details?id=com.julian.apps.SPLMeter&hl=en

[4] http://www.patrick-wied.at/static/heatmapjs/

[5] http://geoserver.org/

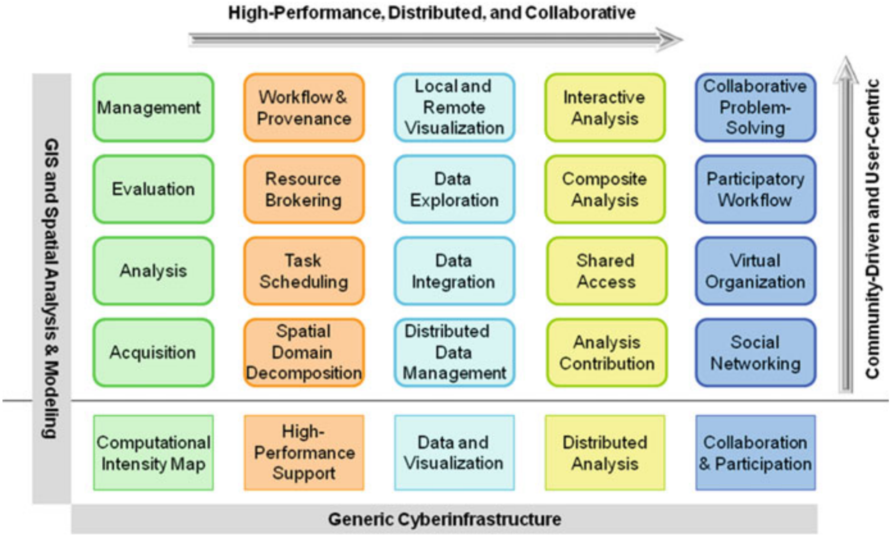


Fig. 1 An overview of the CyberGIS architecture. *Source*: Wang et al. (2013a)

GIS and spatial analysis (Wang 2010). As illustrated in Fig. 1 for the overview of the CyberGIS architecture, CyberGIS provides a range of capabilities for tackling the data and computation-intensive challenges, where the embedded middleware can link different components to form a holistic platform tailored to specific requirements.

In particular, our framework utilizes several components within this architecture. In the "distributed data management" component, we deploy a MongoDB cluster over multiple computing nodes for monitoring data intake and storage, which is scalable to the growth of collected data volume. Compared to a relational database, the NoSQL database supports more flexible data models with easy scale-out ability and high performance advantages (Han et al. 2011; Wang et al. 2013b). In the "high performance support" layer, we rely on the MapReduce functionality of the MongoDB cluster for data processing, such as individual user trajectory extraction, which is used to visualize the pollution exposure to a particular participant; and aggregation of data provided by all participants to a 1-h (this value is defined for the ease of implementation and can be changed according to user specifications) time window. This is then used to dynamically produce noise maps for the monitored environment. And finally, in the "data and visualization" layer, we apply a parallel kernel smoothing algorithm for rapid noise map generation using GPUs. Specific design and implementation details will be discussed in the following section.

3 System Design and Implementation

The framework is designed and implemented to include two main components: a dedicated mobile application (for Android devices) for participants and a CyberGIS workflow for data management, processing and pollution map generation. A diagram for the overall architecture is shown in Fig. 2. For this framework, we employ a service-oriented architecture for the integration between mobile devices and a CyberGIS platform. Specifically, the mobile application utilizes the combination of GPS receivers and environmental sensors on mobile devices to produce geo-tagged and time-stamped environmental measurements. In addition, this application provides a background service that allows user to choose to store or append the measurement to other apps a user is interacting with in the mobile device. It is up to participants to decide when to upload their records to the CyberGIS platform via the implemented RESTful (Representational state transfer) web service interface. CyberGIS workflow filters and parses the input data (into JSON[6] format) and stores them into the MongoDB cluster. It also extracts a trajectory of each individual participant to visualize the pollution exposure along the trajectory. For pollution map generation from the measurements that are uploaded by all of the participants, the data aggregation process is carried out using a specified time window. A pollution map is dynamically generated as a GeoTIFF[7] image via a parallel kernel smoothing method using GPU, which will be displayed as a map overlay on top of the ESRI world street map.[8]

