

Understanding Spatial and Temporal Coverage in Participatory Sensor Networks

Julien Gedeon, Immanuel Schweizer
Technische Universität Darmstadt, Telecooperation Lab
Email: schweizer@cs.tu-darmstadt.de

Abstract—Sensor coverage is a well established problem in sensor networks. Most of the work is focused on optimizing coverage in stationary networks or by controlling the movement of mobile nodes (e.g. robots) in order to maximize their coverage.

In participatory sensor networks, we are faced with non-controllable mobility. Humans move freely and there is no central nor distributed algorithm that optimizes coverage. There is no work in literature yet that explores coverage in the context of non-controllable mobility.

To this end, we report results of a study applying an adapted greedy coverage algorithm onto three different data sets. Given these datasets, we report results studying the effect of different mobility characteristics on the spatial and temporal coverage. Our results show that high coverage can be achieved by a relatively small subset of nodes. Also, given a real-world participatory sensing system, turn-around times are relevant for continuous temporal coverage.

Index Terms—spatial coverage, temporal coverage, participatory sensing, mobility

I. INTRODUCTION

Wireless Sensor Networks are increasingly present in every aspect of today’s digital world. They monitor environmental parameters and critical infrastructures, predict and optimize traffic in growing urbanizations, can provide us with assistance and are used in healthcare applications. Furthermore, they play an important role in military and security applications.

In general, coverage in such sensor networks describes the problem of maximizing spatial coverage given a set of sensors. For participatory sensing, the set of sensors is unpredictable and mobility is non-controllable. Here, coverage has not been studied yet.

Coverage is well understood for wireless sensor networks and was first defined by Gage [1]. Many others [2] have researched coverage in sensor networks. With stationary sensor nodes, the network can be optimized at design time, making this a viable endeavor. Recently, there has been some work on mobile networks. However, these networks are usually based on robots [3], [4]. Again, this enables a certain degree of control on the sensor’s position. Contrary to wireless sensor nodes and robots, the mobility of humans is non-controllable. Nonetheless their locations exhibit a certain amount of predictability. In networks with these characteristics, we want to investigate the following coverage problem in the context of participatory sensing:

Given a large set of n possible users: What is the subset of m users such that the coverage C is maximized. Here, coverage

can be temporal or spatial. Because of the mobility of the sensors it is important to study coverage in space and time.

Given this problem statement, we adapt a common greedy algorithm [5] to investigate the coverage using three different datasets. These datasets differ in their mobility characteristics in that they contain traces from bus routes (mobile, but bound to routes), taxis (mobile, but bound to streets), and people (unbound in their mobility). These datasets also differ in size from 658.425 to 17.762.489 unique GPS positions and number of distinct sensors from 33 to 10.336 sensors.

The scope of this study, thus, allows us to report results on several important questions, such as the increase in coverage per person, the difference between spatial and temporal coverage and the possibility of measuring both spatial and temporal coverage to rate users’ contributions to a crowd sourcing platform.

The contribution of this paper is as follows:

- We introduce the coverage problem in participatory sensor networks with non-controllable mobility.
- We discuss and adapt a simple greedy algorithm to select the next best sensor given the input data.
- We study and discuss coverage on three different dataset using the algorithm to select the best users from the set.

The remainder of the paper is structured as follows. Section II defines the coverage problem. Related work is reviewed in section III. In section IV, we present our approach and the greedy algorithm. Section V introduces the data sets and how this data is filtered and processed. The results are discussed in section VI and a conclusion is given in chapter VII.

II. DEFINITIONS

Coverage is a metric about the quality of service delivered by a sensor network. A common classification [6] distinguishes between three main coverage classes: area, point, and path coverage.

All three classes are concerned with *spatial* coverage. Here, a sensor node can cover the area or any point within its sensing range. We are faced with mobile nodes and, thus, a sensor travels along several paths. For a sensor i , let P^i be the set of all distinct paths $P^i = p_1, \dots, p_m$. The spatial coverage of this node is the buffer $B^i = \bigcup_j B(p_j)$, where $B(p_j)$ is the buffer around the path $p_j \in P^i$ given a node’s sensing range.

The total spatial coverage of a given dataset is the area A of the union of all buffers. Hence, spatial coverage in this paper is defined as:

Definition 1 (Spatial Coverage). *The spatial coverage A is given by $A = \bigcup_i B^i$.*

Here, we use $|A|$ to denote the area units of A (e.g., in square meters). We will use this notation throughout the paper for area or total measurement time, e.g., in seconds.

Most papers in literature are only concerned with spatial coverage. They do not address the temporal aspect of coverage. Nodes are either stationary, i.e., they will always cover the same area, or the nodes are only mobile to optimize the coverage. However, humans participating in participatory sensing will move freely. Depending on the design of the application they might not sense continuously. Hence, we are also concerned with *temporal* coverage. We define temporal coverage as a temporal union \bigcup^T of all time ranges covered by all paths P^i .

The temporal coverage T^i of a sensor node i is, thus, simply $T^i = \bigcup_j^T T_{p_j}$, where T_{p_j} is the time range covered by path p_j .

The total temporal coverage T can now be defined similarly to the spatial coverage.

Definition 2 (Temporal Coverage). *The temporal coverage T is given by $T = \bigcup_i^T T^i$.*

In general, coverage is often defined as *k-Coverage*, i.e., an area or time frame is said to be covered if it is covered by at least k sensors. For this study, we assume 1-coverage both for spatial and temporal coverage, i.e., a single sensor covering the space or time is sufficient.

Given these definitions, we are able to study the coverage problem in three real-world data set. The next section will frame the study in the context of related work, before we go into more details on the study itself.

