Mosaic: A Low-Cost Mobile Sensing System for Urban Air Quality Monitoring

Yi Gao[1,2], Wei Dong[1*], Kai Guo[1], Xue Liu[2], Yuan Chen[1] Xiaojin Liu[1], Jiajun Bu[1], Chun Chen[1]
[1]College of Computer Science, Zhejiang University, China.
[2]School of Computer Science, McGill University, Canada.
Email: {*gaoy, dongw, guok, chenyuan, liuxj*}*@emnets.org*, *xueliu@cs.mcgill.ca*, {*bjj, chenc*}*@zju.edu.cn*

***Abstract*—Air quality monitoring has attracted a lot of attention from governments, academia and industry, especially for $PM_{2.5}$ due to its significant impact on our respiratory systems. In this paper, we present the design, implementation, and evaluation of Mosaic, a low cost urban $PM_{2.5}$ monitoring system based on mobile sensing. In Mosaic, a small number of air quality monitoring nodes are deployed on city buses to measure air quality. Current low-cost particle sensors based on light-scattering, however, are vulnerable to airflow disturbance on moving vehicles. In order to address this problem, we build our air quality monitoring nodes, Mosaic-Nodes, with a novel constructive airflow-disturbance design based on a carefully tuned airflow structure and a GPS-assisted filtering method. Further, the buses used for system deployment are selected by a novel algorithm which achieves both high coverage and low computation overhead. The collected sensor data is also used to calculate the $PM_{2.5}$ of locations without direct measurements by an existing inference model. We apply the Mosaic system in a testing urban area which includes more than 70 point-of-interests. Results show that the Mosaic system can accurately obtain the urban air quality with high coverage and low cost.**

I. Introduction

As reported by the World Health Organization (WHO), about 7 million premature deaths in 2012 are linked to air pollution, and most of the cases are within low- and middle-income countries [1]. Due to the significant impact on our respiratory systems [2] and even blood systems [3], $PM_{2.5}$ (particulate matter with a diameter of 2.5 micrometers or less) has attracted a lot of attention recently, especially in developing countries with severe air pollution. According to Daxue Consulting, total sales of air purifiers in China hit 3.5 billion RMB (559 million USD) in 2013, which is 80-100% growth year-on-year compared to 2012 [4].

As a result, many countries start to monitor the air quality and publish the data, to support effective air pollution control as well as to raise awareness of the citizens. There have been different approaches for air quality monitoring, e.g., remote sensing [5] and static air quality measurement stations. However, such approaches usually require a large amount of money and human resources. For example, a typical air quality measurement station needs about 200,000 USD for construction and 30,000 USD per year for maintenance [6].

Besides the above two government-led approaches, there are also projects with much smaller investments. AirCloud [7] is a novel client-cloud system for pervasive and personal air quality monitoring. The collected data is used to infer the whole air quality map by an analytical engine in a cloud-based back-end. Similar to using high-end measurement stations, however, using low-cost stationary sensors (or human-carried sensors) also suffers the lack of coverage and scalability problem. In order to achieve high coverage and accuracy, more sensors will be deployed in the AirCloud project (1000 AirCloud sensors have been built to monitor a city [7]), increasing the total cost significantly. Therefore, new designs are required to achieve a higher coverage while keeping the low cost feature unaffected.

In this paper, we present Mosaic, a low-cost mobile sensing system for urban air quality monitoring. In order to achieve low-cost air quality monitoring, the basic idea is to deploy air quality sensors to moving city buses to increase the system coverage. However, to the best of our knowledge, current low-cost $PM_{2.5}$ sensors based on light-scattering cannot be directly used on moving vehicles due to the severe airflow disturbance (more details in Section IV). In order to address this problem, we first developed a kind of low-cost air quality sensing nodes, *Mosaic-Nodes*, which are able to achieve good accuracies on moving vehicles. Then we attached the Mosaic-Nodes to buses carefully selected to obtain larger coverage per node. By this mobile sensing design, it becomes possible to cover a large urban area with only a small number of sensing nodes, lowering the total cost significantly. Finally, based on the measured air quality data and an existing inference model [7], we are able to obtain a high resolution air quality map of the monitored area.

Figure 1 shows the Mosaic-Node we developed with an external wind sensor for in-lab emulation and a `Dylos` [10] particle sensor for validation. There are multiple sensors in a Mosaic-Node, including a GPS, a $PM_{2.5}$ sensor, a noise sensor, and a humidity & temperature sensor. In order to lower the cost, we choose a $PM_{2.5}$ sensor based on light-

[*]This work is supported by the National Science Foundation of China (No. 61502417 and No. 61472360), Zhejiang Commonwealth Project (2015C33077), Zhejiang Provincial Natural Science Foundation of China (No. LY16F020006), Fundamental Research Funds for the Central Universities, China Ministry of Education−China Mobile Joint Project (No. MCM20150401), and Zhejiang Provincial Platform of IoT Technology (No. 2013E60005). Wei Dong is the corresponding author.

TABLE I
TYPICAL AIR QUALITY MONITORING APPROACHES

Feature/Approach	Satellites [5]	Stations	AirCloud [7]	UFP [8]	GasMobile [9]	Mosaic
Cost	high	high	low	medium	low	low
Accuracy	low	high	medium	medium	medium	medium
Coverage	very large	small	small	large	medium	large
Type	multiple	all	$PM_{2.5}$	UFP	ozone	$PM_{2.5}$

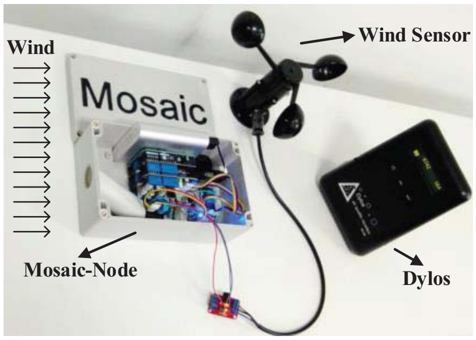


Fig. 1. A Mosaic-Node with an external wind sensor for in-lab emulation, and a Dylos particle sensor for validation.

scattering method. However, when a sensor node based on light-scattering is moving along with the bus, the relative airflow will introduce significant measurement errors. We propose a novel Constructive Airflow-Disturbance method to address this problem. By using this method, the fast relative airflow can be used to carry particles and reduce the unstable measurements, instead of introducing errors. The airflow structure inside the node are carefully designed and tuned by both in-lab and field experiments.