3.1 CyberGIS Workflow

The workflow first filters out invalid data records (e.g., records without valid coordinates) and then parses each record as a JSON object before saved to the MongoDB cluster. The MongoDB cluster is chained in a master-slave style in order to achieve scalability as datasets are accumulated into significant size, which is one of the significant advantages over the existing relational databases. Another advantage brought by the MongoDB cluster is the embedded mechanism for performing MapReduce tasks. Since there is no predefined data schema and the input data are simply raw documents with the only structure of <key, value> pairs, the MapReduce function can efficiently sort the "unstructured" records based on the specified keys, e.g. timestamp, unique user id or even geographical coordinates (or a combination of these). More importantly, the data are stored in a distributed fashion, meaning multiple instances of computing nodes can perform such tasks simultaneously, which is otherwise nearly impossible for conventional database

[6] http://json.org/

[7] http://en.wikipedia.org/wiki/GeoTIFF

[8] http://www.esri.com/software/arcgis/arcgisonline/maps/maps-and-map-layers

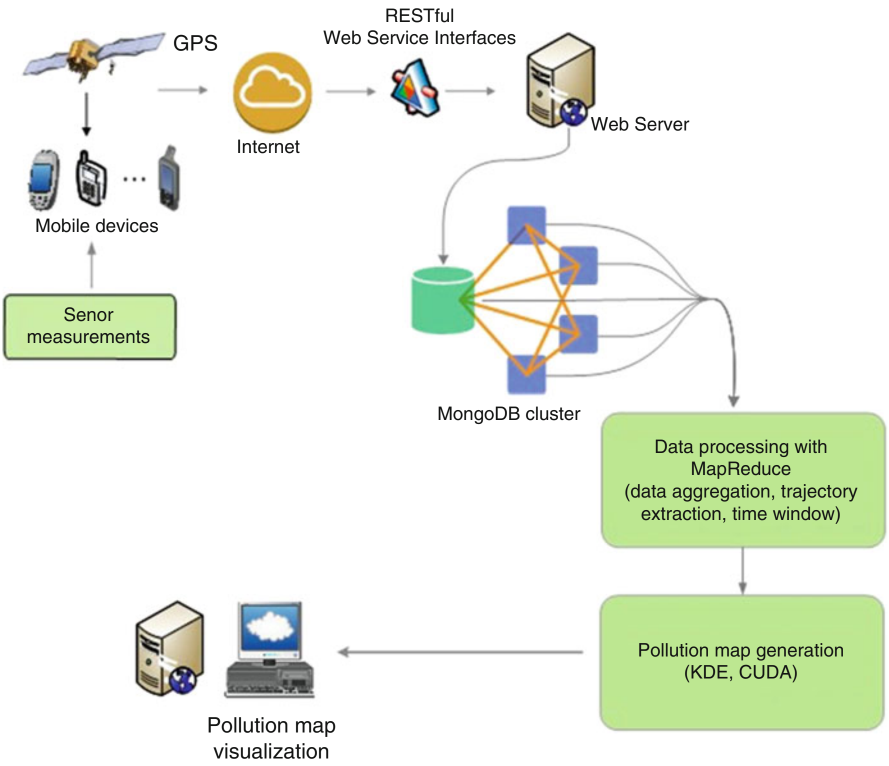

GPS
RESTful
Web Service Interfaces
Web Server
Internet
Mobile devices
Senor measurements
MongoDB cluster
Data processing with MapReduce (data aggregation, trajectory extraction, time window)
Pollution map generation (KDE, CUDA)
Pollution map visualization

Fig. 2 The overall architecture of the framework

queries. To visualize the pollution exposure to each individual user, we utilize MapReduce to simply use the device ID as the key to extract the trajectory of a specific user from the database.