III. RELATED WORK

Coverage in sensor networks has been studied extensively. One of the first definitions is given by Gage [1] in the context of military applications. He defines three types of coverage: (i) blanket coverage (how to achieve a static arrangement of sensors that maximizes the detection rate of targets appearing within an area), (ii) barrier coverage (how to achieve a static arrangement of sensors that minimizes the probability of an undetected penetration through an area) and (iii) sweep coverage, i.e., a moving barrier, which forms a balance between maximizing the number of detections per time and minimizing the number of missed detections.

Several papers address sweep [7] and barrier coverage [8], [9], [10], [11], [12] using either stationary or mobile sensor deployments.

Mobile sensor deployments are mostly done with robots. Techniques for dispersing robots in order to solve the coverage problems have been presented in [3], [13] and [14]. Deployments of sensors using flying robots were described in [4].

So and Ye [15] propose a coverage algorithm based on Voronoi Diagrams. Megerian et al. [16] analyze worst and best-case coverage using Voronoi diagrams and graph search algorithms.

Huang and Tseng [17] have presented an polynomial-time algorithm to solve the problem of k -coverage, i.e., where every point in an area is covered by at least k sensors. They also distinguish between *k-Unit-disk Coverage* and *k-Non-unit-disk Coverage* (where the sensors' sensing ranges are not equal).

Ahmed et al. [18] have proposed a distributed algorithm to evaluate the degree of confidence in the detection probability of events.

A theoretical framework to model spatial and temporal correlation in wireless sensor networks is presented in [19]. Liu and Cao [20] describe how to maximize spatial-temporal coverage by scheduling sensors during a specific network lifetime.

Adlakha and Srivastava [21] present a theoretical solution to find the critical density of sensors for complete coverage given certain characteristics of the sensors and targets. The result is evaluated with a simulation.

Given the depth of research there are a number of survey on coverage [2], [22], [23], [24], [25]. Fan and Jin [6] provide the classification into three categories we gave in the last section: (i) area coverage, (ii) point coverage, and (iii) path coverage.

Another survey is provided by Meguerdichian et al. [26]. They provide a comprehensive overview on the coverage problem in wireless ad-hoc sensor networks. They define coverage as a quality of service metric for sensor networks. The authors distinguish between deterministic and stochastic coverage (i.e., sensors are randomly deployed). Interestingly, their coverage graph obtained from a simulation shows an asymptotic behavior. Something we will further discuss in our results.

Our study is motivated by Liu et al. [27]. They show that sensor mobility can be exploited to improve coverage compared to immobile sensors. However, this is always a trade-off between spatial and temporal coverage. A trade-off that is not well understood for non-controllable mobility and participatory sensing.

Lastly, Zhou, Das and Gupta [5] present a greedy algorithm that aims at keeping only a small subset of sensors active in an densely deployed sensor network. Their goal is to achieve coverage that is both complete and connected (i.e., the communication graph induced by the selected subset is connected). We will introduce a similar greedy algorithm in the next section to study the coverage when selecting only a small active subset of mobile sensors.

A common theme among the related work is a strong focus on theoretical research, validating the results using simulation. Over the next sections, we devise a greedy algorithm to study spatial and temporal coverage on three real-world data sets. This is especially important in the case of non-controllable mobility, where movement patterns cannot be easily simulated or deviate from a priori assumptions.

IV. SYSTEM DESIGN AND GREEDY APPROACH

In this section, we describe the system and the adapted greedy algorithm applied to study both spatial and temporal coverage. Figure 1 highlights the general architecture of the

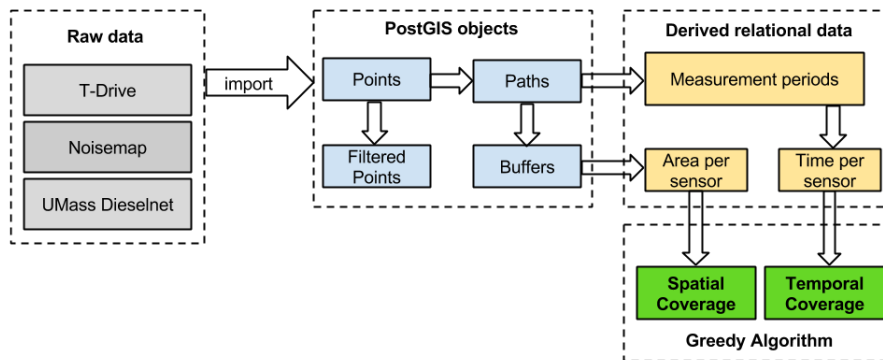


Figure 1. System Architecture

system. First, the raw data from the different data sets is imported to a PostgreSQL database with PostGIS extension. PostGIS adds support for the representation of spatial objects and allows to perform efficient computations on objects such as points, lines and spatial buffers. Data is imported into a relational structure to get a common representation of all the data, which is present in different text-based formats. From the filtering process we obtain the resulting filtered points and paths. A detailed description of the data sets and the filtering process is given in section V.

Based on the processed path, we create spatial buffers to calculate the area that can be sensed from this path. The baseline area is defined by the union of all these buffers, i.e., the entire area that is covered by every sensor in the network (cf. Section II). Obviously, we can also calculate the area covered by each sensor. Similarly, the measurement period of each path is computed, i.e., the time period between the timestamp of the first and the last point of this path. Again, we can compute the baseline for temporal coverage and the temporal coverage of each sensor. This derived data is the input for our greedy algorithm to compute both the spatial and temporal coverage.

A. Greedy Algorithm

To compute spatial and temporal coverage, we adapted a simple greedy algorithm that was described in [5]. We first describe a naive implementation of this greedy algorithm for spatial coverage:

Algorithm 1 Greedy Algorithm for Spatial Coverage

```

1:  $C = \emptyset$ 
2: while  $S \neq \emptyset$  do
3:   for  $i = 1$  to  $S.length$  do
4:      $C_i = B^{s_i} \cup C$ 
5:   end for
6:    $j = \arg \max_i (|C_i|)$ 
7:    $C = C_j$ ,
8:    $S = S \setminus s_j$ 
9: end while
  
```

Here, $S = \{s_1, \dots, s_n\}$ is the set of all sensors. Each sensor

has a spatial coverage of B^{s_i} , which is the union of all the buffers around the paths of this sensor. C is the total coverage achieved by selecting a subset of sensors.

