Spring 2018

Optimal Constrained Wireless Emergency Network Antennae Placement

Swapnil Mohan Gaikwad
San Jose State University

Recommended Citation
Gaikwad, Swapnil Mohan, "Optimal Constrained Wireless Emergency Network Antennae Placement" (2018). *Master's Projects*. 614.
DOI: https://doi.org/10.31979/etd.z8w5-h6nc
https://scholarworks.sjsu.edu/etd_projects/614

Optimal Constrained Wireless Emergency Network Antennae Placement

A Writing Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfilment

of the Requirements for the Degree

Master of Computer Science

By

Swapnil Mohan Gaikwad

Spring 2018

SAN JOSÉ STATE UNIVERSITY

The Undersigned Thesis Committee Approves the Thesis Titled

Optimal Constrained Wireless Emergency Network Antennae Placement

By

Swapnil Mohan Gaikwad

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

Dr. Melody Moh, Department of Computer Science 05/07/2018

Dr. David Anastasiu, Department of Computer Engineering 05/07/2018

Dr. Teng Moh, Department of Computer Science 05/07/2018

ABSTRACT

With increasing number of mobile devices, newly introduced smart devices, and the Internet of things (IoT) sensors, the current microwave frequency spectrum is getting rapidly congested. The obvious solution to this frequency spectrum congestion is to use millimeter wave spectrum ranging from 6 GHz to 300 GHz. With the use of millimeter waves, we can enjoy very high communication speeds and very low latency. But, this technology also introduces some challenges that we hardly faced before. The most important one among these challenges is the Line of Sight (LOS) requirement. In the emergent concept of smart cities, the wireless emergency network is set to use millimeter waves. We have worked on the problem of efficiently finding a line of sight for such wireless emergency network antennae in minimal time. We devised two algorithms, Sequential Line of Sight (SLOS) and Tiled Line of Sight (TLOS), both perform better than traditional algorithms in terms of execution time. The tiled line of sight algorithm reduces the time required for a single line of sight query from 200 ms for traditional algorithms to mere 1.7 ms on average.

ACKNOWLEDGMENT

I would like to express my honest gratitude to my project advisor Dr. Melody Moh for her continuous guidance, time, and assistance during this project. I am grateful, and I thank my committee members Dr. David C. Anastasiu and Dr. Teng Moh for their thoughts, time, and suggestions.

Special thanks to Dr. David C. Anastasiu and Department of Computer Engineering for providing access to the hardware required for this project. Special tanks to Dr. Teng Moh for teaching me approximation algorithms which were useful in methods devised in this project.

We would also like to thank Department of Computer Science for help in setting up this project defense. We would also like to express our thanks to the library for giving required books and materials to learn different aspects and concepts of our project and extended study hours for students.

Last but not the least, I would also like to heartfully thank my friends and family for their moral support, encouragement, and blessings without which I could not able to finish my project successfully.

TABLE OF CONTENTS

1. Introduction.. 10

2. Related Work 14

3. Proposed Framework 16

3.1. Wireless Emergency Network Planning 16

3.2. Antennae Links 18

4. Estimating Line of Sight 21

4.1. Sequential Line of Sight..21

4.2. Tiled Line of Sight 24

5. Experimental Setup.. 28

5.1. Execution environment 28

5.2. PostgresSQL and PostGIS extension 29

5.3. qGIS 31

5.4. Test data 32

6. Results & Discussion 34

7. Conclusion 38

REFERENCES 39

TABLE OF TABLES

Table 1. Frequency Applications and Allocations in U.S. 11

Table 2. Algorithm SLOS 23

Table 3. Algorithm Tile Creation.. 25

Table 4. Algorithm TLOS 26

Table 5. Algorithm Populating building heights... 30

Table 6. Accuracy comparison for SLOS and TLOS 34

Table 7. Precision, recall, and F1-measure for SLOS.. 35

Table 8. Precision, recall, and F1-measure for TLOS 35

TABLE OF FIGURES

Figure 1 Tower with multiple point-to-point antennae 17

Figure 2 Communication between a transmitting antenna (TX) and receiving antenna (RX) 19

Figure 3 LOS obstruction scenarios. (a) Building C is not tall enough to obstruct the LOS between buildings A and B. (b) Building C obstructs the LOS between buildings A and B. 21

Figure 4 Potential Fresnel obstruction 22

Figure 5 Surface area tiling in the TLOS algorithm. Green tiles have a maximum height lower than the minimum intersecting point of the transmission line slope, while red tiles exceed that height 26

Figure 6 qGIS visualization of building structures in city of San José........................31

Figure 7 Height data superimposed in building structures 32

Figure 8 FCC tower location links data on Google Fusion table................................33

Figure 9 Distribution of execution time in our effectiveness experiment 36

Figure 10 Time-Distance co-relation of line-of-sight queries using SLOS and TLOS algorithms 37

LIST OF ABBREVIATIONS

WEN – Wireless Emergency Network

LOS – Line of Sight

SLOS – Sequential Line of Sight

TLOS – Tiled Line of Sight

IoT – Internet of Things

VHF – Very High Frequency

UHF – Ultra High Frequency

GIS – Geographical Information Systems

1. Introduction

In case of natural disaster or an emergency, the ability to communicate with outside world is of the highest consequence to resolve the situation. A large floor or an earthquake may damage power cables, telephone lines or cellular antennae resulting in loss of traditional communication lines such as the landline phones or the cellphones. In such situations, emergency responders still must have a reliable communication with each other and outside world to effectively mitigate the situation. To address this need, Office of Emergency Services (OES) of City of San José and similar offices in other major cities are deploying Wireless Emergency Networks (WENs). These networks will allow fire department, police department, hospitals, shelters, and food banks to communicate securely with each other and the outside world. For deploying these WENs, the most important question is to identify suitable locations to install wireless access points (APs) within the limits of cities.

The WENs will be constructed using millimeter wave antennae. There are multiple reasons for moving from traditional microwave to millimeter wave technology. The prime reason among them is congestion of microwave frequency spectrum, 3 kHz to 6 GHz. There are a wide variety of applications that already use wireless communications and almost all of them use microwave frequency spectrum. Microwave frequency ranges are reserved for these specific usages and applications. Some of these applications include aviation, military and government use, television broadcasting, radio broadcasting, global positioning systems (GPS), RADARs, and even microwave ovens [1]. The reserved frequencies of some of these applications are listed in Table *1*. Hence, for wireless emergency networks, the obvious choice of frequency range was outside microwave, i.e. millimeter frequency spectrum, 6 GHz to 300 GHz. In the millimeter wave spectrum, frequency range 60 GHz to 70 GHz is reserved for mission-critical services. These services include healthcare, communications for self-driving cars, smart city infrastructures and similar mission-critical services. This allows wireless emergency networks, a mission-critical service to gain easily reserved spectrum in millimeter frequency.

Table 1. Frequency Applications and Allocations in U.S.