In producing pollution maps for the measured environmental properties, many existing applications provide the visualization based on individual points. However, such an approach is subject to two major challenges. (1) When dealing with a massive collection of measurements in the form of geographical points, the visualization process will experience longer processing time, which may not be able to provide an effective response to dynamic user requests. (2) Since the framework is intended for a large group of people's collected measurements, visualizing the measurements (even calibrated) collected at the same location (or locations nearby) at different times will provide confusing results. In this regard, we aggregate all users inputs based on a predefined time window and kernel band-width and calibrate according to factors such as the sound decay distance. To simplify the process, we define a 1-h time window and 50-m kernel bandwidth. In other words, we assumed that each measurement will last for 1 h and covers an area of 50-m radius. The value of this assumption needs to be more carefully determined based on the real-world measurements once there are enough data collected by multiple users. The aggregation is implemented also using the MapReduce method, where the device ID is treated as the map key and the reduction process is based on the timestamps that fall in a specified 1-h time window.

The pollution map is dynamically generated by using a kernel smoothing method. Kernel smoothing is used to estimate a continuous surface of environmental measures (e.g. noise level) from point observations. The estimated measurement at each location (target location) is calculated as a weighted average of the observations within a search window (or bandwidth). The weight of each observation is decided by applying a kernel function to the distance between the target location and that observation. The kernel function is typically a distance decay function with a maximum value when the distance is zero and with a zero value when the distance exceeds the bandwidth. The formula of kernel smoothing is shown below, where $K(\ )$ is the kernel function, h is the bandwidth, (X_i, Y_i) is the location of observation i, and Z_i is the environmental measures of observation i.

$$\frac{\sum_{i=1}^{n} K\left(\frac{x-X_i}{h}, \frac{y-Y_i}{h}\right) Z_i}{\sum_{i=1}^{n} K\left(\frac{x-X_i}{h}, \frac{y-Y_i}{h}\right)}$$

Performing kernel smoothing with a massive number of observations from multiple users is extremely computationally intensive. Hence, a parallel kernel smoothing algorithm is implemented based on CUDA[9] (Compute Unified Device Architecture) to exploit the computational power of GPUs. Multiple parallel threads are launched simultaneously, each of which estimates the measurement at one location (one cell for the output raster). Each thread searches through each of the observations, calculates the weight of this observation to its cell, and outputs the weighted average of these observations as an estimated measurement of its cell. In this case, the 50-m kernel bandwidth distance is also incorporated as the bandwidth of the kernel smoothing method, and the output is a GeoTIFF image, which is overlaid on top of ESRI world street map for visualization purposes.

4 User Case Scenario

A noise mapping user case is investigated by collecting data of sound pressure using a mobile application. The application utilizes the microphone of a mobile device to measure sound pressure with the noise level calculated in decibels (dB) using the following equation (Bies and Hansen 2009; Maisonneuve et al. 2009):

[9] http://www.nvidia.com/object/cudahome new.html

$$L_p = 10\log_{10}\left(\frac{{p_{rms}}^2}{{p_{ref}}^2}\right) = 20\log_{10}\left(\frac{p_{rms}}{p_{ref}}\right) dB$$

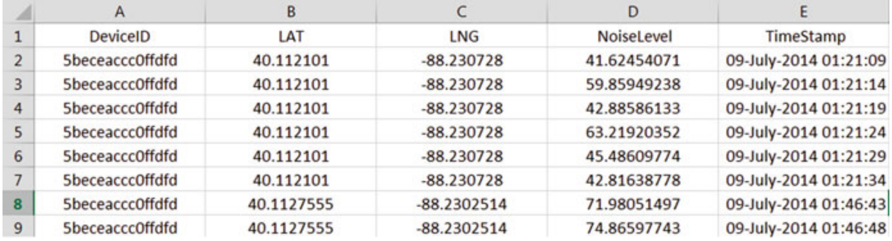
where p_{ref} is the reference sound pressure level with a value of 0.0002 dynes/cm^2 and p_{rms} is the measured sound pressure level. According to the World Health Organization Night Noise Guidelines (NNGL) for Europe,[10] the annual average noise level of 40 dB is considered as equivalent to the lowest observed adverse effect level (LOAEL) for night noise, whereas a noise level above 55 dB can become a major public health concern and over 70 dB can cause severe health problems. This calculated value is also calibrated by users according to physical environment conditions and the type of mobile device.