The Mosaic system uses a POI-oriented (point-of-interest-oriented) Bus Selection algorithm to select buses to deploy Mosaic-Nodes. A POI is a location which someone may find useful or interesting, e.g., a school or a hospital. Intuitively, we want to deploy Mosaic-Nodes near POIs to improve the measurement accuracy. Based on this intuition, we define a POI-oriented coverage for each location in the monitoring area, and design an efficient algorithm to select buses which maximize the total coverage. We apply the proposed algorithm to a dataset including 1415 buses with 282 different routes. Compared with a random-walk algorithm and an evolutionary algorithm, the proposed algorithm selects buses with much larger total coverage efficiently. We summarize contributions of this paper as follows.

- We design, implement, and evaluate a low-cost air quality sensing node Mosaic-Node, with carefully tuned airflow structure.
- We propose a novel Constructive Airflow-Disturbance method to utilize the relative airflow to improve the measurement accuracy, by using GPS data to filter out some unstable raw data. To the best of our knowledge, Mosaic-Node is the first low-cost $PM_{2.5}$ monitoring node designed for working on moving vehicles, which is essential to enable high coverage at low cost.
- We propose an efficient POI-oriented Bus Selection algorithm to select buses for sensor deployment. Compared with a random algorithm and an existing approach, the proposed algorithm achieves much higher total coverage with a low computation overhead.
- We implement the Mosaic system and deploy the Mosaic-Nodes to city buses for urban air quality measurement. Results show that the Mosaic system is able to monitor urban air quality at a much lower cost with good accuracies.

The rest of this paper is structured as follows. Section II describes the related work about air quality monitoring. Section III presents the overview of our approach. Section IV gives the constructive airflow-disturbance in detail. Section V presents the POI-oriented bus selection algorithm. Section VI shows the evaluation results, and finally, Section VII concludes this paper and gives directions of future work.

II. RELATED WORK

A. Air Quality Monitoring

Air pollution is one of the most important environmental problems and has attracted great attention in recent years. Table I shows some typical works focusing on air quality monitoring in recent years.

Remote sensing using satellites [5] can help us obtain a coarse-grained information about the surface air quality. It can easily cover a large-scale area by using only one satellite. However, the cost of this method is very high, and the accuracy of remote sensing highly depends on factors like weather and land-use characteristics. Using stationary air quality measurement stations is the approach that most of the countries currently take to obtain more reliable air quality data. Such a station can provide very accurate air quality measurements at the deployed location. However, these stations usually requires a large amount of money and human resources to build and maintain (about 200,000 USD for construction and 30,000 USD per year for maintenance [6]).

AirCloud [7] uses much affordable sensors to obtain air quality data, $PM_{2.5}$ concentration in particular. In order to improve the accuracy, it uses a novel air quality analytic engine to calibrate the sensed data at a cloud-based back-end. The AirCloud project validates that low-cost sensors based on light-scattering can also be used for public air quality monitoring. In order to achieve high coverage to improve accuracy, a large number of stationary sensing nodes should be deployed, increasing the total cost.

Besides $PM_{2.5}$, there are also works focusing on other atmospheric constituents, such as ultrafine particles [8] (UFP) and ozone [9]. In order to monitor ultrafine particles, a mobile sensing approach [8] is proposed. Measuring ultrafine particles is much more expensive (e.g., 10k USD per sensor) than measuring larger particles such as $PM_{2.5}$ (e.g., <100 USD per sensor). Since it uses mobile sensing to monitor air quality, its coverage is much larger than the stationary station approach. There are also participatory sensing approaches [9], [11] to measuring gasses by mobile phones. GasMobile [9] is a typical such approach to measuring ozone based on mobile phones. Since monitoring air pollutants like $PM_{2.5}$ is not as easily portable as gasses, it is still challenging to monitor $PM_{2.5}$ based on mobile phones.

In this work, we use customized sensors based on light-scattering method, to achieve the low-cost goal and address the relative airflow disturbance problem on moving vehicles. Further, by deploying the Mosaic-Node on buses carefully selected, the coverage of our Mosaic system is significantly improved.

B. Air Quality Inference

Besides monitoring the air quality directly, there are also approaches based on modeling and inference. Classical dispersion models [12] infer the air quality of a certain location without direct measurement as a function of traffic volumes, emission factors, meteorology, and etc. LUR (land-use regression) [8], [13] uses the land-use and traffic characteristics to model pollution concentrations. Combining satellite data with meteorological features and land-use information, Liu et al. [14] calculate daily $PM_{2.5}$ maps with a 4km spatial resolution. For ultrafine particles, Clifford et al. [15] develop models using land-use information and meteorological features. These air quality inference models could be more accurate when there is more direct air quality measurements. Since we mainly focus on how to get more accurate direct measurements at low cost in this paper, we use an existing inference model [7] to calculate the air quality at locations without any direct measurements. Using different models with different data sources could be beneficial in terms of accuracy or efficiency, which is considered as future work.

III. Mosaic Overview

A. Sensing Nodes

We use two kinds of sensing nodes in the Mosaic system, `Dylos` and Mosaic-Nodes. `Dylos` [10] is a portable air quality sensor costing around 400 USD. It is able to measure $PM_{2.5}$ concentrate accurately by a laser counter. In the Mosaic system, several `Dylos` monitors are used to provide accurate air quality measurements to calibrate the low-cost Mosaic-Nodes. Note that `Dylos` nodes cannot replace Mosaic-Nodes in the Mosaic system due to the following reasons. First, `Dylos` nodes have a much higher energy consumption, making it dependent to external power source for long-term sensing. Second, more importantly, `Dylos` nodes are not designed to work outdoor with different weather, e.g., raining.