In each iteration of the while loop, the greedy algorithm always selects the next best sensor, i.e., the one that leads to the maximum increase in overall coverage. Hence, we can state the following lemma:

Lemma 1. *Let j be the j -th iteration of the greedy algorithm and $\Delta|C_j|$ the coverage added in this iteration. Then $\forall j : \Delta|C_j| \leq \Delta|C_{j-1}|$*

However, we can also observe that selecting the next best sensor involves computing $B^{s_i} \cup C$ for each remaining sensor. This is unnecessary if the sensors are sorted by their spatial coverage and the coverage of a sensor remains larger than the coverage of the next sensor even though we subtract the already added area from his buffer. We can then pick this sensor as the next sensor in our greedy approach.

This idea can be formalized through the following lemma:

Lemma 2. *Let $S = \{s_1, \dots, s_n\}$ be the set of sensors sorted in decreasing order of coverage, i.e., $\forall i : |B^{s_i}| \geq |B^{s_{i+1}}|$. If $|B^{s_i} \setminus C| \geq |B^{s_{i+1}}|$ then s_i is the next best sensor.*

Proof: We will proof this by contradiction. Assume there is a $j > 1$ such that $|B^{s_{i+j}}| > |B^{s_i} \setminus C|$. Since $|B^{s_i} \setminus C| \geq |B^{s_{i+1}}|$ it follows that $|B^{s_{i+j}}| > |B^{s_{i+1}}|$. This contradicts the assumption that the sensors are sorted in decreasing order of coverage. ■

Given this shortcut, we can implement a more efficient greedy algorithm as shown in Algorithm 2.

Computing temporal coverage is analogous. Here, the operations are performed on time ranges instead of spatial buffers.

V. DATA SETS AND PROCESSING

Given the algorithms as outlined last section, we can now study spatial and temporal coverage based on three real-world data sets. Before we report our results, this section will describe the data sets and the necessary processing.

A. Data Sets

To generate meaningful results, we apply the outlined approach on three diverse datasets: We use mobility traces from

Algorithm 2 Optimized Greedy Algorithm for Spatial Coverage

```

1:  $C = \emptyset$ 
2: while  $S \neq \emptyset$  do
3:   for  $i = 1$  to  $S.length$  do
4:      $C_i = B^{s_i} \cup C$ 
5:     if  $|C_i| \geq |B^{s_{i+1}} \cup C|$  then
6:        $C = C \cup B^{s_i}$ ,
7:        $S = S \setminus s_i$ 
8:     goto 2
9:   end if
10:  end for
11:   $j = \arg \max_i (|C_i|)$ 
12:   $C = C_j$ ,
13:   $S = S \setminus s_j$ 
14: end while

```

Table I
STATISTICS ON THE RAW DATA

| Data type | Total number of data points | Number of sensors | Average points per sensor |
|-----------------|-----------------------------|-------------------|---------------------------|
| Noisemap | 658,425 | 504 | 1,306 |
| T-Drive | 17,762,489 | 10,336 | 1,719 |
| UMass Dieselnet | 1,037,460 | 33 | 31,438 |

(1) Noisemap, a real-world participatory sensing application, from (2) T-Drive, featuring taxi traces from Beijing, and the (3) UMass DieselNet data set, featuring traces from bus routes.

The three data sets chosen for the study feature distinctively different characteristics. They differ in size, number of sensors and the overall deployment scenario (cf. Table I). Most importantly, the sensors have different mobility pattern, i.e., the nodes have different restrictions on their mobility. We predict that these differences will have an impact on the coverage and the rate at which a certain coverage can be achieved.

1) *Noisemap Data*: Noisemap [28] is a participatory sensing project that aims at monitoring urban noise pollution using personal mobile devices as sensors. Noisemap is a participatory sensing application, where users opportunistically carry out measurements using their own personal devices.

Noisemap is part of the urban sensing platform *da_sense* [29], where readings from different kinds of sensors can be visualized on a web-based map or accessed through an API.

The application has been developed at the Technical University of Darmstadt, thus, most of the measurements are from Darmstadt (Germany). For our analysis, we will remove all measurements not within the boundaries of Darmstadt.

In this deployment scenario, users are carrying the mobile devices. Thus, there are no restriction to the mobility of the users. They might drive by car or walk freely and use the app whenever they wish at various location. Besides the lack of spatial restriction, we have to note that users will not sense continuously. This is important as it will impact temporal coverage. It is also the only dataset with churn, as users did join and leave the Noisemap system at their own discretion.

This again will have an impact on temporal coverage. To try and minimize this impact, incentives were used [30] in a more advanced version of Noisemap in order to motivate participants to increase measurement times.

2) *T-Drive Taxi Traces*: The T-Drive project¹ has collected data traces from thousands of taxis operating in Beijing. This data has been used to mine smart driving directions from historical GPS trajectories recorded by the taxis [31], [32], [33]. Unfortunately, the complete set of data has not been released to the public. However, a large sample has been made available. This dataset contains 17,762,489 data points featuring 10,336 unique sensors. It is the largest dataset used in this study.

In the T-Drive dataset sensors are more restricted in their movement compared to Noisemap since taxis are obviously bound to streets. However, on these streets they can move freely within the rules.