Frequencies	Allocated Purpose
510 – 535 kHz	Government
535 – 1605 kHz	AM radio
74.5 – 75.2, 108 – 137, 328.5 – 335.4, 960 – 1215, 1427 – 1525, 220 – 2290, 2310 – 2320, 2345 – 2390 MHz	Aviation
54 – 72, 76 – 88, 174 – 216, 470 – 608 MHz	Television broadcasting VHF and UHF
88 – 99, 100 – 108 MHz	FM radio broadcasting
824 – 849 MHz	Cellular telephones
1215 – 1240, 1350 – 1400, 1559 – 1610 MHz	Global Positioning System (GPS)
2.40 – 2.4835 GHz	Microwave ovens

In this project, we created a framework which allows a user to identify the best locations for wireless APs in the city limits. The locations are generated given multiple constraints. Generating such optimal AP locations is a multi-constrained optimization problem. Even though network antennae placement problems are studies for many years, the problem of placing WENs is unique due to use of millimeter waves. In this case, there are several additional constraints that we never faced before in microwave antennae placement. A wireless emergency network is created using several access points that use high-speed millimeter waves. So, existing algorithms are insufficient for optimally finding locations for these APs. The constraints posed include physical limitations of antennae such as transmission range, point to point transmission or sector transmission and many soft constraints. We consider following constraints on the overall network for finding optimal locations:

- *Antennae Type*: Millimeter wave antennae are generally point-to-point antennae, but even these antennae can be bidirectional or omnidirectional. In some cases, WENs can use sector antennae, but it reduces the range of transmission.
- *Range*: Depending on the range of antennae, we need to place the different number of antennae in the same area.
- *Line of Sight (LOS)*: The line of sight constraint is the most important difference between millimeter waves and microwaves. The millimeter waves cannot penetrate most of the solid subjects like terrain, buildings, bridges, etc. Hence, there must be a clear line of sight between transmitting and receiving antennae.
- *Minimum Degree*: To ensure safety and reliability of communication during times of disaster, every AP must have the ability to communicate with more than one AP.
- *Mobile Access Points*: In congested areas or in areas without reliable communication, mobile access points can be deployed in form of a cell on wheels (CoW) or small cells on drones.

In this project, we focus on identifying the line of sight (LOS) between two locations in the city. The rest of the report is organized as follows. Section 2 describes related works and their limitations. Section 3 gives an overview of component parts of our proposed framework. Section 4 includes two algorithms for calculating line of sight between two locations within the city. Section 5 and 6 describe the experimental setup and experimental result respectively. Section 7 concludes the report.

2. Related Work

There is very limited research present to address the placement of millimeter wave antennae. Many academic works are focused on automatic adjustment of receiver antennae in order to capture maximum signal for transmitting antennae. There are also studies on effects of obstructions and increasing or decreasing distance on strength of the signal. There are few studies which work on millimeter waves with similar constraints, but these studies were specifically focused on small range antennae up to 200 m [2] [3] [4] [5] [6]. Palizban et al. [5] and Szyszkowicz et al. [6] only considered only the single type of obstacles, buildings. They assumed these buildings as only 2-dimensional polygon shape on a flat map at an equal altitude as potential obstacles for signal transmission. In addition, none of these studies addressed additional constraints presented by wireless emergency networks discussed in section 1.

Many manufacturers of millimeter wave antennae have published online tools that help their customers deciding the locations for placing the antennae. These tools address the problem of the line of sight to some extent, but these are very primitive tools and have many limitations. Some of these limitations are man-made objects like buildings. Some tools can only address the line of sight problem for terrain, some actually consider buildings as an obstacle. SCADACore RF – Line of Sight tool [7] and AirLink - Outdoor Wireless Link Calculator [8] only consider natural topologies such as mountains. On the other hand, tools like HeyWhatsThat Path Profiler [9] and Solewise Surface Elevation Tool [10] consider man-made obstacles, but these tools do have limitations considering certain objects such as small objects like traffic signals or organic objects like trees as a potential hindrance.

Most the above tools use Google Maps Elevation API [11] to query the heights of terrain or buildings. Even though this API is fairly accurate, it is not designed considering WENs in mind. Rapidly changing city infrastructure, new buildings being constructed, and old buildings being demolished can render elevations data obsolete in such sources. Additionally, there must be multiple data sources for height estimation for both natural terrains and man-made objects. In our proposed extensible framework, city authority is able to address the need for frequent changes and multiple data sources to automatically identify the optimal locations for network antennae placement.

3. Proposed Framework

In this section, we discuss the components of our proposed framework. These components include wireless emergency network planning, i.e. what aspects need to be considered to efficiently plan WENs, millimeter wave links and their properties, and how can improve estimates of the height of any objects in the city.

3.1. Wireless Emergency Network Planning

In conventional network planning, we are concerned about sectoral coverage of the network. In other words, we are mainly focused on how we can serve more people with a minimum number of antennae. But in the problem like wireless emergency network planning, there is a very limited audience to receive these signals. Hence, rather than focusing on local coverage, we have to focus on high-speed connectivity between pairs of important locations which are at a considerably higher distance. If we have such high-speed connectivity between distant locations, the signal can be distributed using traditional techniques such as Wi-Max or Wi-Fi. To ensure reliable long-distance communication, point-to-point antennae are used. Figure 1 shows multiple point-to-point antennae hosted on a single tower.

There are two components for solving optimal constrained wireless emergency network antennae placement problem in the proposed framework. Component one provides an efficient way of estimating the line of sight between given points in terms of latitude and longitude on a map. This component considers all types of topographies, man-made structures and organic vegetation that can interfere wireless communication. The related systems discussed in sections 2 use Google Maps or in some cases Google Earth data for estimating heights. These sources as not specifically designed for the task at hand can be highly inaccurate for a long-distance line of sight estimations. Most of such sources assume the earth is flat and approximate the heights of small unimportant buildings. Component two will take a list of pairs of locations which have a clear line of sight and are suitable for emergency situations and produce the list of locations where the antennae must be placed. This component will use *min-k degree multi-constrained optimization* method to solve the optimization problem. The method ensures reliable k degrees of communication links between all access points.

Figure 1 Tower with multiple point-to-point antennae [1]

The component two algorithm will be based on recommender system techniques. If we have a list of selected access points which are optimal, next access point can be

[1] User:GeorgeLouis [CC BY 3.0 (https://creativecommons.org/licenses/by/3.0)], via Wikimedia Commons

added to this list based on following the top recommendation. This recommendation will be ranked based on most similar access point. The access points having most reliable locations and highest possible communication speed based on the constraints discussed in section 1 will be ranked higher. The algorithm can be repeated till optimal size of the network is reached. Even though this problem is not convex, some overall good solutions can be obtained if recommender systems based on tensor factorization and matrix, because they may capture inherent structures of this problem.

3.2. Antennae Links

The most important components of wireless emergency network planning are antennae links. The millimeter waves antennae emit ultra-high-frequency directed beams of waves to transmit data between two fixed points. Figure 2 demonstrates an example of such communication. The transmission antenna (TX) modulates the signal to encode transmission data. The antenna then sends the signal in the form of a directed beam in the atmosphere. As the waves propagate through the free space, some signal is deteriorated or gets lost. But on a clear day, such transmission can reach up to 65 km distance. The receiving antenna (RX) collects the signal transmitted by the antenna (TX) and re-translates it into binary form.