The mobile application assigns a pair of geographic coordinates (in the format of latitude and longitude) to each measured value together with a timestamp. The update time interval for each recording is set to every 5 s. The recorded measurements are saved directly on the mobile device and we let users decide when to upload their data to the server, whether immediately after taking the measurements or at a later time. An example of the data format of the measurements is shown in Fig. 3. Note that the measurements of other sensors on a mobile device can be included. Given the diversity of sensors on different devices, we use a flexible data management approach based on MongoDB.

In this user case scenario, we choose the campus of University of Illinois at Urbana—Champaign and its surroundings as the study area and asked the participants to go around the campus to collect the noise level measurements. The user interface of the mobile application is shown in Fig. 4, where users have the options to record, upload and interact with noise maps. The mobile application is implemented as a background service on the device and therefore participants are free to engage in other activities.

From a generated noise map, we can identify those spots at which the noise level exceeds such ranges. In Fig. 5, we can examine the visualization of the noise exposure to an individual participant along their trajectory. At the current stage, we have not quantitatively estimated accumulated noise exposure, which will be taken into account in our future work. Figure 6 shows the noise map of a specified hour using a 50-m kernel bandwidth, which is generated from the measurements uploaded by all of the participants during this period. From the visualized results, we can identify the spots where the noise pollution occurs (shown in red) within the specified hour. A new feature to be evaluated for providing in-depth information about what causes such noise pollution is to allow users to append descriptive text when they carry out monitoring using their mobile devices (Maisonneuve et al. 2009). Figure 7 is the noise map of the same hour but using 100-m kernel bandwidth, which demonstrates the effects of choosing different sound decay distance since the value can be changed in framework.

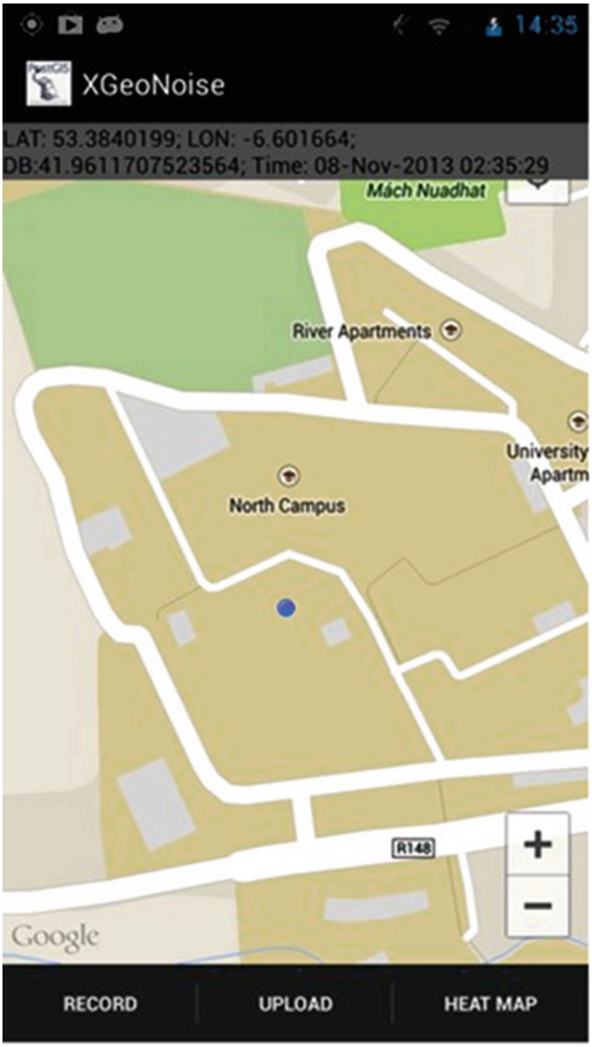
[10] http://www.euro.who.int/data/assets/pdf file/0017/43316/E92845.pdf