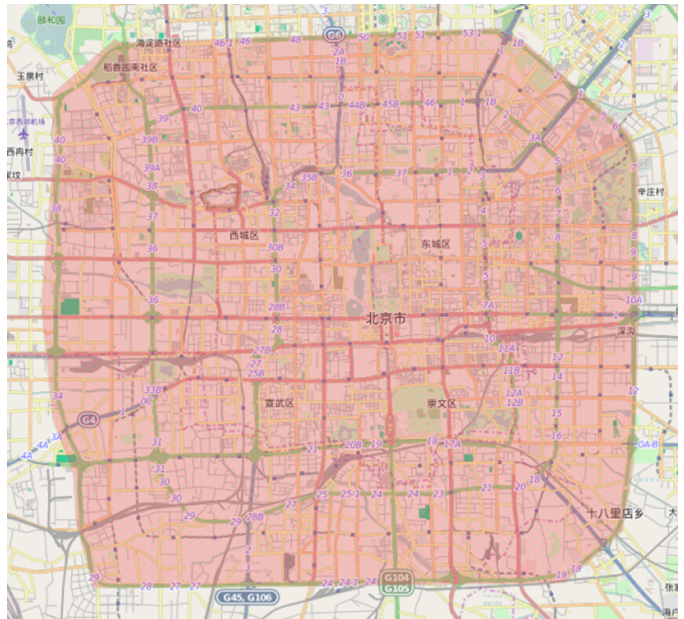


Figure 2. Boundaries for the T-Drive dataset (Scale: 1:180,000)

Also, we expect the taxis to not be uniformly distributed across the area as they rely on customers. Hence, there should be a higher density in denser areas or around popular destinations within the city. Hence, we limited the spatial extend of the dataset to only cover the inner city as depicted in Figure 2.

3) *UMass DieselNet Bus Traces*: To further restrict mobility, we used the UMass Dieselnet mobility traces from fall 2007². During the UMass Dieselnet project, buses within the city of Amherst, Massachusetts were equipped with a computer and a wireless interfaces. The buses provided wireless connectivity to passengers and scanned the surroundings for other networks. In addition to that, buses recorded their traces

¹<http://research.microsoft.com/en-us/projects/tdrive/>

²<http://crawdad.org/download/umass/diesel/dieselnet-fall-2007.tar.gz>

using a GPS receiver and exchanged data among each other. This data has been used to study disruption-tolerant networks (DTN) [34], [35] and is made available through the Crawdad³ platform.

Similar to taxis, buses can only operate on streets and within the rules. However, they are also bound to routes defined by the operator. Hence, their mobility is further restricted. We therefore expect the overlap between different buses to be very high, leading to a strong decline in additional spatial coverage per sensor.

B. Data processing

The general architecture of the system was already illustrated in Figure 1. All three datasets are provided as CSV/TSV text files with slightly different formats. We then extract the timestamp, position (i.e. GPS coordinates), and a unique sensor ID from the data. As explained earlier, the data is then stored into a relational PostgreSQL database.

The data is then filtered and paths are generated. The results of filtering the data is a subset of the imported data and a mapping of data points to paths. From this, we construct PostGIS line and buffer objects to represent the path of a sensor and its sensing range. Figure 3 shows a graphical representation of these objects. Throughout this paper, we use a sensing range of 30m to generate the buffers. These buffers are then the input to the algorithm described in the last section.

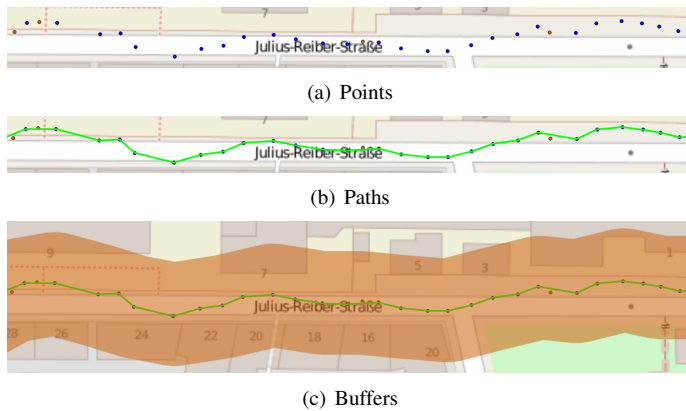


Figure 3. Constructing a representation of the spatial coverage - from single data points to buffers representing the sensing range

Filtering is, thus, the most crucial step before generating the paths, buffers, or any results on the data. Filtering the data is done for two reasons: First, raw data can be erroneous (e.g. a wrong GPS position) and secondly, to be able to perform analysis in reasonable time. Hence, we want to remove irrelevant or redundant data points. We applied the following filters to the data:

Invalid positions or timestamps We removed data points, where the latitude or longitude is equal to zero or the time is not within the bounds of the data set.

Equal positions or timestamps We removed a data point, if it either has the same position as the data point before (based on timestamp) or if two data points have the same timestamp.

Distance threshold If the distance between two consecutive points is below a certain threshold, the second point is dropped. This is done to reduce the number of raw data since data points that are very close in space and time are redundant. We chose this value to be 5 meters, because this is the maximum accuracy of GPS receivers in mobile devices.

Time threshold This filter does not remove any data points. Instead, data points are grouped into paths as described before. The time threshold sets the threshold for the time between two data points for those points to be considered belonging to one path. If the time difference between two data points is greater than this threshold, a new path is started. Since the users in the Noisemap dataset do not measure continuously this threshold is crucial to generate sensible paths. The time threshold was set to 10 minutes. Therefore, if two data points are further apart than 10 minutes, a new path is generated.

Speed threshold This represents the maximum speed between two data points. If the calculated speed between such two points is higher than the threshold, the second point is assumed to be an invalid sensor reading and is not added to the path. Figure 4 shows the differences in the paths that occur when different speed thresholds are set. This is the most significant filter and further discussed later.

All filters remove invalid data or reduce the data set significantly. Furthermore, given the time threshold, points are grouped into paths, as discussed last section.