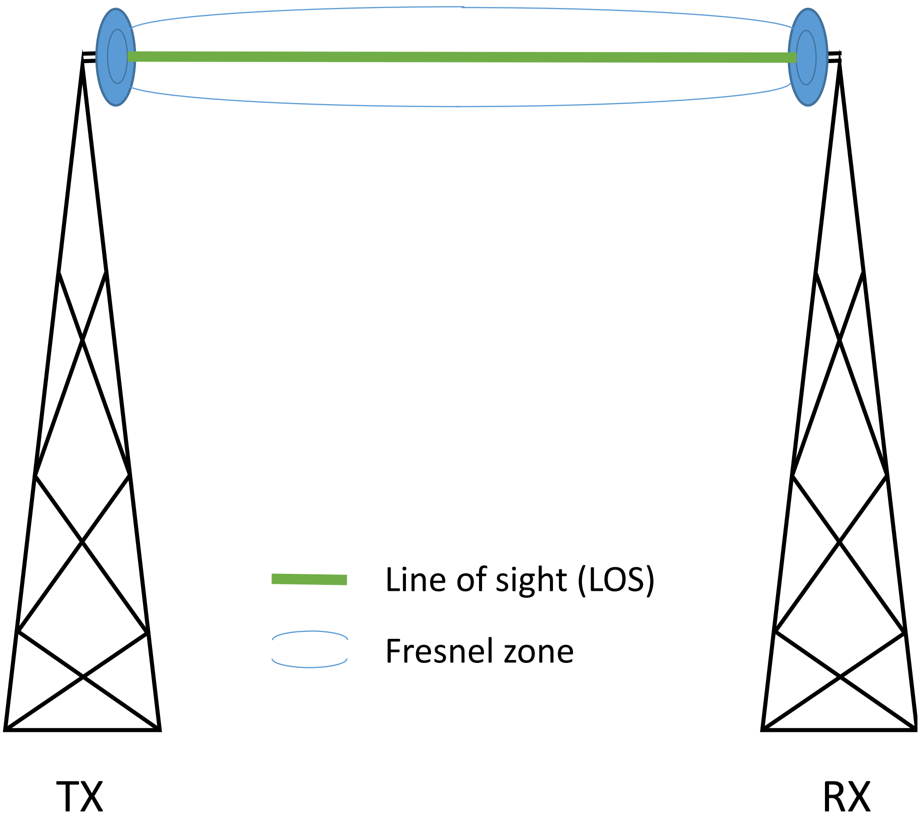

Figure 2 Communication between a transmitting antenna (TX) and receiving antenna (RX)

Even though millimeter waves can transmit huge amounts of data at almost the speed of light and over large distances, they still require a clear path between the antennae TX and RX, i.e. Line of Sight (LOS), for an effective transmission. Trees, buildings, bridges, and any other hindrance in the transmission path must be avoided either by relocating one or both of the antennae TX and RX points or by bridging to relay information. It is understood in the second case that bridge point has a clear path to both antennae TX and RX. A millimeter wave passing through a vegetation can be somewhat stronger in the winter but can be weaker once trees grow leaves in the spring.

Smooth terrains, water reservoirs, or any other reflective surfaces in the transmission path can reflect the signal in the unintended direction. This sometimes causes the signal to never arrive at receiving antenna RX. In some cases, the signal does arrive at RX, but it is out of phase. This causes significant degradation of the signal

[12]. Physicist Augustin-Jean Fresnel invented concentric ellipsoidal volumes in the transmitted radiation pattern that help visualize the reflection potential of the radio waves [13]. These ellipsoidal volumes are named as Fresnel zones, demonstrated in Figure 2. If Fresnel zone is obstructed more than 40%, transmission signal gets severely degraded. Hence, while predicting the line of sight, it is also important to consider Fresnel zone in addition to the direct path.

4. Estimating Line of Sight

Obstructions in communication path degrade the efficiency of the wireless network. Millimeter waves in wireless communication get reflected off buildings. When we choose a pair of locations for antennae placement, we have to make sure that there is a clear line of sight for optimal utilization of Fresnel zone ensuring maximum bandwidth. In this section, we discuss our two algorithms that may be used to effectively estimate the line of sight between two arbitrary locations within the city.

4.1. Sequential Line of Sight

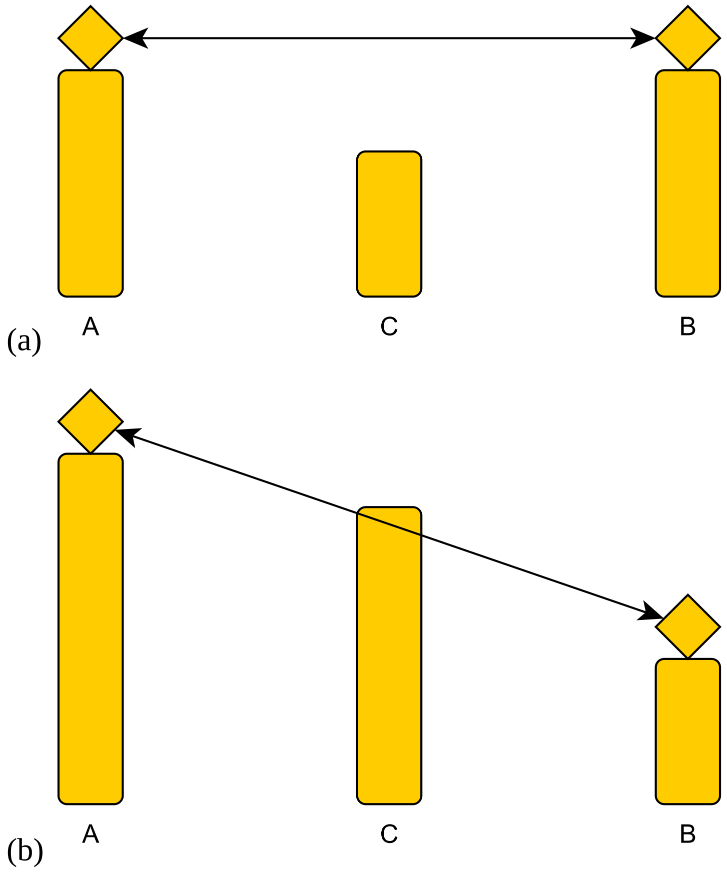

Figure 3 LOS obstruction scenarios. (a) Building C is not tall enough to obstruct the LOS between buildings A and B. (b) Building C obstructs the LOS between buildings A and B.

Our simple baseline algorithm of the line of sight estimation queries the GIS database for all buildings C between two selected antennae locations A and B. These buildings are OpenStreetMap polygons intersecting the path between the locations. Each returned building is then verified. Each building's height must be lower than the intersecting point on the slope from the location A to location B. Figure 3 shows two scenarios for one such building C. In the first case (a), it is clear that building C will not obstruct LOS between buildings A and B. The three buildings start at the same altitude, and buildings A and B are both taller than building C. In the second case (b), it is not as clear, as building B is not as tall as building C. Note, however, that LOS would be clear, for example, if building C and B were the same height. Specifically, the height of building C must be lower than the height of the transmission line (connecting the tops of the A and B buildings) at building C.