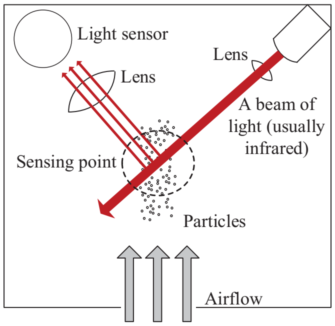


Fig. 2. The light-scattering method to sense the particle concentration.

A Mosaic-Node includes multiple sensors: a GPS, a customized $PM_{2.5}$ sensor, a noise sensor, and a humidity & temperature sensor. In this paper, we only use the $PM_{2.5}$ data, the GPS data, and the humidity & temperature data. The GPS used in Mosaic-Node is a `NEO-6 GPS` [16]. The customized $PM_{2.5}$ sensor is a modified `SHINYEI PPD42NS` [17] sensor. We modified the sensor to improve its accuracy on moving vehicles. Details are given in Section IV. A Mosaic-Node also includes other components, such as computation/storage/communication components, a portable power pack, plastic encapsulation, and PVC pipe-based airflow structures. In total, a Mosaic-Node costs 90 USD at a small production. It is expected that the cost per node can be further lowered when we want to manufacture a large number of nodes for large-scale deployment.

B. Design overview

When a light-scattering based air quality sensor is moving with a bus, the relative airflow will introduce significant errors. By using a Constructive Airflow-Disturbance method (including a customized airflow structure and a GPS-assisted filtering method), we are able to improve the measurement accuracy of Mosaic-Nodes on moving vehicles significantly (details in Section IV). In order to select buses to better cover the targeted area, we use a POI-oriented (POI: point of interest) bus selection method to select a number of buses to deploy Mosaic-Nodes (details in Section V). A POI is a specific location that someone may find useful or interesting. Based on the two key components of the Mosaic system, Constructive Airflow-Disturbance and POI-oriented Bus Selection, eight Mosaic-Nodes are deployed to eight buses in Hangzhou, China. Then we use an existing inference model to calculate the air quality of locations without direct measurement, and obtain the $PM_{2.5}$ map with high resolution.

IV. Constructive Airflow-Disturbance

A. Light-scattering Particle Sensors

We first give a brief description about the light-scattering method used by low-cost air quality sensors. The basic idea is

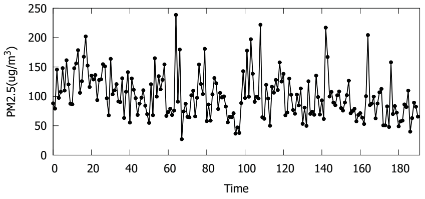


Fig. 4. Raw data from an original `PPD42NS`.

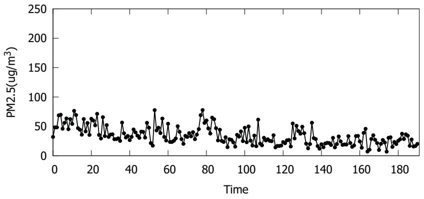


Fig. 5. Raw data from a customized `PPD42NS`.

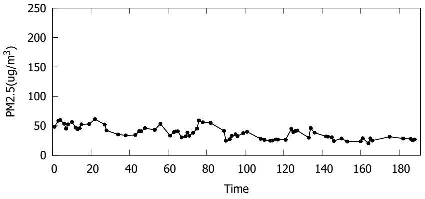


Fig. 6. After using GPS filtering.

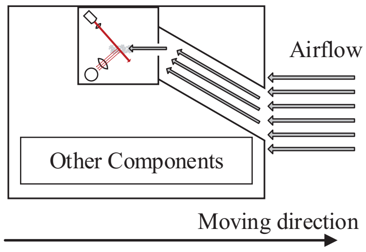


Fig. 3. Use relative airflow to carry particles.

to use a light sensor to count the amount of particles per unit volume. Figure 2 shows how it works. A beam of light from a LED is focused with lens to the sensing point (or viewing volume). Then the particles passing through the sensing point scatter the light. A light sensor receives the scattered light through the lens and generates pulse signal. Finally, the particle concentration can be obtained after conversion.

Intuitively, when there are more particles at the sensing point, the light sensor can receive more scattered light. One important issue of this light-scattering method is how to control the airflow carrying the particles, so that the particle amount can be measured by the readings of the light sensor. In the original `PPD42NS` sensor, a 100Ω resistor is used to generate heat to enable a slowly rising airflow passing through the sensing point. When the sensor is working in a static environment without severe airflow disturbance, the speed of the rising airflow is stable, supporting stable particle concentration measurement. However, when such a sensor is deployed on a moving vehicle, there will be severe airflow disturbance, decreasing the measurement accuracy significantly.

B. Customized Particle Sensor

In order to address the airflow-disturbance problem, we modified the airflow structure of the original `PPD42NS` sensor. As mentioned above, the original `PPD42NS` sensor includes a resistor to generate heat to enable a slowly rising airflow. We removed the resistor due to the following two reasons. First, the resistor costs a large amount of energy (250 mw, while the whole sensor costs 450 mw). Like many sensor networks, energy efficiency is one of the most important design requirements [18]–[20]. Second, the slowly rising airflow could be severely affected when the sensor is deployed on a moving vehicle, causing it to be useless. Figure 4 shows the recorded particle concentration when we deploy the Mosaic-Node with an original `PPD42NS` sensor to a bus. We can see that the readings are very unstable, due to the severe airflow-disturbance on a moving vehicle.

We found out that the relative airflow introduced by the moving vehicle can be used constructively. The basic idea is to use the relative airflow to carry particles through the sensing point. In order to achieve this, we carefully designed the airflow structure inside the Mosaic-Node. Figure 3 shows the basic design of using the relative airflow to carry the particles. By such an airflow design, the relative air flow can be used constructively, and the light from outside the Mosaic-Node does not cause interference to the light sensor.