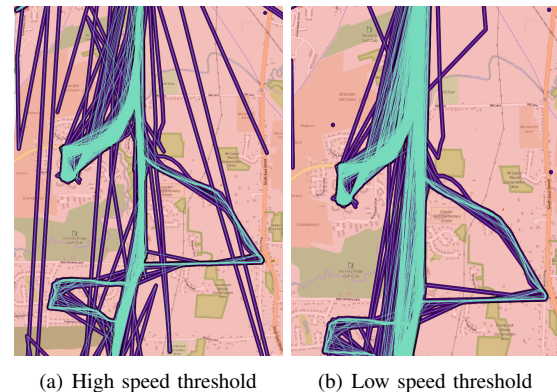


Figure 4. Different speed thresholds

In comparison, the speed threshold is the most sophisticated. It must be adjusted per data set as it has to reflect the mobility patterns in these networks. This threshold should remove no valid data points, while filtering out any unreasonable data. Hence, the threshold should be as close as possible to the maximum speed in the network. For example, a speed threshold of 20 kph might seem

³<http://crawdad.cs.dartmouth.edu/>

Table II
STATISTICS ON THE FILTERED DATA

| Data type | Points | Paths | Area baseline (square meters) | Time baseline (seconds) |
|--------------------|-----------|---------|----------------------------------|----------------------------|
| Noisemap | 21,193 | 837 | 6,400,975 | 364,443 |
| T-Drive | 6,377,074 | 672,105 | 314,513,617 | 529,575 |
| UMass Dieselnet | 427,475 | 3892 | 14,034,420 | 1,312,008 |

reasonable for sensors carried by humans, because humans will rarely walk faster than 20 kph. However, sensors deployed on taxis or buses will generally move at higher speeds. Hence, there is a need to evaluate the speed threshold to find a suitable value for a given data set. We evaluated the speed threshold (in kph) for speed $s \in \{200, 175, 150, 125, 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 5\}$ for each data set. The speed threshold affects the resulting number of data points and paths. Figure 5 illustrates this for all three data sets.

We want to set the speed threshold such that there is no significant drop in the number of data points. For Noisemap, we observe a sharp decline between 10kph and 5kph (cf. Fig. 5(a)). However, we already drop below 20,000 data points going from 20kph to 10kph. Hence, we chose 20kph for the Noisemap dataset.

Evaluating the other two dataset, we chose a speed threshold of 50 kph for T-Drive and 70 kph for the UMass Dieselnet dataset. This is, again, an interesting characteristic of the data set. The buses seem to be moving at the fastest pace. Hence, they should be able to cover a large spatial area in less time. The people in Noisemap, however, are moving at a much slower pace.

Table II shows statistics on the filtered data, including the number of data points and distinct paths left after filtering for each dataset. In addition, it provides the baselines for area and measurement times.

Given the three filtered data sets and the greedy algorithm, the next section will report the impact these different characteristics have on both spatial and temporal coverage.

VI. RESULTS

Based on the system and datasets described above, we can now study spatial and temporal coverage. This is the first study on spatial and temporal coverage for participatory sensing. The main question of our study is: How big does the subset of nodes need to be in order to achieve a certain degree of coverage. Translated to a participatory sensing scenario this means: If you had a budget for a subset of only x users, how high could the possible coverage be?

It follows from Lemma 1 that each iteration will add a sensor featuring a smaller increase in overall coverage. This is true for both spatial and temporal coverage. It will be interesting, how the overall coverage of the subsets will converge against the total coverage over all nodes.

We expect spatial coverage to converge faster when the mobility of sensors is restricted, e.g., with bus routes.

Temporal coverage is most interesting for the Noisemap dataset given the fluctuation of users in real-world participatory sensing scenarios. Because sensors in the system do not measure continuously and users do leave the system after some time, the opportunistic nature of participatory sensing should influence the temporal coverage. For a long deployment time, there might be significant gaps in temporal coverage in such applications. On the other hand, deployments on buses or taxis are generally active for longer periods of time, because sensors are permanently mounted. The sensors can also remain active during the time these vehicles are in operation. One might argue that therefore cars are better than humans in participating in sensing campaigns.

In each iteration, the algorithm will find the next best sensor and calculate the total area and the time that is covered up to this iteration. It will also calculate the area or time added in this iteration. We will first report results for spatial coverage before discussing temporal coverage.

1) *Spatial Coverage:* Again, let us summarize what we expect. We have three datasets with different mobility patterns. Spatial coverage is about overlap. We can assume that more spatial overlap between the sensors leads to a steeper increase in coverage. Hence, we expect the bus traces to converge faster against the baseline compared to the other two data sets.

In Figure 6, we plot the increase in coverage per sensor (cf. Fig. 6(a)) and the convergence of spatial coverage against the baseline (cf. Fig. 6(b)).

Obviously, the spatial coverage behaves as expected. Selecting only one sensor will lead to a spatial coverage of almost 50% of the overall coverage for the UMass dataset. So the first sensor covers 50% of the area all sensors cover. For Noisemap and T-Drive this number is close to 20%.

Now, the first sensor for T-Drive adds a little more coverage than the first Noisemap sensor. This is due to the fact that taxi routes do overlap. However, this is reversed for the 4th to 8th sensor. We explain this with the fact that Noisemap has a small number of really active participants. These participants might share some paths, but are generally more disjunct than the taxi routes. Hence, they will still add a larger area to the overall set. With both taxis and buses the network is formed by nodes with a common objective. Thus, the overlap is by design leading to a sharp decline in added coverage after the first few sensors. With Noisemap the overlap is not by design. Overlap in Noisemap happens in popular public places and if people live close to each other. This diversity in neighborhoods leads to a small plateau in spatial coverage.

We can conclude that depending on the nodes in your mobile network different nodes should be picked. For human participants, finding people that share almost no route are beneficial to the system. In reality this can only be decided after gathering the data so prediction methods are required to select the best users before any involvement into the platform. For networks of commercial nodes, the best node is the most active node. Given the amount of planning for bus and to a lesser extend taxi operations, this could be evaluated at design time.