	A	B	C	D	E
1	DeviceID	LAT	LNG	NoiseLevel	TimeStamp
2	5beceaccc0ffdfd	40.112101	-88.230728	41.62454071	09-July-2014 01:21:09
3	5beceaccc0ffdfd	40.112101	-88.230728	59.85949238	09-July-2014 01:21:14
4	5beceaccc0ffdfd	40.112101	-88.230728	42.88586133	09-July-2014 01:21:19
5	5beceaccc0ffdfd	40.112101	-88.230728	63.21920352	09-July-2014 01:21:24
6	5beceaccc0ffdfd	40.112101	-88.230728	45.48609774	09-July-2014 01:21:29
7	5beceaccc0ffdfd	40.112101	-88.230728	42.81638778	09-July-2014 01:21:34
8	5beceaccc0ffdfd	40.1127555	-88.2302514	71.98051497	09-July-2014 01:46:43
9	5beceaccc0ffdfd	40.1127555	-88.2302514	74.86597743	09-July-2014 01:46:48

Fig. 3 An example of recorded noise measurements saved on a mobile device

Fig. 4 The user interface of the mobile application



LAT: 53.3840199; LON: -6.601664; DB:41.9611707523564; Time: 08-Nov-2013 02:35:29

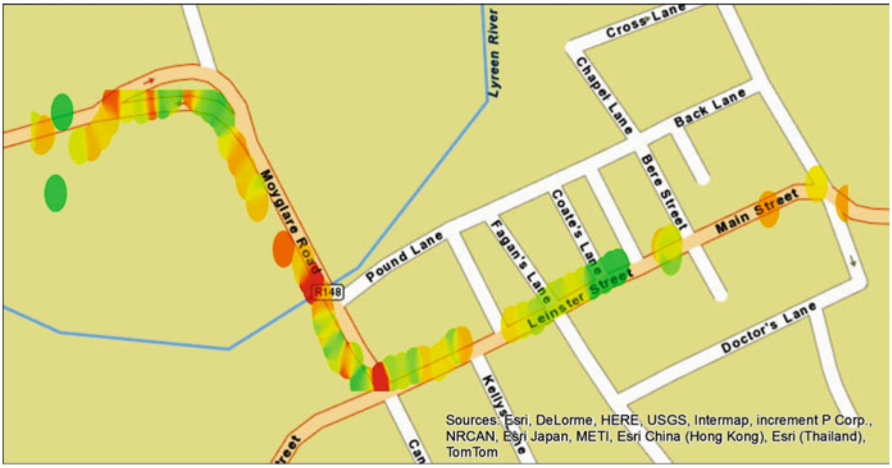


Fig. 5 Noise mapping along the trajectory of an individual participant

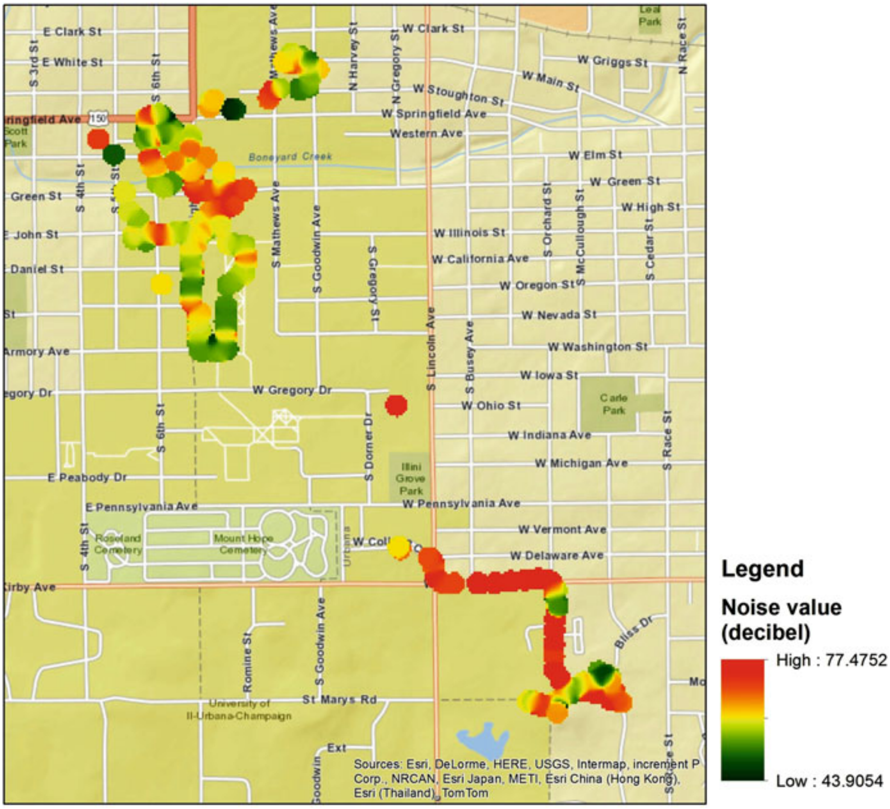
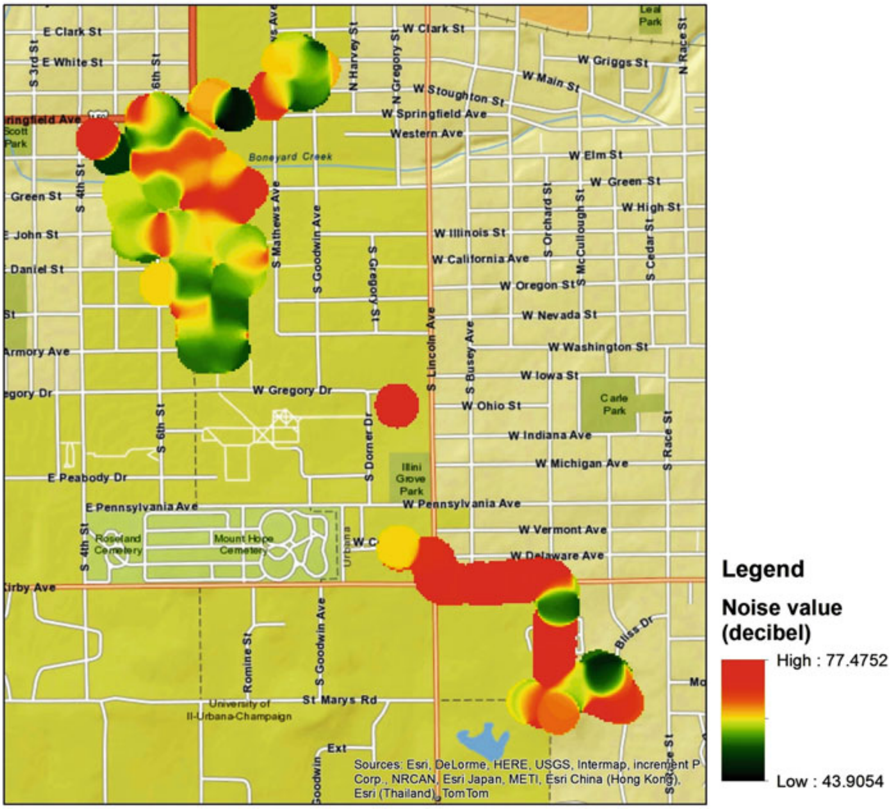


Fig. 6 The generated noise map using a 100-m kernel bandwidth during a specified hour



Sources: Esri, DeLorme, HERE, USGS, Intermap, increment P Corp., NRCAN, Esri Japan, METI, Esri China (Hong Kong), Esri (Thailand), TomTom

Fig. 7 The generated noise map using a 100-m kernel bandwidth during a specific hour

5 Conclusions and Future Work

