

Evidential Obstacle learning in Millimeter wave D2D communication using Spatial Correlation

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

Master of Technology
in
Computer Science

by

Harish Ganesan

[Roll No: CS-1818]

under the guidance of

Dr. Sasthi Charan Ghosh

Associate Professor

Advanced Computing and Microelectronics Unit



Indian Statistical Institute
Kolkata-700108, India

July 2020

To my family and my guide

CERTIFICATE

This is to certify that the dissertation entitled “**Evidential Obstacle learning in Millimeter wave D2D communication using Spatial Correlation**” submitted by **Harish Ganesan** to Indian Statistical Institute, Kolkata, in partial fulfillment for the award of the degree of **Master of Technology in Computer Science** is a bonafide record of work carried out by him under my supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of this institute and, in my opinion, has reached the standard needed for submission.

Dr. Sasthi Charan Ghosh

Associate Professor,
Advanced Computing and Microelectronics Unit,
Indian Statistical Institute,
Kolkata-700108, INDIA.

Abstract

The 5G technology has created a lot of interest recently to support the increasing demand for higher data rates by the *user equipments* (UEs). The most interesting part of 5G is the use of *device to device* (D2D) communication, using millimeter waves. Millimetre waves can immensely increase the data rates compared to 4G. But millimetre waves suffer from a host of problems ranging from high free space loss restricting range, and also extremely high penetration loss, making it almost line of sight (LOS) communication. To handle these problems, communication links can be broken into multiple hops by using intermediate devices as relays, avoiding obstacles and thereby extending the range. Knowing the location and size of obstacles is key to choosing good relays. Satellite imagery can be used to do the same. But small obstacles like trees cannot be captured properly using satellite imagery. Moreover presence of an obstacle in the image, does not guarantee obstruction. Our work concentrates on mapping static obstacles in a given area to help in the process of relay selection. We propose a learning based strategy, which considers spatial correlation of obstacles, to build the static obstacle map efficiently without relying on satellite imagery. We propose a evidential framework to model the confidence in the knowledge gained for each cell, as this can model uncertainty better than a typical probabilistic model. We also propose an operation similar to Gaussian smoothing, to learn information about a cell, using the information available from the nearby cells. We have proposed a visibility graph algorithm which will reduce the overhead due to repeated updation of the location information by the UEs, and thereby improve the overall coverage dramatically. We have also proposed a *exploration* mode during relay selection, which can be used in the place of the *normal* mode of relay selection, which can reduce the time taken to learn the map, by selecting relays which encourage exploration rather than prioritizing maximizing throughput only. Through simulations, we show that our evidential framework based approach can learn the map faster and more accurately than the typical probabilistic model based work [18].

Keywords: *Learning, Obstacle avoidance, Evidential Framework, 5G D2D Communication, relay selection*

Contents

1	Introduction	6
1.1	Introduction to D2D Communication	6
1.2	Problems Associated with D2D Communication	7
1.3	Existing Solutions	7
1.4	Our Contribution	8
1.5	Thesis Outline	10
2	Related Work	12
2.1	Occupancy grid mapping	12
2.2	Spatial correlation in occupancy grids	12
2.3	Detection without range information	13
2.4	Evidential framework in map building	14
3	System model and Assumptions	16
3.1	Considered Architecture	17
3.2	Assumptions	18
3.2.1	Mapping Area	18
3.2.2	User Equipment Behaviour	19
3.2.3	Obstacle behaviour	19
4	Evidential Framework	20
4.1	Belief Functions	20
4.2	Combination Rules	21
4.2.1	Dempster’s Rule of Combination	21
4.2.2	Yager’s Rule of Combination	22
5	Learning based map building algorithm	24
5.1	Learning from communication	25
5.1.1	Forward Updation	26
5.1.2	Backward Updation	29
5.2	Learning from neighbourhood	31
5.2.1	Discrete Gaussian Smoothing	33
5.2.2	Smoothing based learning	34

5.2.3	Implementation Details	35
5.3	Building Visibility graph and Relay Selection Algorithm	36
5.3.1	Visibility Graph	37
5.3.2	Relay Selection	40
6	Simulation and Results	43
6.1	Simulation Setup	43
6.2	Results	45
7	Conclusion and Future Work	51

Chapter 1

Introduction

1.1 Introduction to D2D Communication

The increasing demand for higher data rates by the user equipments (UEs) has created great interest in 5G device to device (D2D) communications. In D2D communication, two proximity UEs can directly communicate with each other without involving the base station (BS). Generally D2D communication is studied for short distance communication which makes millimeter wave (mm wave) in the ranges between 30GHz to 300GHz as the suitable candidate for it. At this frequency, the data rate is of the order of multi gigabits per second and the corresponding wavelengths is in the range of millimetres. This extremely high frequency also means that the antennas can be further small, which can produce highly directional beams, and hence can pave the way for efficient D2D communication.