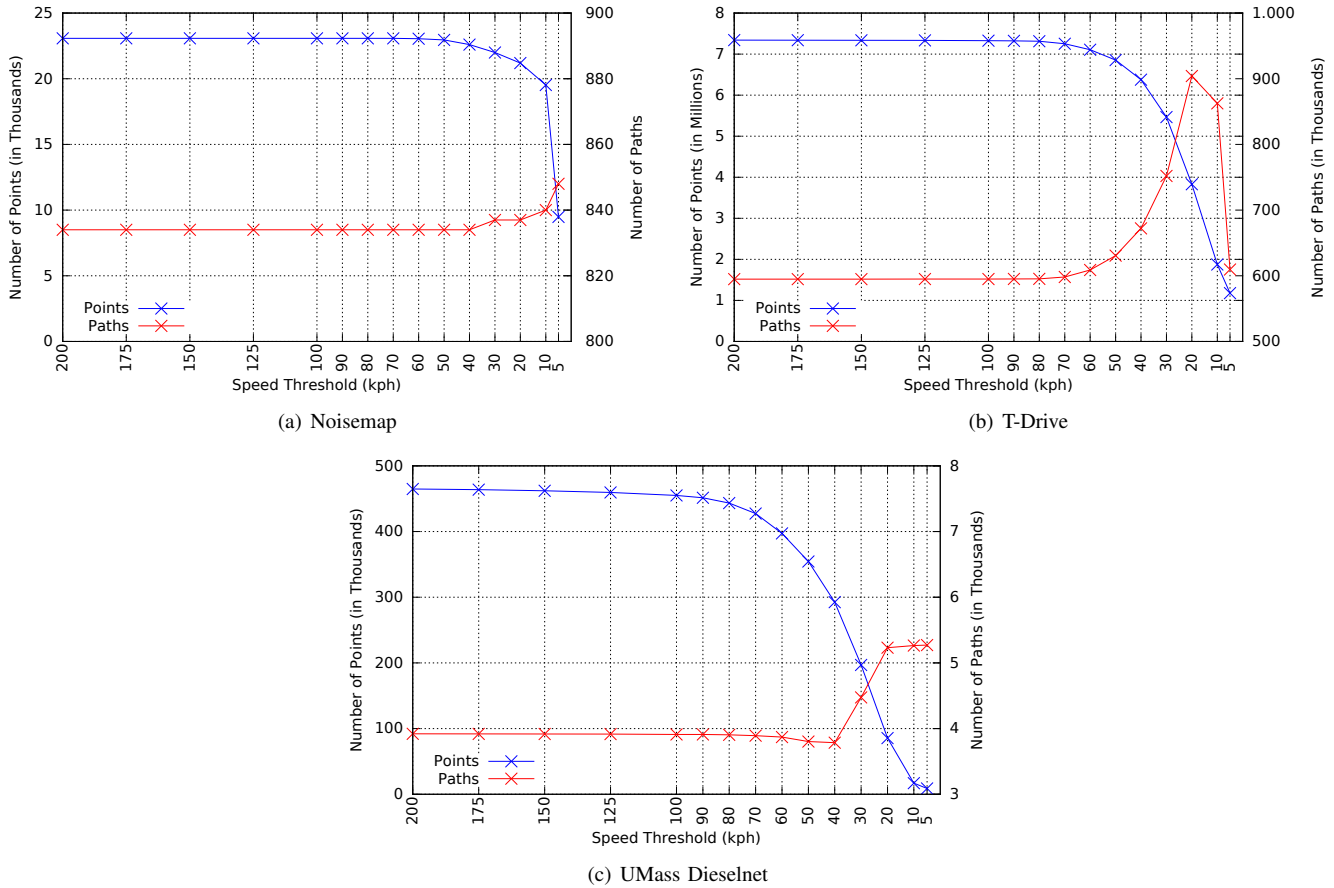


Figure 5. Evaluation of different speed thresholds for all three data sets

2) *Temporal Coverage*: For spatial coverage, most of the difference was down to the fact that T-Drive and UMass are commercial networks with a common goal and Noisemap is a spontaneous network of participating humans. We expect this to be the same for the temporal coverage. Both buses and taxis can sense continuously and as long as the vehicle is in operation it will send data. With Noisemap there is real churn, loss of interest, active participation, and, thus, a different temporal pattern. It was also collected over the course of years, while T-Drive and UMass are weeks only.

Again, we plot the increase in coverage per sensor (cf. Fig. 7(a)) and the convergence of temporal coverage against the baseline (cf. Fig. 7(b)).

As expected, UMass and especially T-Drive converge very fast. The first sensors account for 50% and 90% of the overall temporal coverage. For Noisemap, however, the first sensor accounts for only 20% of the temporal coverage and the overall behavior is much closer to that of spatial coverage. Even sensors that are chosen later add significant new time ranges to the system. We only reach 90% of the baseline coverage after adding 25 sensors.

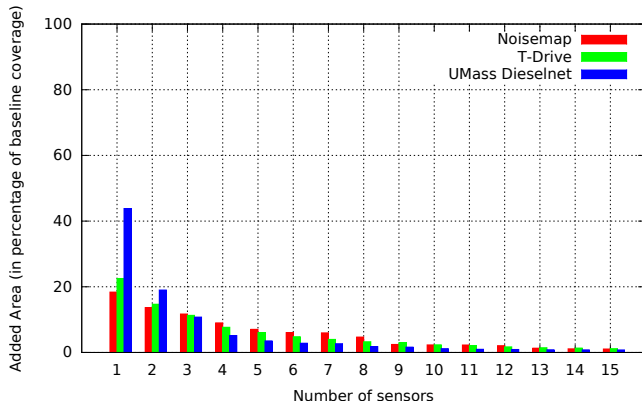
This is due to the churn observable in real-world systems and must be considered at design time. The system will lose all coverage, if new participants cannot be added to the system

at the same rate they are leaving. Increasing time participants spend with the system will have the most immediate effect on temporal coverage. Hence, for real-world system picking users based on spatial diversity is not enough. We want to also include users that will stay long enough or have enough time to measure throughout the day.