The availability of a variety of affordable mobile sensors is fostering volunteered participation of citizens in sensing urban environments using mobile devices. By utilizing embedded GPS receivers to append geographic coordinates to sensor measurements, the collective efforts from participatory urban sensing activities can provide high-resolution spatiotemporal data for creating pollution maps of large cities. In relation to the big data collected from such crowdsourcing activities, CyberGIS provides high performance computing and participatory computing architecture to support scalable user participation and data-intensive processing, analysis and visualization.

In this paper, we present a framework that utilizes several components of the CyberGIS platform to facilitate citizens in engaging with environmental monitoring using mobile devices. This framework is intended to incorporate readings from the environmental sensors on the mobile device. As the availability of sensors varies on different devices, this framework chooses a MongoDB (without the requirement for a predefined schema) cluster for data storage. A MapReduce approach is used to filter and extract trajectories of each individual participant to visualize the pollution exposure. It is also used for dynamically generating pollution maps by aggregating the collected sensor measurements using a time window. The pollution maps are rapidly generated by using kernel method via paralleled GPU. In this study, we only demonstrate the functionality of the framework using the case for dealing with the geo-tagged and timestamped noise level measurements, which is collected from our dedicated prototype mobile application using the combination of an integrated GPS receiver and a microphone on a mobile device.