1.2 Problems Associated with D2D Communication

As the frequency of the transmitted wave increases, the signals suffer from high transmission and penetration loss than their lower frequency counterparts. Authors in [17] have found out that a 10cm thick concrete wall causes an attenuation of 175dB at 40 GHz. Hence, nearby UEs cooperate with each other and relay signals to establish a *line of sight* (LOS) path for a given source-destination pair. With the help of the relays, the communication links between a pair of UEs can be broken into multiple short hops, avoiding obstacles and thereby effectively lowering transmission and penetration losses. Hence detecting the presence of static as well as dynamic obstacles is of paramount importance to efficiently select an appropriate relay for a UE, lowering the chance of allocating an obstacle prone link.

1.3 Existing Solutions

One simple way of knowing the positions of the static obstacles is through satellite imagery. But the use of satellite imagery comes with two problems. Satellite information is proprietary, and it is costly to obtain detailed maps of all locations where the systems will be deployed. The other one is, smaller static obstacles such as lamp posts, banners, poles and trees, do not get registered in satellite imagery as easily, though they can potentially block mm wave signals. Learning based relay selection [18] provides an interesting framework by which static and dynamic obstacles can be learnt by using the history of mobile communications over the occupancy grid. In occupancy grid representation, the area in question is divided into a discrete cells of equal area, where the status of a cell at a particular time is either occupied by or

free from any obstacles. Obstacles may be learnt in a occupancy grid, by making use of the link failures/successes over a period of time. The above method of learning the occupancy of a discretized area of grids is known as occupancy grid mapping. Occupancy grid mapping is a well studied problem [7,21] in robotics. It is usually studied as a part of simultaneous localisation and mapping (SLAM) problem. In this report, we take a different look at the problem of static obstacle detection as a standalone map building problem.

1.4 Our Contribution

Learning based relay selection algorithms using occupancy grids may take large number of cycles to identify the static and dynamic obstacles. Most existing algorithms make a simplifying assumption that the occupancy of each cell of the occupancy grid is independent of the others. Whereas, in reality, obstacles in any area, is not independently distributed in a given area and are usually clustered in blocks. This is because, obstacles like buildings and many others usually span over multiple grids. Apart from this, existing algorithms do not model adequately the high uncertainty in each measurement. The presence of an obstacle in the initial few cells, would essentially mean that, we don't have much information about the remaining cells in the communication path. This would mean that instead of observing information about obstacles in all the cells that fall in the communication path, we would in reality be getting information about the first obstacle only, the number of obstacle blocks after that would not matter. This is because, a single obstacle along the communication path is sufficient to block the mm wave communication.

Using the spatial correlation of obstacles not only increases the accuracy of the map, but also can decrease the time taken for the map to converge. Many works

which deal with learning maps using spatial correlation use methods like Gaussian process (GP), belief propagation to learn from other labelled points after the traditional training step completed. The occupancy grid filter (OGF) in [20] made an improvement over GP by merging the training and prediction steps, thereby enabling continuous learning of the map using OGF as opposed to training GP with a large stream of input data, and then predicting the remaining cells in the map using GP. This way of updating other cells by simultaneous prediction and training is more suitable for our problem of relay selection, due to 2 reasons.

- In relay selection, the presence of an obstacle in the communication path is detected by measuring the difference between the expected *signal-to-interference-plus-noise ratio* (SINR) and the actual SINR and comparing the difference with a predefined threshold. However, the SINR measurements have a very high uncertainty due varying channel conditions. It will take a lot of iterations with these measurements for the confidence to build up to a level where GP can be used for prediction.
- Merging of training and prediction, can produce more complete occupancy maps while the process is running. This map can be used to choose relays in the meantime. Whereas, building maps solely with measurements produces a map with a very high uncertainty which will not be of much use until it finishes training.

In this report, we propose a evidential framework to model the confidence in the knowledge gained for each cell, as this can model uncertainty better than a typical probabilistic model. We also propose an operation similar to Gaussian smoothing, to learn information about a cell, using the information available from the nearby cells. Also, we have proposed a visibility graph algorithm which will reduce the

overhead due to repeated updation of the location information by the UEs, and also improve the overall coverage dramatically. We have also proposed a *exploration* mode during relay selection, which can be used in the place of the normal relay selection algorithm, which can reduce the time taken to learn the map, by selecting relays which encourage exploration rather than prioritizing maximizing throughput. This mode can be useful in cases, where a setting up time is available. Through simulations, we compare our work with the work in [18], and show that it learns the same map faster and more accurately using our proposed model, and we also show that due to the use of evidential framework, our map is more precise than the map prepared using the probability model in [18].

1.5 Thesis Outline

The rest of the report is organized as follows. In Chapter 2, we detail the related works in map building which incorporate spatial correlation in the process of learning. Chapter 3 deals with the system models and assumptions the proposed evidential framework based algorithm uses while arriving at the solution. In Chapter 4, we explain the basics of the evidential framework and about belief functions and some of the combination rules that have been used throughout the report. In Chapter 5, we present the details of our evidential framework based learning procedure. This section is divided into 3 parts. Section 5.1 details how basic learning of the map takes place. In Section 5.2, we will deal about how the algorithm relies on spatial correlation to better the learning speed and accuracy. Section 5.3 will deal with the visibility graph algorithm and the relay selection which works along with the learning procedure to assign relays by making use of the learnt map. Later in Chapter 6, the details about the simulation setup and the results are mentioned. Finally conclusions and future

works are presented in Chapter 7.