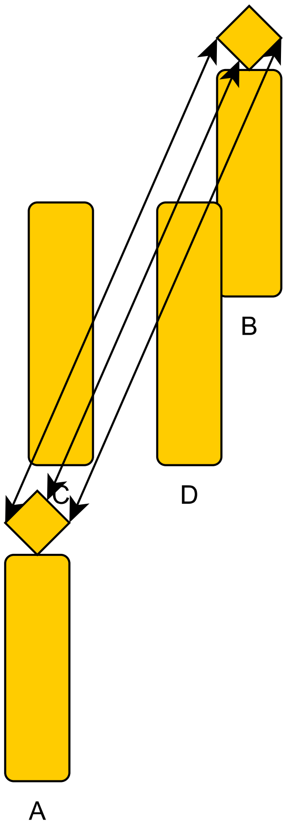

Figure 4 Potential Fresnel obstruction

One potential problem with the SLOS method is the case when the transmission line passes narrowly between two tall structures. As noted in Section II-B, the path between the TX and RX antennas should be clear not only on the direct line between them but also within at least 60% of the Fresnel zone of the two antennas. Figure 4 shows one such potential example. To account for this scenario, we execute two additional queries, for parallel lines situated at γ m on each side of the transmission line. SLOS then checks the height of buildings that were not present in the original query result to ensure they will not create obstacles in the transmission path. The SLOS method is formulated in Algorithm 1.

Table 2. Algorithm SLOS

Algorithm 1: Sequential Line of Sight

```
Input: Points p1 and p2 in terms of latitude and longitude)
Output: Binary Line of Sight Result
polygon poly1 = ST_Within(p1, polygons);
polygon poly2 = ST_Within(p2, polygons);
distance = ST_Distance(poly1, poly2);
diffHeigth = poly1.maxHeight - poly2.maxHeight;
$\theta$ = diffHeight / distance;
polygonsInWay = ST_Intersects(ST_MakeLine(poly1, poly2), polygons);
for each polygon in polygonsInWay do
    tempDistance = ST_Distance(poly1, polygon);
    allowedHeight = tempDistance*$\theta$ + poly1.maxHeight;
    if polygon.maxHeight > allowedHeight then
        return FALSE;
    end
end
return TRUE;
```

4.2. Tiled Line of Sight

In the sequential line of sight, we noticed some discrepancies like uneven building polygon size and missing buildings polygon in OpenStreetMap data. If a certain polygon is of a very big area, having a single height for that entire area reduces the accuracy of the Line of Sight queries. On the contrary, very small polygon structure does not efficiently reduce the time required for the query. To address these issues, we decided to draw our own artificial polygons of even sizes. We call this data structure as *Tile*. Unlike OSM polygons, each *Tile* is adjacent to another *Tile*. So, there is very less room for error due to non-polygon structures like area immediately outside a building. The *Tile* data structure also addresses the issue of overlapping polygons. Hence, we will not consider the same area twice in this data structure. These *Tiles* can then further be aggregated to higher level data structures. Figure 5 shows *Tile* representation and pooling [14]. These tiles are constructed and populated using Algorithm 2. Method for finding a *Tile* ID in this algorithm is derived from Bresenham's line algorithm for Computer Graphics [15].

In the case of a clear line of sight, the SLOS algorithm has to check each building that intersects the transmission line to ensure they will not obstruct transmission. We propose an efficient data structure that can significantly reduce the number of buildings whose height must be retrieved and considered, based on a hierarchical tiling of the city’s surface area. Figure 5 shows an initial decomposition of the surface area into equally sized tiles. Note that the number of tiles may be different between levels. In the example, green tiles have a maximum height lower than the minimum intersecting point of the transmission line slope, while red tiles exceed that height. Given the number of tiles per level, the tiling algorithm is straight-forward. For each tile, we pre-compute and store the highest elevation of any building in the tile. Note that the maximum heights of lower-level tiles can be easily aggregated to find the maximum height of a higher-level tile. The highest elevation in a higher-level tile is the maximum elevation among all child tiles.

Table 3. Algorithm Tile Creation

Algorithm 2: Tile Creation

Input: GIS data tuples, Number of rows, columns, and levels of Tiles

Output: Tile data structure

```
totalRows = rowslevels;
totalColumns = columnslevels;
tileWidth = (globalMaxLon - globalMinLon) / totalColumns;
tileHeight = (globalMaxLat - globalMinLat) / totalRows;
tileArray = newTile[totalRows*totalColumns];
for each point in tuples do
    id = getTileID(point, tileWidth, tileHeight);
    if tileArray[id].maxHeight < point .height then
        tileArray[id].maxHeight = point .height;
    end
end
return tileArray;
```

When predicting LOS, we first check the maximum height of tiles intersecting the transmission path. If a tile’s maximum height does not obstruct the transmission path, we can skip checking all the buildings within that tile. On the other hand, in the case of an obstruction, we can dig into the next lower level, retrieving child tiles that intersect the transmission path. At the lowest (leaf) level, data is stored without tiling and we determine the line of sight using the baseline SLOS approach. This method will significantly reduce the number of calculations required to solve the LOS prediction problem.

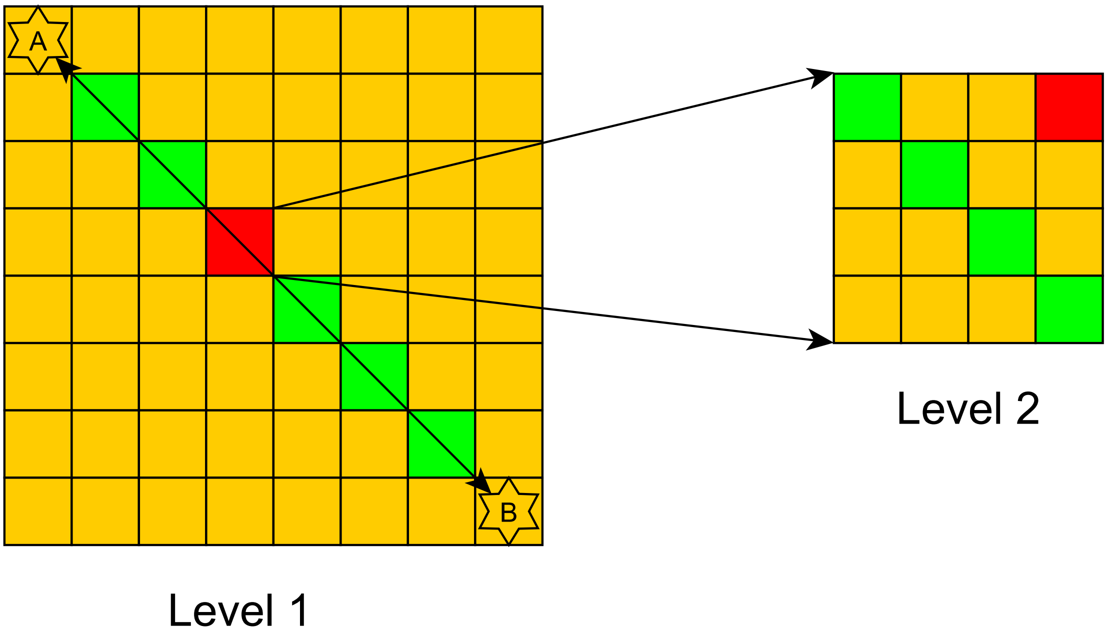

Figure 5 Surface area tiling in the TLOS algorithm. Green tiles have a maximum height lower than the minimum intersecting point of the transmission line slope, while red tiles exceed that height

This method is detailed in Algorithm 3.