At the current stage, there are still some limitations regarding the implementation of the framework. For example, the selection of the kernel method assumes the measured values stay the same within the kernel bandwidth, which may not be the case in real-world scenarios. Also, the kernel method may not be suitable for generating other pollution maps, for example, air pressure. Therefore, some domain knowledge is required for future improvement of the framework. In relation to trajectory extraction for visualizing pollution exposure to individual participants, quantitative methods for estimating actual exposure need to be explored. Furthermore, we plan to acquire environmental measurements from pertinent government agencies to validate the results that are produced based on data from volunteered participants. Finally, the current MapReduce method relies on the MongoDB cluster, where Apache Hadoop is being explored to improve computational performance.

Acknowledgements This material is based in part upon work supported by the U.S. National Science Foundation under grant numbers: 0846655, 1047916, and 1354329. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors are grateful for insightful comments on the earlier drafts received from Yan Liu, Anand Padmanabhan.

References

Bies DA, Hansen CH (2009) Engineering noise control: theory and practice. CRC press, Boca Raton, FL

Bryant RE (2009) Data-intensive scalable computing harnessing the power of cloud computing (Tech. Rep.). CMU technical report. Retrieved from http://www.cs.cmu.edu/bryant/pubdir/disc-overview09.pdf

Dean J, Ghemawat S (2008) MapReduce: simplified data processing on large clusters. Commun ACM 51(1):107–113

Directive E (2002) Directive 2002/49/ec of the European parliament and the council of 25 June 2002 relating to the assessment and management of environmental noise. Off J Eur Communities 189(12):12–26

Ghosh S, Raju P.P, Saibaba J, Varadan G (2012) Cybergis and crowdsourcing–a new approach in e-governance. In Geospatial Communication Network. Retrieved from http://www.geospatialworld.net/article/cybergis-and-crowdsourcing-a-new-approach-in-e-governance/

Goodchild MF (2007) Citizens as sensors: the world of volunteered geography. GeoJournal 69 (4):211–221

Han J, Haihong E, Le G, Du J (2011) Survey on NoSQL database. In Proceeding of 6th international conference on pervasive computing and applications (ICPCA), pp 363–366

Howe J (2006) Crowdsourcing: a definition. In: Crowdsourcing: tracking the rise of the amateur

Maisonneuve N, Stevens M, Niessen ME, Steels L (2009) Noisetube: measuring and mapping noise pollution with mobile phones. In: Information technologies in environmental engineering. Springer, New York, pp 215–228

Maisonneuve N, Stevens M, Ochab B (2010) Participatory noise pollution monitoring using mobile phones. Inform Polity 15(1):51–71

Stevens M, DHondt E (2010) Crowdsourcing of pollution data using smartphones. In: Workshop on ubiquitous crowdsourcing, Ubicomp'10, September 26–29, 2010, Copenhagen, Denmark

Stonebraker M, Madden S, Abadi DJ, Harizopoulos S, Hachem N, Helland P (2007) The end of an architectural era: (it's time for a complete rewrite). In Proceedings of the 33rd international conference on very large data bases, pp 1150–1160

Wang S (2010) A cybergis framework for the synthesis of cyberinfrastructure, GIS, and spatial analysis. Ann Assoc Am Geogr 100(3):535–557

Wang S, Wilkins-Diehr NR, Nyerges TL (2012) Cybergis-toward synergistic advancement of cyberinfrastructure and giscience: a workshop summary. J Spat Inform Sci 4:125–148

Wang S, Anselin L, Bhaduri B, Crosby C, Goodchild MF, Liu Y, Nyerges TL (2013a) Cybergis software: a synthetic review and integration roadmap. Int J Geogr Inf Sci 27(11):2122–2145

Wang S, Cao G, Zhang Z, Zhao Y, Padmanabhan A, Wu K (2013b) A CyberGIS environment for analysis of location-based social media data. In: Hassan AK, Amin H (eds) Location-based computing and services, 2nd edn. CRC Press, Boca Raton, FL