We tested the customized sensor on a bus and obtained the particle concentration data shown in Figure 5. We can see that the data is much more stable compared with that from the original sensor. However, there are still many noisy readings. After many rounds of experiments, we have the following two observations. First, when the vehicle is moving slowly, the relatively airflow can be easily interfered, causing unstable readings. Second, when the vehicle is accelerating or decelerating quickly, the readings become very unstable. Note that the relative airflow speed of a Mosaic-Node on a moving vehicle depends on the vehicle speed and the wind speed. Since the wind speed is usually much smaller than the vehicle speed and slowed down by the buildings in an urban area [21], we do not include the wind speed into the sensing data calibration of the Mosaic system. Therefore, we use the speed and the acceleration of a vehicle, which are calculated by the GPS, to filter out some noisy readings.

Figure 6 shows the readings after filtering. We can see that the readings are much more stable and the sensing accuracy can be improved significantly. We also calculate the variances of the readings before and after filtering. The variance before filtering is 246.4 and the variance after filtering is 115.8, validating the observation. By using this GPS-assisted filtering method, the raw data obtained by the customized sensor is more reliable to be further analyzed.

The Constructive Airflow-Disturbance method proposed in this paper mainly includes a customized particle sensor design and a GPS-assisted filtering method, as described in the previous subsection. The sensing data after the filtering is further calibrated by the help of meteorological data and a more accurate particle sensor. We summarize the whole data calibration process as follows. The raw data is filtered by the speed and acceleration obtained by the GPS. After the filtering, Mosaic employs an existing quadratic smoothing method and a three-layer ANN model [7] (i.e., Artificial Neural Network model) to further calibrate the filtered data. We use the meteorology factors (e.g., temperature, humidity) and the air quality data from the `Dylos` sensor to train the ANN model. After model training, we use the trained ANN model to calculate the final calibrated air quality data.

V. POI-oriented Bus Selection

We use the location data of POI (point-of-interest, e.g., schools and hospitals) and the bus GPS data to select a number of buses for deploying the Mosaic-Nodes.

A. Problem Formulation

We first give the problem formulation in this subsection. The input of the bus selection problem includes the bus GPS trace, the locations of the POIs, and the number of buses that we want to select. The output is a number of buses which can maximize the sensing coverage.

There are different ways [22]–[24] to define the sensing coverage. However, for an urban air quality monitoring system, it is usually not possible to achieve full coverage without using a large number of sensors. In practice, inference models are used to infer the air quality at locations without direct measurements. The locations of those direct measurements play a key role in the air quality inference. Further, we may be more interested in some locations (i.e., POIs), such as schools and hospitals, than other locations. Therefore, in Mosaic, we define the coverage by considering air quality inference and the locations of POIs. If we have a sufficient number of sensors and there are always at least one bus passing through each POI, we can easily solve the maximum coverage bus selection problem. In practice, however, we want to monitor the air quality in a low-cost manner, which means that we can only use a small number of sensors. Further, there is not always a bus passing through each POI. Therefore, the air quality of some POIs should be calculated indirectly. In order to improve the accuracy, we want the measured locations to be close to POIs. Concretely, we define the *coverage* of a certain location l as a function of its distance to possible direct measurements and POIs. When there are more adjacent direct measurement and more adjacent POIs, the coverage of that location will be larger. Let B be the set of all buses, $S \subset B$ be the set of selected buses, and P be the set of all POIs. Given n sensors to deploy, we want to obtain an S which can maximize the total coverage on the monitored area L. The bus selection problem is shown as follows.

$$\begin{aligned} \underset{S}{\text{maximize}} \quad & \sum_{l \in L} c(l, S, P) \\ \text{subject to} \quad & |S| = n, \ S \subset B. \end{aligned} \tag{1}$$

In the above problem formulation, the location l is defined discretely, in order to improve the calculation efficiency of the bus selection algorithm. This problem is essentially a weighted maximize cover problem [25], which has been proved to be NP-hard. A greedy algorithm that selects a bus with the largest coverage gain is able to achieve an approximation ratio of $1 - 1/e$ [25]. In the air quality monitoring scenario, however, direct measurements from directly covered locations are further used to infer the air quality of other locations without direct coverage. It causes the coverage of each location to be correlated to the selected buses, introducing extra difficulties for solving the problem. Therefore, we propose a customized greedy algorithm to solve the problem efficiently. In the following, we first give a basic solution of selecting bus routes, instead of selecting buses. Then we introduce the actual bus selection algorithm used in Mosaic by adding more details.

B. Route Selection: A Basic Solution

We discretize the monitored area into a set L of 100m×100m blocks $\{l_1, l_2, ...l_{|L|}\}$. Then we discretize the coverage calculation by two steps, considering the POIs and the buses in each step.

First, given the locations of the POIs, we define the *importance* $r(l, P)$ of each location l as a monotonic function with respect to the distance from a location to the nearest POI, which is formally given as follows.

$$\begin{aligned} r(l, P) = \delta;\ &\text{if } \exists p \in P,\ p \text{ and } l \text{ are in the same block;} \\ r(l_1, P) > r(l_2, P);\ &\text{if } \exists p \in P, d(p, l_1) < d(p', l_2)\ (\forall p' \in P), \end{aligned} \tag{2}$$

where $d(p, l)$ is the distance from a POI p to a location l and δ is a constant representing the maximum importance. Since the location importance can be calculated separately before solving the bus selection problem, the complexity of the monotonic function does not affect the calculation efficiency of bus selection. In Mosaic, we use a quadratically decreasing function to calculate the importance of different locations.

Second, we discretize the coverage calculation considering the buses selected. Here, we view different buses of the same bus route as one bus. Then the bus selection problem actually becomes a *route selection problem*. Let R be the set of bus routes. We will revisit the actual bus selection problem after giving the route selection algorithm. According to the number of adjacent bus routes passing through a certain block l, we define the coverage of l as follows.

$$c(l, S, P) = \begin{cases} r(l, P); & > 2 \text{ routes passing through } l \\ 0.75 \cdot r(l, P); & 1 \text{ or } 2 \text{ routes passing through } l \\ 0.50 \cdot r(l, P); & \geq 1 \text{ routes passing through } l' \\ 0.25 \cdot r(l, P); & \geq 1 \text{ routes passing through } l'' \\ 0; & \text{otherwise,} \end{cases} \tag{3}$$

where l' is an adjacent block of l and l'' is an adjacent block of a certain l'. If a block satisfies more than one conditions in Equation 3, its coverage is the largest one.