Table 4. Algorithm TLOS

Algorithm 3: Tiled Line of Sight

Input: Points p1 and p2 in terms of latitude and longitude, level (default 1), allowed heights of points (default null)

Output: Binary Line of Sight Result

```
tile1 = getTile(p1, level);
tile2 = getTile(p2, level);
distance = distance(tile1, tile2);
```

```
diffHeigth = tile1.maxHeight - tile2.maxHeight;
$\theta$ = diffHeight / distance;
tilesInWay = BresenhamsLineAlgorithm(tile1, tile2, level );
for each tile in tilesInWays do
    tempDistance = distance(tile1, tile);
    allowedHeight = tempDistance*$\theta$ + tile1.maxHeight;
    if tile.maxHeight > allowedHeight then
        if level < totalLevels then
            point1, point2 = Points of intersection of BresenhamLine for tile;
            h1, h2 = Allowed heights at point1, point2;
            if ! RecursiveCall(point1, point2, level + 1, h1, h2) then
                return FALSE;
            end
        else
            return FALSE;
        end
    end
end
return TRUE;
```

5. Experimental Setup

In order to determine the effectiveness of our baseline method, we compared its ability to detect clear or obstructed LOS, which is solved as a classification problem. We use accuracy to measure the effectiveness of our method, which is defined as the ratio between the number of correctly predicted samples and the total number of samples predicted. Additionally, we report precision, recall, and F1-measure for the prediction. Precision is defined as the fraction of samples that are relevant in the prediction, recall is the fraction of relevant samples that were successfully predicted, and F1-measure is the harmonic mean of the precision and recall scores. Note that precision, recall, and F1-measure depend on which class is considered relevant and produce different results in an imbalanced binary classification scenario. We measure efficiency in the time (wall-clock) taken by each query, in milliseconds. In following subsections, details about execution environment and technologies used are explained.

5.1. Execution environment

Both algorithms SLOS and TLOS were executed on a server with dual-socket 12-core 2.5 GHz Intel Xeon E52680 v3 (Haswell) processors and 384 GB RAM. We also made sure that there are no other programs running on the server when these experiments were carried out. We used PostgreSQL version 9.6.1 to host our GIS database. As a proof of concept, we used a subset of OSM and LiDAR data covering the city of San José, which takes up 685GB of storage, including all indexes. While our system has 24 cores available, the PostgreSQL engine used only one core for executing each query. In all experiments, we set $\gamma = 6$.

5.2. PostgresSQL and PostGIS extension

For testing our both algorithms, we chose the city of San José as our testing ground. Mapzen [16] creates custom GIS data batches on request. We submitted a request for GIS data within the boundary of the city of San José. This data is basically OpenStreetMap (OSM) data [17]. This selected GIS data was in ‘shape’ formatted files, which is a common GIS data compression format. We used LibLas utility [18] to convert the GIS data in normal SQL statements using ‘shp2pgsql’ function. The SQL statements were then executed to create the 2-dimensional model of the city of San José. PostgreSQL needs an extension called PostGIS [19] to handle the GIS data. This extension allows us to create a column of type geometry, each of its cells can hold the structure of one building. These geometries can also be indexed efficiently using PostGIS so we can search these buildings using latitude and longitude. We did index the buildings for our SLOS algorithm.

The OSM data we imported in PostgreSQL database is only flat geometries without any height information associated with them. For our accurate height estimates, we used data obtained using LiDAR posted on nationalmaps.gov [20]. This data contains latitude, longitude, and height at distance of 1/3 arc second, which is approximately 10 meters. This data is also compressed in the form of ‘las’ file format. We used las2text utility again form LibLas to convert this data into comma separated values. These text files were again imported into PostgreSQL tables. This data contains records in the Lambert Conical projection format, which we had to convert to the WSG84 projection format of the OSM records. The guide provided by Yuriy Czoli [21] was very helpful in successfully converting between LiDAR and OSM data records.

Algorithm 4 details the PostGIS methods and technique used to populate the heights of the buildings in our database.

Table 5. Algorithm Populating building heights

Algorithm 4: Populating building heights

Input: Open Street Map polygons, GIS data tuples (latitude, longitude, height)

Output: Open Street Map polygons with heights

```
for each polygon in polygons do
    points = ST_Within(tuples, polygon);
    maxHeight = -6000.0;
    for each point in points do
        if point .height >maxHeight then
            maxHeight = point .height;
        end
    end
    polygon.maxHeight =maxHeight;
end
```

5.3.qGIS

qGIS [22] is the utility we used to visualize the information in PostgreSQL database. All the building structures in the city of San José can be seen in

Figure 6. Figure 7 shows the height data points superimposed on building geometries. In some locations where the LiDAR data was not available, we used Google Maps Elevation API to fill the missing information [11]. After this process, our database was ready to carry of the experiments.