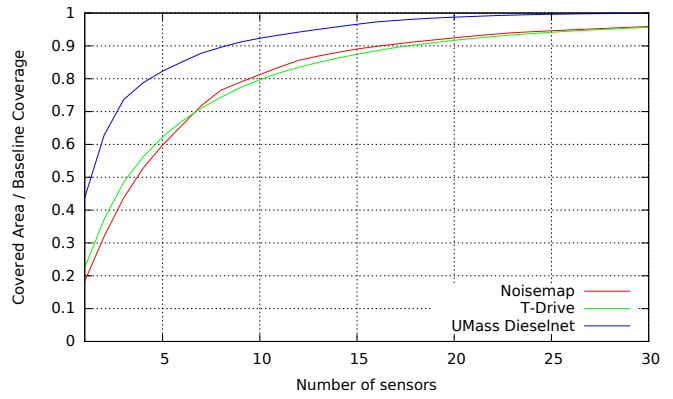
In summary, for both temporal and spatial coverage, we found that a relatively small number of sensors is sufficient to achieve good coverage. This is even true in real-world scenarios like Noisemap, where users behave unpredictably. Table III summarizes how many sensors are needed to achieve coverage rates of 50%, 70%, and 90%. For spatial this is always less than 20. So for T-Drive, 20 sensors achieve 90% of the overall coverage of over 10,000 sensors combined. Picking the right sensors to build a system based on limited budget is, thus, crucial in achieving high coverage, especially if it involves human participants.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have analyzed the problem of both spatial and temporal coverage in sensor networks that are neither stationary nor mobile all the time. We have presented a simple greedy algorithm to select the next best sensor (i.e. the one that adds most coverage given an already covered subset of the area or time). The evaluation performed on different sets of

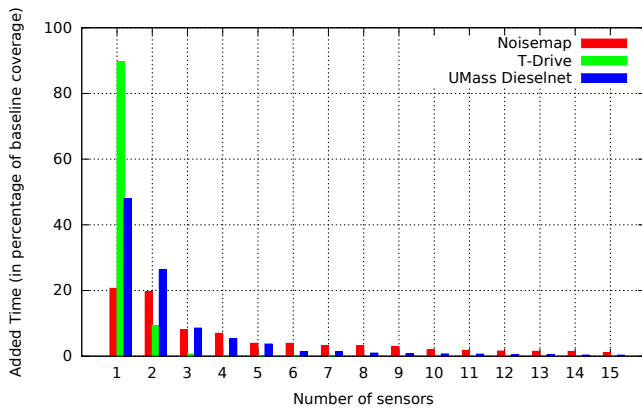


(a) Increase in coverage per Sensor

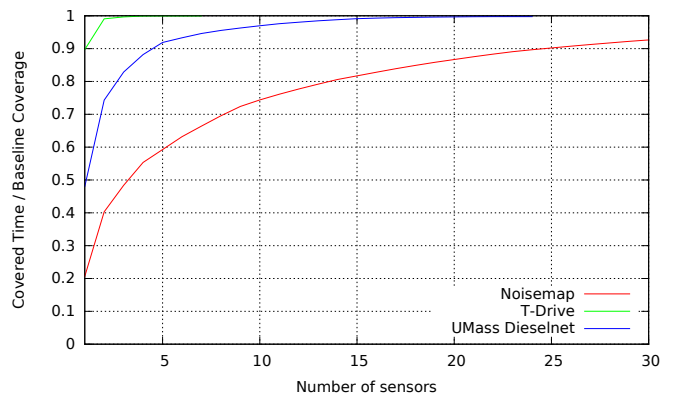


(b) CDF

Figure 6. Spatial Coverage



(a) Increase in coverage per Sensor



(b) CDF

Figure 7. Temporal Coverage

Table III
NUMBER OF SENSORS REQUIRED TO ACHIEVE A CERTAIN PERCENTAGE OF COVERAGE

| Coverage | Noisemap | | T-Drive | | UMass Dieselnet | |
|----------|----------|----------|---------|----------|-----------------|----------|
| | spatial | temporal | spatial | temporal | spatial | temporal |
| 50% | 4 | 4 | 4 | 4 | 2 | 2 |
| 70% | 7 | 9 | 7 | 7 | 3 | 2 |
| 90% | 17 | 25 | 18 | 18 | 9 | 5 |

data showed that the rate at which coverage increases depends heavily on the mobility pattern of the sensors.

In future work, we will study spatio-temporal coverage, i.e. how to combine the analysis of spatial and temporal coverage. The question here will be: Given an already selected set of sensors that cover a certain area at a certain time, the next best sensor should cover new areas at times at which no sensor has performed a measurement yet.

The results of coverage analysis can be used to estimate the number of sensors needed to achieve a certain coverage in similar deployment scenarios. Furthermore, this helps to answer the question which sensors should be selected to perform dynamic queries in a sensor network. In such a scenario, coverage can be one metric among others to determine the

best sensor.

ACKNOWLEDGMENTS

This work has been funded by the German Research Foundation (DFG) as part of project B02 within the Collaborative Research Center (CRC) 1053 – MAKI.