Figure 7 shows an example of the coverage calculation. The left subfigure shows the importance $r(l, P)$ of each block l,

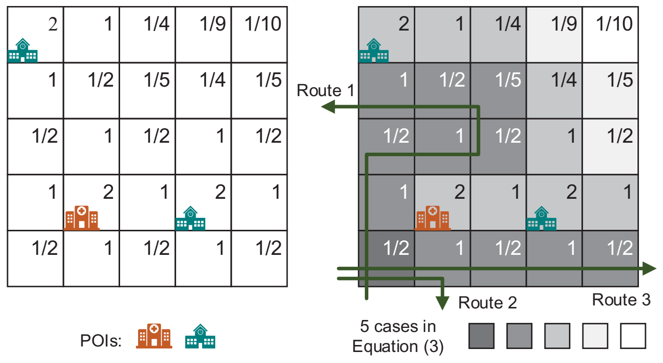


Fig. 7. An example showing the coverage calculation.

Algorithm 1 Route Selection

```
    Input: $P$: a set of POIs; $R$: a set of bus routes;
        $n$: the number of routes, $n \leq |R|$; $L$: a set of blocks
Output: $S$: a set of selected routes in $R$ to deploy sensors
 1: procedure ROUTE-SELECTION
 2:     $totalCoverage = 0$
 3:     $maxCoverage = 0$
 4:     $S = \emptyset$
 5:     for $i = 1$ to $|L|$ do
 6:         $c(l_i, S, P) = 0$
 7:     while $|S| < n$ do
 8:         for each $r \in R$ do
 9:             $tmpS = S \cup \{r\}$
10:             update all $c(l_i, tmpS, P)$
11:             $totalCoverage = \sum_{1 \leq i \leq |L|} c(l_i, tmpS, P)$
12:             if $maxCoverage < totalCoverage$ then
13:                 $maxCoverage = totalCoverage$
14:                 $maxGainRoute = r$
15:         $S = S \cup \{maxGainRoute\}$
16:         $R = R \setminus \{maxGainRoute\}$
17:     return $S$
```

given three POIs (i.e., two schools and one hospital). When there is a POI in a certain block l, its importance is $\delta = 2$. For the blocks without the POI, its importance decrease quadratically with respect to the distance to the nearest POI. The right subfigure shows the five cases in Equation 3, given the routes of three buses. For example, the block with the hospital is the third case, in which two buses passing through one of its adjacent blocks.

Since there are usually a large number of buses passing through the monitored area, performing exhaustive search is not feasible. In [26], an evolutionary algorithm is used to solve a similar problem. In the scenario of Mosaic, however, the evolutionary algorithm cannot select buses with sufficiently large total coverage efficiently. Instead, we employ a greedy algorithm which chooses an additional bus route with the largest new total coverage. Formally, Algorithm 1 shows how the bus routes are selected by a greedy algorithm. After initialization (line 2 to 4), the algorithm tries to add one bus route to S in each iteration of the while-loop (line 15), till that n bus routes have been added to S. In each iteration, the algorithm tries to add each unselected bus route to S (line 8), and finds the bus route with the largest new total coverage (line 9 to 14).

C. Bus Selection

The above route selection does not consider the timing information. In this subsection, we improve the basic version of route selection algorithm to a fine-grained bus selection algorithm with timing information. A new parameter T is introduced into the problem formulation, representing the coverage of every T hours. We consider a typical bus schedule which has 16 operating hours a day. Then these 16 hours are divided into $16/T$ monitoring time windows (slots). Given this new timing parameter, we modify the calculation of $c(l, S, P)$ in Equation 3 as follows.

$$c^T(l, S, P) = \begin{cases} r(l, P); & > 16/2T \text{ slots are covered} \\ 0.75 \cdot r(l, P); & 1 \text{ to } 16/2T \text{ slots are covered} \\ 0.50 \cdot r(l, P); & \geq 1 \text{ buses passing through } l' \\ 0.25 \cdot r(l, P); & \geq 1 \text{ buses passing through } l'' \\ 0; & \text{otherwise,} \end{cases} \tag{4}$$

The first two cases are modified. When more than half of the time windows (i.e., $> 16/2T$ slots) are covered, we consider the block as fully covered.

There is one more problem which is the time complexity. The time complexity of the basic bus selection algorithm is $O(n \cdot |R| \cdot |L|)$, where $|R|$ is the number of different bus routes (282 routes in our system) and $|L|$ is the number of blocks. When we view different buses of the same route differently, there could be a large number of different buses (1415 buses in our system). Further, when we want to monitor a large scale area or a higher spatial resolution, $|L|$ could also be a large number, causing high computation overhead of the algorithm. Note that there are a lot of unnecessary computation in Algorithm 1, such as the coverage update (line 10) and the total coverage calculation (line 11). Since we are only interested in the bus with the largest coverage gain at each iteration, the coverage of blocks far from that bus does not need to be updated. Based on this intuition and the timing information mentioned above, Algorithm 2 shows the fine-grained bus selection algorithm with timing information.

There are several key improvements of the new algorithm compared with Algorithm 1. First, timing information is considered into the calculation (calculation of $c^T(.)$, Equation 4), and different buses of the same route are viewed differently (line 8). Second, the new algorithm only calculates the coverage gain of blocks within two blocks away from a bus, reducing unnecessary calculation (line 13 to 15). Third, in order to further improve the calculation efficiency, the new algorithm pre-computes an upper bound of coverage gain for each bus b (line 4, line 23 to 29). Then based on these upper bounds, the algorithm can skip many attempts to add bus to S (line 9, 10). In order to calculate the time complexity of this new algorithm, we first calculate the number of blocks within two blocks from each bus b, which is $O(|L|^{0.5})$.