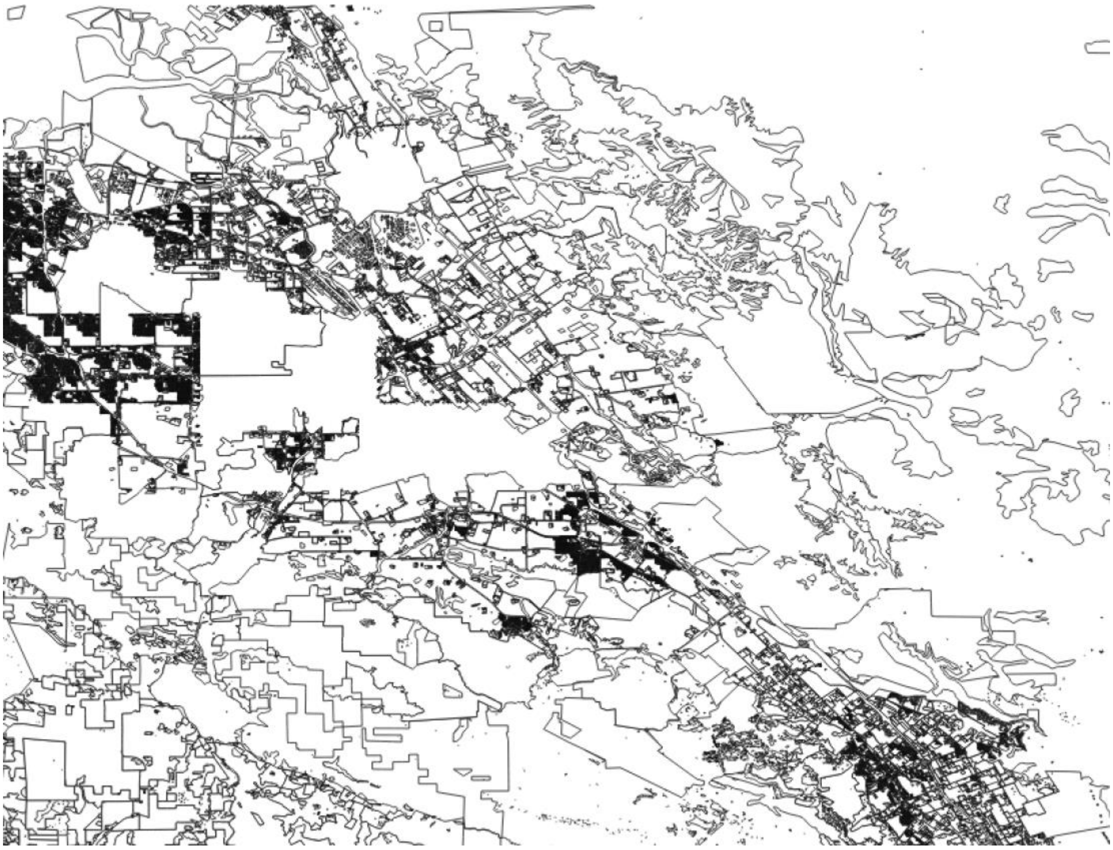

Figure 6 qGIS visualization of building structures in city of San José

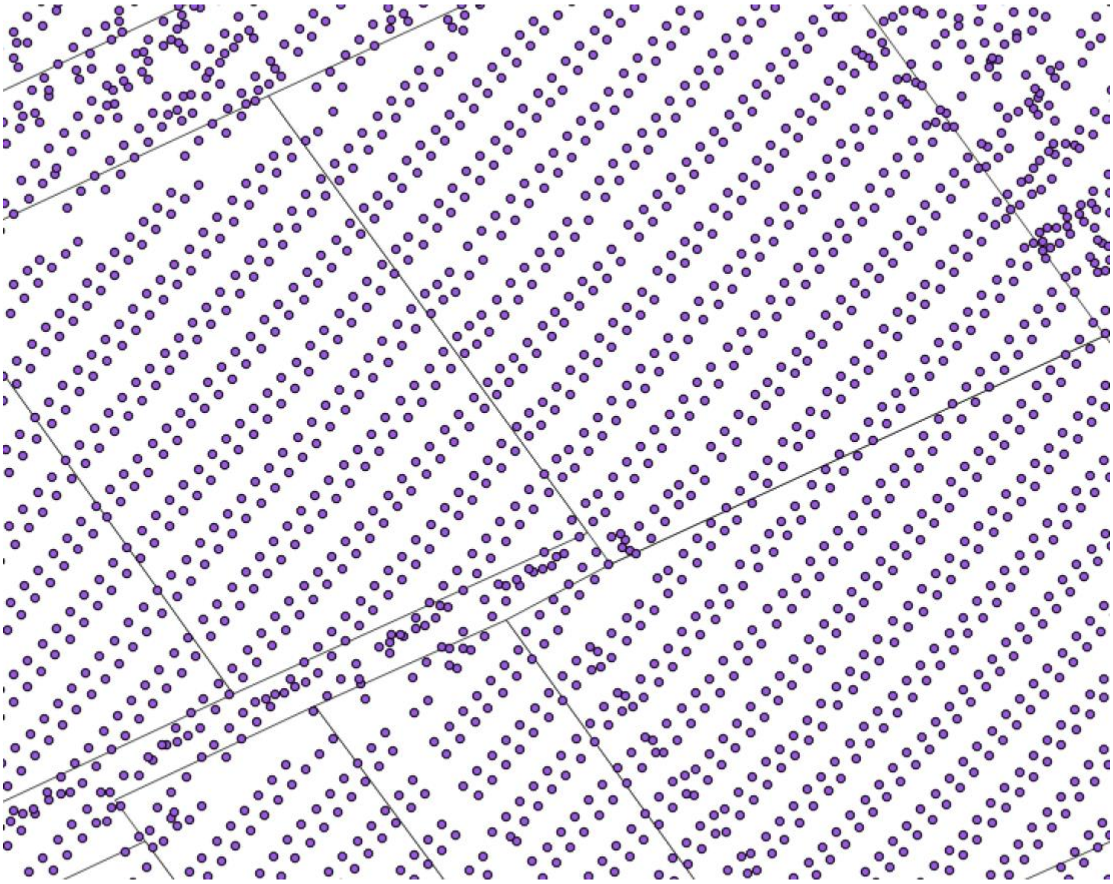

Figure 7 Height data superimposed in building structures

5.4. Test data

For our experiments, we used public FCC tower locations dataset [23] as true positive test data. This data is stored in form of Google fusion table. For the negative sample, we generated random pairs of points in the city of San Jose and tested them using online utilities. The negative results were stored as true negatives. For our efficiency tests described in section 6, we again used randomly generated 20000 data point pairs. But, in this case, the emphasis was on efficiency rather than the accuracy of the results.