REFERENCES

- [1] D. Gage, "Command control for many-robot systems," in *Proceedings of the Nineteenth Annual AUVS Technical Symposium*, vol. 10, 1992.
- [2] B. Wang, "Coverage problems in sensor networks," *ACM Computing Surveys*, vol. 43, no. 4, pp. 1–53, Oct. 2011.
- [3] M. Batalin and G. Sukhatme, "Spreading out: A local approach to multi-robot coverage," in *Distributed Autonomous Robotic Systems 5*, H. Asama, T. Arai, T. Fukuda, and T. Hasegawa, Eds. Springer Japan, 2002, pp. 373–382.
- [4] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme, "Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 4, April 2004, pp. 3602–3608 Vol.4.
- [5] Z. Zhou, S. Das, and H. Gupta, "Connected k-coverage problem in sensor networks," in *Proceedings of 13th IEEE international conference on Computer Communications and Networks*, Oct 2004, pp. 373–378.
- [6] G. Fan and S. Jin, "Coverage Problem in Wireless Sensor Network: A Survey," *Journal of Networks*, vol. 5, no. 9, pp. 1033–1040, Sep. 2010.

- [7] M. Li, W. Cheng, K. Liu, Y. Liu, X. Li, and X. Liao, "Sweep coverage with mobile sensors," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1534–1545, Nov. 2011.
- [8] B. Liu, O. Dousse, J. Wang, and A. Saipulla, "Strong barrier coverage of wireless sensor networks," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '08. New York, NY, USA: ACM, 2008, pp. 411–420.
- [9] S. Kumar, T. H. Lai, and A. Arora, "Barrier coverage with wireless sensors," in *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking*, 2005, pp. 284–298.
- [10] A. Chen, T. H. Lai, and D. Xuan, "Measuring and guaranteeing quality of barrier-coverage in wireless sensor networks," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2008, pp. 421–430.
- [11] A. Saipulla, B. Liu, G. Xing, X. Fu, and J. Wang, "Barrier coverage with sensors of limited mobility," in *Proceedings of the 11th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2010.
- [12] C. Shen, W. Cheng, X. Liao, and S. Peng, "Barrier coverage with mobile sensors," in *Parallel Architectures, Algorithms, and Networks, 2008. I-SPAN 2008. International Symposium on*, May 2008, pp. 99–104.
- [13] M. A. Batalin and G. S. Sukhatme, "Coverage, exploration and deployment by a mobile robot and communication network," *Telecommunication Systems*, vol. 26, no. 2-4, pp. 181–196, 2004.
- [14] J. Pearce, P. Rybski, S. Stoeter, and N. Papanikolopoulos, "Dispersion behaviors for a team of multiple miniature robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2003, pp. 1158–1163.
- [15] A. M.-C. So and Y. Ye, "On solving coverage problems in a wireless sensor network using voronoi diagrams," in *Proceedings of the 1st Conference on Internet and Network Economics*, ser. WINE'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 584–593.
- [16] S. Megerian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Worst and best-case coverage in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 1, pp. 84–92, 2005.
- [17] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," in *Proceedings of the 2Nd ACM International Conference on Wireless Sensor Networks and Applications*, ser. WSNA '03. New York, NY, USA: ACM, 2003, pp. 115–121.
- [18] N. Ahmed, S. Kanhere, and S. Jha, "Probabilistic coverage in wireless sensor networks," in *Proceedings of the IEEE Conference on Local Computer Networks*, 2005, pp. 672–681.
- [19] I. Akyildiz, M. Vuran, and O. Akan, "On exploiting spatial and temporal correlation in wireless sensor networks," *Proceedings of WiOpt*, vol. 4, pp. 71–80, 2004.
- [20] B. Liu and D. Towsley, "A study of the coverage of large-scale sensor networks," in *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, Oct 2004, pp. 475–483.
- [21] S. Adlakha and M. Srivastava, "Critical density thresholds for coverage in wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking*, vol. 3, no. C, 2003, pp. 1615–1620.
- [22] H. M. Ammari, "Coverage in Wireless Sensor Networks: A Survey," *Network Protocols and Algorithms*, vol. 2, no. 2, Jun. 2010.
- [23] J. Chen and X. Koutsoukos, "Survey on coverage problems in wireless ad hoc sensor networks," in *Proceedings of IEEE SouthEastCon*, 2007.
- [24] M. Cardei and J. Wu, "Coverage in wireless sensor networks," *Handbook of Sensor Networks*, pp. 1–12, 2004.
- [25] A. Ghosh and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: A survey," *Pervasive and Mobile Computing*, vol. 4, no. 3, pp. 303–334, 2008.
- [26] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 3, 2001, pp. 1380–1387 vol.3.
- [27] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, "Mobility improves coverage of sensor networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2005, pp. 300–308.
- [28] I. Schweizer, R. Bärtl, A. Schulz, F. Probst, and M. Mühlhäuser, "Noisemap - real-time participatory noise maps," in *Second International Workshop on Sensing Applications on Mobile Phones*, 2011.
- [29] A. Schulz, J. Karolus, F. Janssen, and I. Schweizer, "Accurate Pollutant Modeling and Mapping: Applying Machine Learning to Participatory Sensing and Urban Topology Data," in *IEEE Natsys*, 2015.
- [30] I. Schweizer, C. Meurisch, J. Gedeon, R. Bärtl, and M. Mühlhäuser, "Noisemap - Multi-tier incentive mechanisms for participative urban sensing," in *Third International Workshop on Sensing Applications on Mobile Phones*, 2012.
- [31] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-drive: Driving directions based on taxi trajectories," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2010, pp. 99–108.
- [32] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "T-drive: Enhancing driving directions with taxi drivers' intelligence," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 220–232, Jan 2013.
- [33] Y. Zheng, J. Yuan, W. Xie, X. Xie, and G. Sun, "Drive smartly as a taxi driver," in *Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing in Conjunction with the UIC 2010 and ATC 2010 Conferences*, 2010, pp. 484–486.
- [34] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications*, April 2006, pp. 1–11.
- [35] X. Zhang, J. Kurose, B. N. Levine, D. Towsley, and H. Zhang, "Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing," in *Proceedings of the 13th annual ACM International Conference on Mobile Computing and Networking*, 2007, pp. 195–206.