Algorithm 2 Bus Selection with Timing Information

```
    Input: $P$: a set of POIs; $B$: a set of buses; $n$: the number of buses, $n \leq |B|$; $L$: a set of blocks; $T$: number of hours per time window
Output: $S$: a set of selected buses in $B$ to deploy sensors
 1: procedure BUS-SELECTION
 2:     $totalCoverage = 0$
 3:     $S = \emptyset$
 4:     $coverageB =$ COVERAGE-UPPER-BOUND
 5:     for $i = 1$ to $|L|$ do
 6:         $c^T(l_i, S, P) = 0$
 7:     while $|S| < n$ do
 8:         for each $b \in B$ do
 9:             if $maxCoverageGain \geq coverageB[b]$ then
10:                 continue
11:             $tmpS = S \cup \{b\}$
12:             $coverageGain = 0$
13:             for each $l$ within two blocks from $b$ do
14:                 update $c^T(l, tmpS, P)$
15:                 $coverageGain$ += coverage-gain-of-$l$
16:             if $maxCoverageGain < coverageGain$ then
17:                 $maxCoverageGain = coverageGain$
18:                 $maxGainBus = b$
19:         $S = S \cup \{maxGainBus\}$
20:         $B = B \setminus \{maxGainBus\}$
21:     return $S$
22:
23: procedure COVERAGE-UPPER-BOUND
24:     for each $b \in B$ do
25:         $sumBound = 0$
26:         for each $l$ within two blocks from $b$ do
27:             $sumBound$ += $c^T(l, \{b\}, P)$
28:         $coverageB[b] = sumBound$
29:     return $coverageB$
```

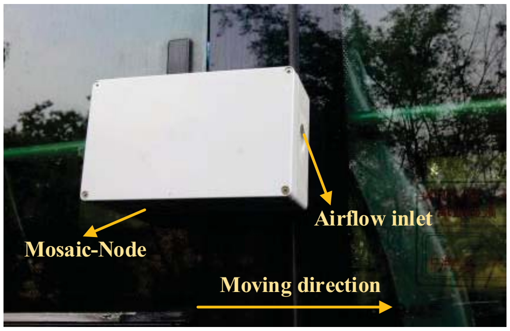


Fig. 8. A Mosaic-Node deployed on a bus. The airflow inlet is heading to the moving direction of the bus.

Therefore, the time complexity of the bus selection algorithm is $O(n \cdot |B| \cdot |L|^{0.5})$. In practice, many attempts are skipped by testing the coverage gain upper bound of a bus (line 9, 10 in Algorithm 2). In our data set with 1415 buses, the bus selection can be completed in 10 seconds, on a desktop computer.

VI. Evaluation

In this section, we present the evaluation methodology and results.

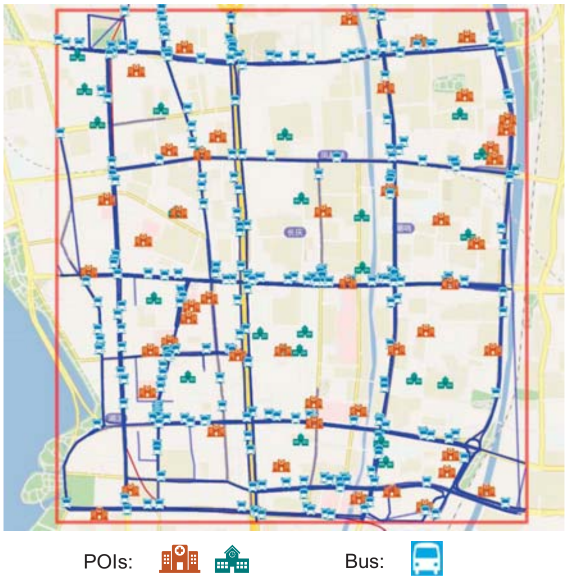


Fig. 9. The testing area, including all POIs and buses.

A. Experiment Setup and Collected Data

The current Mosaic system includes eight Mosaic-Nodes, each is deployed on a city bus selected by the proposed POI-oriented bus selection algorithm. Each Mosaic-Node is deployed outside a bus, and the airflow inlet is heading the moving direction of the bus. Figure 8 shows one Mosaic-Node deployed on a bus. Each Mosaic-Node senses the environment and records the following data, the temperature, humidity, location/time (from a GPS), and the particle concentration (from the customized particle sensor).

The data used in this paper is collected during a deployment over a month, including a training dataset of four weeks and a testing dataset of one week. During the training period, we deploy the Mosaic-Nodes and a `Dylos` node on the same vehicle to collect raw data. We also use the original `PPD42NS` particle sensor to collect data for comparison. During the testing period, only Mosaic-Nodes are deployed on buses, and no `Dylos` node is deployed.

The testing area is a 2.9×3.1 km^2 area located in a middle-scale city with a population of about 9 million, in China. Within the testing area, we collected the following dataset, the 282 bus routes and schedule of all 1415 buses, the locations of 48 hospitals and 24 schools. Figure 9 shows the testing area, including all bus routes passing through it and all POIs (i.e., hospitals and schools). Since the testing area is the downtown of the city, there are a large number of buses and POIs within the area, which makes it be a good testing area to test the scalability of the Mosaic system.

B. Evaluation of Constructive Airflow-Disturbance

We first evaluate the Constructive Airflow-Disturbance method. This method is used to address the problem of severe airflow interference on a moving vehicle. In the Mosaic

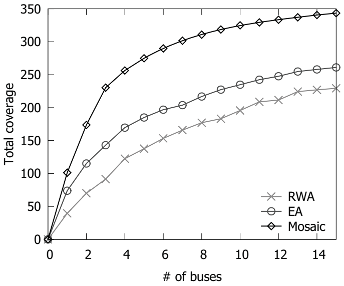


Fig. 11. Total coverage.