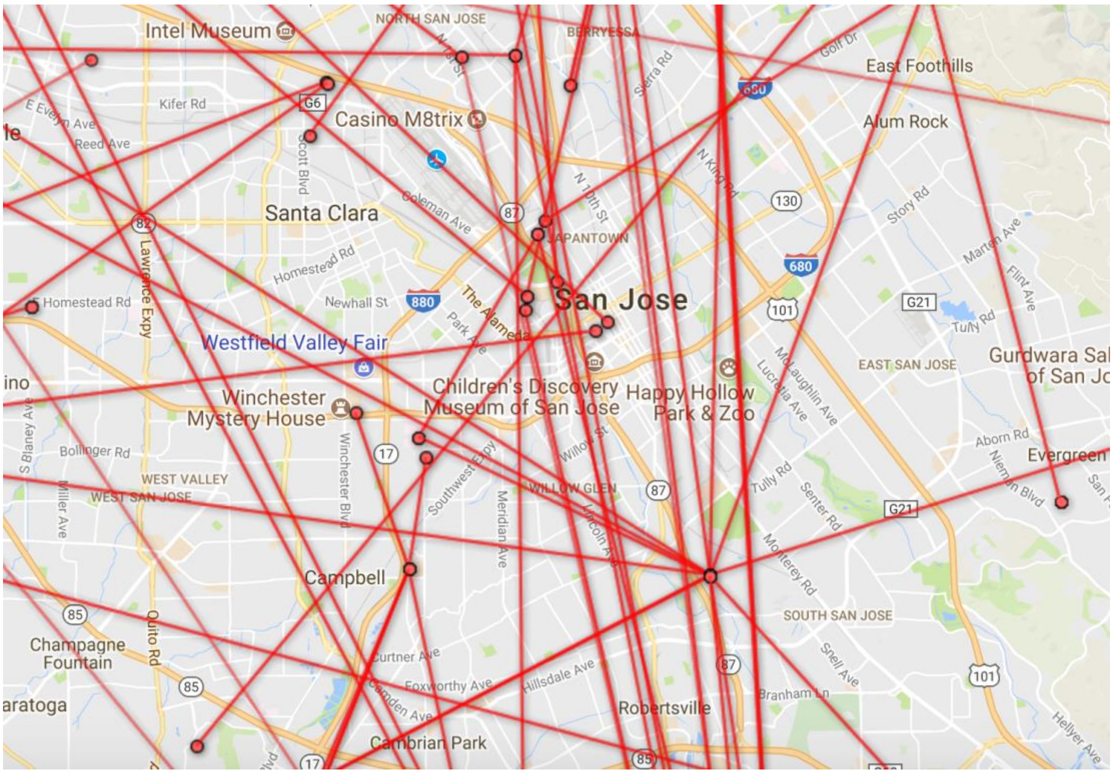

Figure 8 FCC tower location links data on Google Fusion table

6. Results & Discussion

We executed two experiments in order to test the efficiency and effectiveness of our baseline SLOS method and improved TLOS method. In the first experiment, we compared the ability to detect clear or obstructed LOS on 1506 our generated dataset. We selected LOS queries by choosing two buildings to form these sets from the city of San José and predicting the link between them. As we can see the results in Table 6, among the 1506 pairs of queries, 47 were misclassified, resulting in the accuracy of 96.87% for the sequential line of sight (SLOS) and only 18 were misclassified, resulting in the accuracy of 98.80% for the tiled line of sight (TLOS). These results were encouraging when compared to our last set of the experiment [14], as this dataset is balanced, and we had very good results in both cases clear and obstructed line of sight queries. As shown in Table 7, for SLOS, we had 96.07% precision, 96.67% recall, and 96.87% F1-measure for clear links. For obstructed links, we received 96.69% precision, 97.07% recall, and 96.88% F1-measure. As shown in Table 8, for TLOS, clear links had 99.19% precision, 98.40% recall, and 98.80% F1-measure and obstructed links had 98.41% precision, 99.20% recall and 98.80% F1-measure.

Table 6. Accuracy comparison for SLOS and TLOS

	SLOS	**TLOS**
Queries	1506	1506
Correct Classification	1459	1488
Incorrect Classification	47	18
Accuracy	96.87%	98.80%
Mode Execution Time	~60 ms	~1.7 ms

Table 7. Precision, recall, and F1-measure for SLOS

SLOS	Precision	Recall	F1-measure
Obstructed	96.69%	97.07%	96.88%
Clear	96.07%	96.67%	96.87%

Table 8. Precision, recall, and F1-measure for TLOS

TLOS	Precision	Recall	F1-measure
Obstructed	98.41%	99.20%	98.80%
Clear	99.19%	98.40%	98.80%

On average, the distance between selected points in the queries was approximately 17 km. The classification line in Figure 6 shows the distribution of execution time among the 200 queries in our effectiveness experiment. Note that the y-axis is log-scaled. Most queries took between 60 and 100 ms to execute for SLOS and between 1 and 2 ms for TLOS. The time improvement was also due to the parallel implementation of the queries in both the methods using multi-threading in Java. We also made sure to keep the threads alive after the execution and they could pick up next task in thread-safe blocking queues. This improved the performance by almost 20% than creating new threads for every query reducing the overhead of creation and destruction of threads.

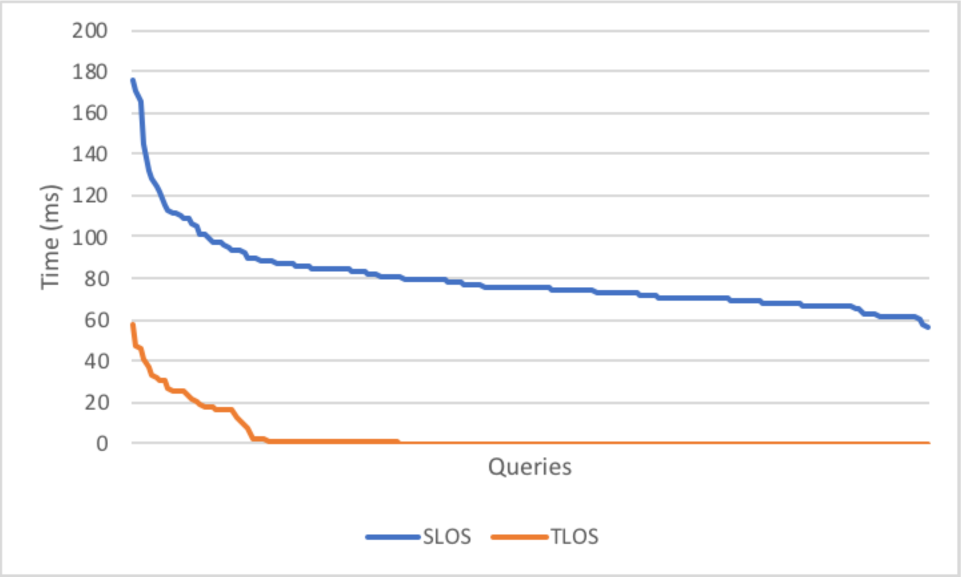

Figure 9 Distribution of execution time in our effectiveness experiment

Without the multi-threading architecture, the SLOS method tool 120 ms on average to execute and TLOS method took 17 ms on average. As a way to further investigate the efficiency of our method, we executed a set of 20,000 random LOS queries, recording the execution time for each. The random line in Figure 9 shows the distribution of execution times among these queries, which almost exactly matches that of the classification experiment, with the exception of the extreme start of the distribution. A few points in our classification experiments took 150-217 ms to execute in SLOS. In TLOS, the behavior was somewhat irregular. At the end of experiments, there were some spikes in execution time. Similarly, a few of the points in our random query experiment took 500-2567 ms to execute for SLOS, but in TLOS they took around 100-159 ms. However, we could not see a clear correlation between-point distance and query execution time among these few points in either experiment. We conclude that the high execution times in these few queries are likely due to the GIS system loading certain indexes from disk in case of SLOS. To better understand the relationship between query execution time and distance in our experiments, we computed the correlation between the two variables. The 200 sample classification experiment results showed a correlation of 0.49, while the 20,000 sample random experiment results showed a correlation of 0.21.