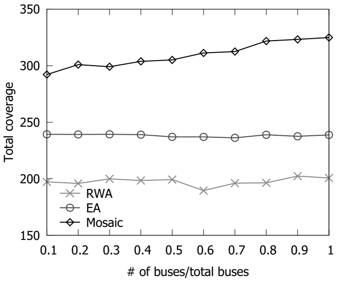


Fig. 12. Total coverage.

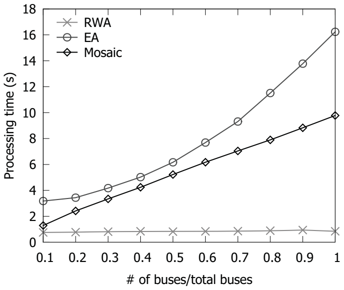
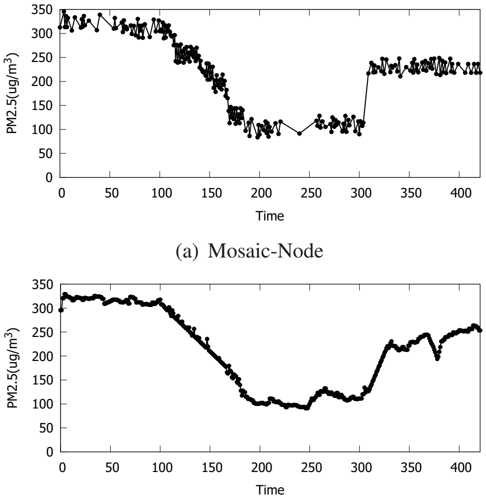


Fig. 13. Processing time.



(a) Mosaic-Node

(b) Dylos

Fig. 10. Air quality data of Mosaic-Node after calibration and Dylos.

system, we develop the Mosaic-Node with a customized particle sensor and an optimized airflow structure, to utilize the relative airflow to carry particles. As mentioned in Section IV, the raw readings of the customized sensor are further filtered by the speed and acceleration of the bus. We put a `Dylos` and a Mosaic-Node on the same bus to collect air quality data. Then we use the proposed method to calibrate the raw data of the Mosaic-Node. The calibrated data is finally compared with the data obtained by the `Dylos`.

Figure 10 shows the final calibrated data after applying the data smoothing and the ANN model, as well as the air quality data obtained by a `Dylos` node. We can see that the calibrated air quality data well matches the data from `Dylos`. Compared with the air quality data obtained by a `Dylos` node, the relative error is only 0.054 on average, enabling practical particle concentration measurement by the Mosaic system.

C. Evaluation of POI-oriented Bus Selection

We then evaluate the POI-oriented bus selection algorithm proposed in this paper. In this subsection, we compare the performances of three bus selection algorithms, the proposed algorithm in this paper ("Mosaic" in the figures), a random walk algorithm ("RWA" in the figures), and an evolutionary algorithm ("EA" in the figures). Since the goal of the bus selection algorithm is to get a higher coverage, we use the total coverage as the main evaluation metric. A higher coverage can enable fine-grained air quality monitoring as well as further air quality related applications, such as pollution source detection.

Figure 11 shows the total coverage calculated by the three algorithms, when different number of buses are selected. We can see that the buses selected by the proposed algorithm always achieve the highest coverage. Also, it is clear that more buses can achieve a higher coverage. In the testing area used in this paper, we deploy eight Mosaic-Nodes to eight buses, since it achieves a good trade-off between cost and coverage.

Figure 12 shows the total coverage calculated by the three algorithms, when different number of *total buses* are considered in the calculation. For example, when $0.4 \times 1415 = 566$ buses are considered in the bus selection, the total coverage obtained by the eight buses selected the three different algorithms are 198.4, 239.0, and 299.1, respectively. From this figure, we can see that when more buses are considered into calculation, only Mosaic can achieve higher total coverage.

We also evaluate the processing speed of the three algorithms. Figure 13 shows the results, when different number of total buses are considered in the calculation. The random walk algorithm achieves the shortest and nearly constant processing time and the processing time of the evolutionary algorithm grows rapidly when more buses should be selected. The processing time of the POI-oriented bus selection algorithm in Mosaic increases linearly when more buses should be selected, making the algorithm be scalable to a larger monitoring area with more buses.

D. Evaluation of Inference Accuracy

The inference accuracy mainly depends on the inference model used and the coverage of direct measurements. In the literature, there are many different inference models which take different information as input. In this work, we use a Gaussian inference model [7], [27] to calculate the air quality of locations without direct measurement. Other inference models with more sophisticated designs and more input data can also be used in the Mosaic system, and we leave this as future work.

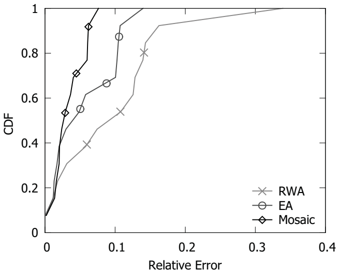


Fig. 14. CDFs of relative error.

Figure 14 shows accuracy of all POIs within the testing area obtained by the three bus selection approaches, i.e., Mosaic, random walk algorithm, and evolutionary algorithm. We use the relative error compared with the ground truth as the metric. On average, the relative error of the Mosaic system is 0.035, the relative error of using random walk algorithm is 0.058, and the relative error of using evolutionary algorithm is 0.103. Since Mosaic uses the same inference model as AirCloud, the accuracy gain comes from two design considerations of the Mosaic system. First, the mobile sensing design of Mosaic enables a large number of direct air quality measurements at different locations, which is essential for an accurate inference. Second, the buses are selected considering the distances to POIs, i.e., buses adjacent POIs are selected. Then the direct measurements near these POIs are further used to improve the inference accuracies of these POIs.

VII. Conclusion

In this paper, we present the design, implementation, and evaluation of Mosaic, a mobile sensing system for low-cost urban air quality monitoring. By using a novel constructive airflow-disturbance design, we successfully realized low-cost $PM_{2.5}$ monitoring on moving vehicles for the first time, achieving a better cost/coverage trade-off than state-of-the-art. We also propose a POI-oriented bus selection algorithm to efficiently select the buses to deploy monitoring nodes, which outperforms two baseline algorithms significantly. We deploy eight Mosaic-Nodes to the selected buses to collect air quality data in a testing area with 72 POIs. Results show that the Mosaic system is able to monitor urban air quality at a much lower cost with good accuracies, enabling possible wide deployments in practice.

As future work, we plan to first extend the current deployment to the rest districts of Hangzhou and to more cities. An initial deployment has been conducted in Ningbo, another middle-scale city in China. Then based on a larger dataset, we hope to develop more air quality related applications, such as pollution source detection, traffic scheduling and travel planning.

References

[1] "WHO News Release," http://www.who.int/mediacentre/news/releases/2014/air-pollution/en/.

[2] E. Boldo, S. Medina, A. Le Tertre, F. Hurley, H.-G. Mücke, F. Ballester, I. Aguilera, and D. Eilstein, "Apheis: Health Impact Assessment of Long-term Exposure to PM2.5 in 23 European Cities," *European Journal of Epidemiology*, vol. 21, no. 6, pp. 449–458, 2006.

[3] M. Sørensen, B. Daneshvar, M. Hansen, L. O. Dragsted, O. Hertel, L. Knudsen, and S. Loft, "Personal PM2. 5 Exposure and Markers of Oxidative Stress in Blood," *Environmental Health Perspectives*, vol. 111, no. 2, p. 161, 2003.

[4] "Air Purifiers Market in China," http://daxueconsulting.com/air-purifiers-market-in-china/.

[5] R. V. Martin, "Satellite Remote Sensing of Surface Air Quality," *Atmospheric Environment*, vol. 42, no. 34, pp. 7823 – 7843, 2008.

[6] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-Air: When Urban Air Quality Inference Meets Big Data," in *Proceedings of ACM KDD*, 2013, pp. 1436–1444.

[7] Y. Cheng, X. Li, Z. Li, S. Jiang, Y. Li, J. Jia, and X. Jiang, "AirCloud: A Cloud-based Air-quality Monitoring System for Everyone," in *Proceedings of ACM SenSys*, 2014, pp. 251–265.

[8] D. Hasenfratz, O. Saukh, C. Walser, C. Hueglin, M. Fierz, and L. Thiele, "Pushing the Spatio-Temporal Resolution Limit of Urban Air Pollution Maps," in *Proceedings of IEEE PerCom*, 2014, pp. 69–77.

[9] D. Hasenfratz, O. Saukh, S. Sturzenegger, and L. Thiele, "Participatory Air Pollution Monitoring Using Smartphones," in *Proceedings of International Workshop on Mobile Sensing*, 2012.

[10] "Dylos," http://www.dylosproducts.com.

[11] Y. Jiang, K. Li, L. Tian, R. Piedrahita, X. Yun, O. Mansata, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang, "MAQS: A Personalized Mobile Sensing System for Indoor Air Quality Monitoring," in *Proceedings of UbiComp*, 2011, pp. 271–280.

[12] S. Vardoulakis, B. E. Fisher, K. Pericleous, and N. Gonzalez-Flesca, "Modelling Air Quality in Street Canyons: A Review," *Atmospheric Environment*, vol. 37, no. 2, pp. 155 – 182, 2003.

[13] G. Hoek, R. Beelen, K. de Hoogh, D. Vienneau, J. Gulliver, P. Fischer, and D. J. Briggs, "A Review of Land-use Regression Models to Assess Spatial Variation of Outdoor Air Pollution," *Atmospheric Environment*, vol. 42, pp. 7561–7578, 2008.

[14] Y. Liu, C. J. Paciorek, and P. Koutrakis, "Estimating Regional Spatial and Temporal Variability of PM2.5 Concentrations Using Satellite Data, Meteorology, and Land Use Information," *Environmental Health Perspectives*, vol. 117, no. 6, pp. 887+, 2009.

[15] S. Clifford, S. Low Choy, T. Hussein, K. Mengersen, and L. Morawska, "Using the Generalised Additive Model to model the particle number count of ultrafine particles," *Atmospheric Environment*, vol. 45, no. 32, pp. 5934–5945, 2011.

[16] "NEO-6 GPS," http://ec-mobile.ru/user_files/File/u-blox/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf.

[17] "SHINYEI PPD42NS," http://www.sca-shinyei.com/pdf/PPD42NS.pdf.

[18] X. Zheng, Z. Cao, J. Wang, Y. He, and Y. Liu, "Zisense: Towards interference resilient duty cycling in wireless sensor networks," in *Proceedings of SenSys*, 2014, pp. 119–133.

[19] Z. Li, M. Li, and Y. Liu, "Towards energy-fairness in asynchronous duty-cycling sensor networks," in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 801–809.

[20] Y. Zhang, S. He, and J. Chen, "Data gathering optimization by dynamic sensing and routing in rechargeable sensor networks," *Networking, IEEE/ACM Transactions on*, vol. PP, no. 99, pp. 1–15, 2015.

[21] R. D. Bornstein and D. S. Johnson, "Urban-Rural Wind Velocity Differences," *Atmospheric Environment (1967)*, vol. 11, no. 7, pp. 597 – 604, 1977.

[22] S. Kumar, T. H. Lai, and J. Balogh, "On K-coverage in a Mostly Sleeping Sensor Network," in *Proceedings of ACM MobiCom*, 2004, pp. 144–158.

[23] S. Kumar, T. H. Lai, and A. Arora, "Barrier Coverage with Wireless Sensors," in *Proceedings of ACM MobiCom*, 2005, pp. 284–298.

[24] L. Kong, M. Zhao, X.-Y. Liu, J. Lu, Y. Liu, M.-Y. Wu, and W. Shu, "Surface coverage in sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 1, pp. 234–243, 2014.

[25] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functionsłI," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[26] O. Saukh, D. Hasenfratz, A. Noori, T. Ulrich, and L. Thiele, "Route Selection for Mobile Sensors with Checkpointing Constraints," in *Proceeding of PERCOM Workshops*, 2012, pp. 266–271.

[27] C. E. Rasmussen, "Gaussian processes for machine learning," 2006.