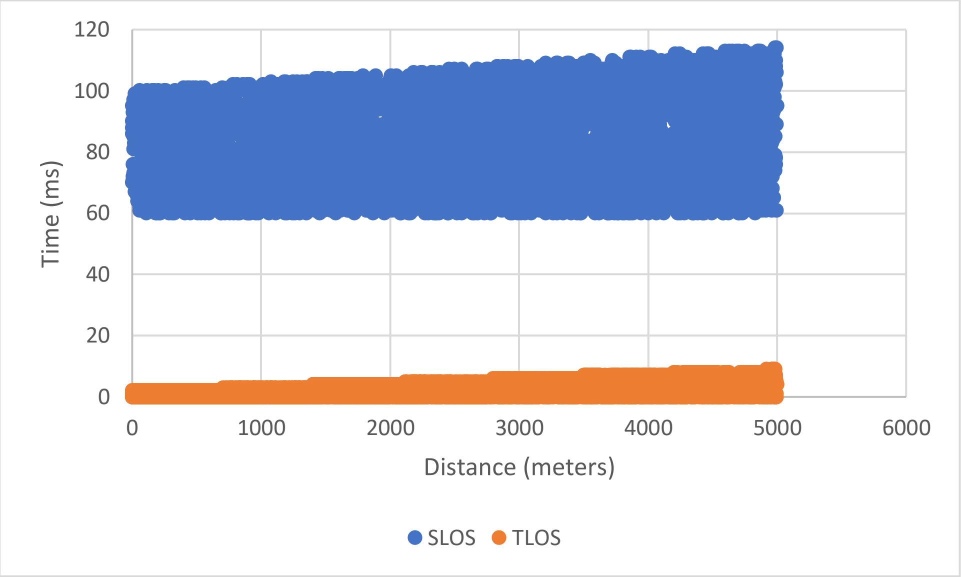

Figure 10 Time-Distance co-relation of line-of-sight queries using SLOS and TLOS algorithms

Figure 10 shows a scatter plot of the random experiment results, denoting the distance and execution time co-relation of these queries. While many queries take between 50–120 ms for SLOS algorithm and between 1 to 17 ms for TLOS, the graph shows a clear slightly positive correlation between distance and execution time. In the sequential line of sight method, queries for points farther than 4,000 m apart take longer to execute, in general, than those are closer together. Same queries have lesser co-relation in the tiled line of sight algorithm.

7. Conclusion

In this paper, we first presented a general framework for solving the optimal constrained wireless network antenna placement problem and then detailed two algorithms for efficiently identifying whether there is a clear line of sight between two locations in the city, which is a critical component in the framework. Our first baseline algorithm, SLOS, uses off-the-shelf GIS aware database systems and open-source data to effectively solve the problem, resulting in 96.87% accuracy in our initial experiments. While its average execution time of 60 ms may impede it from solving the optimization problem in a reasonable amount of time, our Tiling Line of Sight (TLOS) method delivered much higher efficiency by drastically reducing the number of structures whose height must be checked to ensure it does not obstruct LOS. TLOS promises the execution time of 1.7 ms on average at accuracy close to 98.80% which definitely is the best contender to be used in the constrained optimization system.

REFERENCES

[1] "Introduction to RF & Wireless Communications Systems," 23 September 2015. [Online]. Available: http://www.ni.com/tutorial/3541/en/.

[2] T. S. Rappaport, F. Gutierrez, E. Ben-Dor, J. N. Murdock, Y. Qiao and J. I. Tamir, "Broadband millimeter-wave propagation measurements and models using adaptive-beam antennas for outdoor urban cellular communications," *IEEE Transactions on Antennas and Propagation,* vol. 61, no. 4, p. 1850–1859, 2013.

[3] S. S. Sehra, J. Singh and H. S. Rai, "A systematic study of openstreetmap data quality assessment," in *Proceedings of the 2014 11th International Conference on Information Technology: New Generations, ser. ITNG '14*, Washington, DC, USA, 2014.

[4] J. G. Andrews, T. Bai, M. N. Kulkarni, A. Alkhateeb, A. K. Gupta and R. W. Heath Jr., "Modeling and analyzing millimeter wave cellular systems," *CoRR,* vol. abs/1605.04283, 2016.

[5] N. Palizban, S. Szyszkowicz and H. Yanikomeroglu, "Automation of Millimeter Wave Network Planning for Outdoor Coverage in Dense Urban Areas Using Wall-Mounted Base Stations," *IEEE Wireless Communications Letters,* vol. 6, no. 2, pp. 206-209, 2017.

[6] S. S. Szyszkowicz, A. Lou and H. Yanikomeroglu, "Automated Placement of Individual Millimeter-Wave Wall-Mounted Base Stations for Line-of-Sight

Coverage of Outdoor Urban Areas," *IEEE Wireless Communications Letters,* vol. 5, no. 3, pp. 316-319, 16.

[7] "RF Line of Sight - SCADACore," [Online]. Available: http://www.scadacore.com/field-tools/rf-path/rf-line-of-sight/. [Accessed 29 April 2018].

[8] "airLink - Outdoor Wireless Link Calculator," [Online]. Available: https://airlink.ubnt.com/#/. [Accessed 29 April 2018].

[9] "HeyWhatsThat Path Profiler," [Online]. Available: http://www.heywhatsthat.com/profiler.html. [Accessed 29 April 2018].

[10] "Solwise - Surface Elevation Tool," [Online]. Available: http://www.scadacore.com/field-tools/rf-path/rf-line-of-sight/. [Accessed 29 April 2018].

[11] "Developer's Guide | Google Maps Elevation API," Google, Inc., [Online]. Available: https://developers.google.com/maps/documentation/elevation/intro. [Accessed 29 April 2018].

[12] C. A. Balanis, Antenna Theory: Analysis and Design, 3rd Edition, Wiley-Interscience, 2005.

[13] S. F. Adam, Microwave Theory and Applications, Adam Microwave Consulting, 1992.

[14] S. Gaikwad and D. C. Anastasiu, "Optimal Constrained Wireless Emergency Network Antenna Placement," in *Proceedings of the IEEE Smart City Innovations 2017 Conference,* San Francisco Bay Area, USA, 2017.

[15] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25-30, 1965.

[16] "Mapzen ? start where you are," 2017. [Online]. Available: http://mapzen.com/. [Accessed 01 August 2017].

[17] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org," 2018. [Online]. Available: https://www.openstreetmap.org.

[18] "liblas - las 1.0/1.1/1.2 asprs lidar data translation toolset," [Online]. Available: https://www. liblas.org/. [Accessed 18 December 2017].

[19] "PostGIS ? Spatial and Geographic Objects for PostgreSQL," [Online]. Available: http://postgis.net/. [Accessed 29 April 2018].

[20] "TNM Download," 2017. [Online]. Available: https://viewer.nationalmap.gov/basic/. [Accessed 11 April 2018].

[21] Y. Czoli, "Ykczoli/lidar-heights: Extract building heights using building footprints and lidar data with postgis," [Online]. Available: https://github.com/YKCzoli/ Lidar-heights. [Accessed 18 12 2017].

[22] "Wecome to QGIS," [Online]. Available: https://qgis.org/en/site/. [Accessed 29 April 2018].

[23] "FCC 6-7 GHz," Google, [Online]. Available: https://fusiontables.google.com/DataSource?docid=1-teDJHcmE21UMaaWnhCb2oy92eFD6m046UrbpGtQ#rows:id=1. [Accessed 29 April 2018].