Chapter 2

Related Work

2.1 Occupancy grid mapping

There has been lot of work regarding occupancy grid mapping using sensors such as SONAR [7,21], LIDAR [5,6], Laser scanners [11], Global Navigation Satellite System (GNSS) and even mmwaves [23]. Most of the work on occupancy grids work with the assumption that each individual grid is independent from each other. Also, in [23], the authors assume that they can detect the time of arrival and time difference of arrival of the reflections of mmwaves, which will require extra hardware to achieve and will not be possible with a typical mobile equipment.

2.2 Spatial correlation in occupancy grids

To address the problem of independence, there has been many works trying to take advantage of the spatial correlation. The work in [14] uses Gaussian process regression to model the spatial correlation to predict cells' occupancies. Authors in [20] proposes an improvement over Gaussian process regression in terms of computational complex-

ity, by using Kalman filter based OGF. Mark A. Paskin and S. Thrun [16] proposed a geometric representation of the occupancy using polygonal random fields. Belief propagation on high order factor graphs was used in [6] to propagate the learnt information from a cell to its neighbours. However, all of the above works are proposed for mapping using sensors which provide a good probability distribution through models such as forward or inverse sensor models, which they take advantage of in their works.

2.3 Detection without range information

The problem at hand deals with a sensor, which can only sense whether a particular LOS is blocked or not. Range information is not available to us, since we are assuming that each device in this process is a typical mobile device, which generally does not have sensors like radar or ultrasonic sensors which can detect range of obstacles. Mapping using the difference between the expected SINR and the actual SINR also has been explored, using Global Positioning System (GPS) in [10] or Global Navigation Satellite System (GNSS) in [8, 22]. In these works the SINR of the signal from a set of satellites, has been used to build 3D occupancy maps of buildings and other obstacles which block the rays. The advantage that these works have are, all the obstacles are clustered on one side of the communication path i.e. in the communication path from the satellite to the person on the ground, it is highly unlikely that an obstacle will be present anywhere in the middle, apart from some grids near the ground level. Unlike these methods, in our problem an obstacle may lie anywhere in the communication path. Also these works make use of data from multiple satellite sources simultaneously, using which we can identify some of these grids as empty, by using the intersecting information of these multiple satellites, whereas each communication path is examined only using one beam at a time in our problem.

2.4 Evidential framework in map building

Evidential approaches to map building have been explored by multiple authors. In [15], a sensor model has been developed using Dempster-Shafer theory, for a SONAR sensor beam, for handling its bigger arc of uncertainty in SONAR based detection. In [5], a LIDAR based mapping is done using Dempster-Shafer evidence theory. They also make use of a modified updation rule called PCR6 rule, to suit their needs better. They also make use of discounting to incorporate unreliable evidences into their learning process. Evidential approach have been used for vehicle trajectory planning in [13] to capture the uncertainty in environment perception. However, almost all of the works which uses evidential frameworks deal with sensors which have well-defined sensor models connecting the outputs of these sensors with range information.

In our work, we take the evidential approach to map building and apply it for the problem of relay selection. The map is constructed from data obtained from the history of communications, which doesn't have any well defined sensor model unlike all the other works which takes the evidential approach to occupancy grid map construction. Also we propose a method similar to Gaussian Smoothing, for taking advantage of the spatial correlation of the distribution of obstacles in the area, which improves the learning rate and also the quality of the map. We have also proposed a visibility graph algorithm for relay selection which reduces the communication overhead and the computational overhead due to the collection of location information at every time-step. Due to the availability of uncertainty information of each cell in the grid, relay selection algorithm can also be improved by specifically suggesting relays, which can help reduce the map uncertainty faster than the existing relay selection algorithm. We have proposed this as a separate mode of relay selection called *exploration mode*,

which can be used in situations where link quality is not of priority.

Chapter 3

System model and Assumptions

Millimeter wave communication can provide considerable improvement in the data rate only if the distance between the sender and the receiver is within a threshold, and they should have a LOS transmission path. If the distance is high, the data rate reduces considerably due to the high free space loss of mm waves.

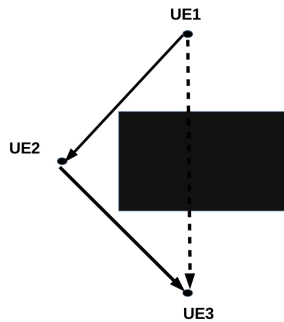


Figure 3.1: Example of a relaying

In Figure 3.1, UE1 wants to communicate with UE3, but since it is obstructed by a building, it uses UE2 as a relay, since UE1 has an LOS with UE2 which in turn has an LOS with UE3.

3.1 Considered Architecture

In our work, we are considering a similar architecture to the ones mentioned in [9,18]. There is a central base stations (BS) and 6 other mm wave BS (MMW BS) on the sides of a regular hexagon with the central long term evolution base station (LTE BS) at the centre. The MMW BSs are connected to the LTE BS through a high speed back-haul network. The UEs are spread across the area uniformly.

Each MMW BS can communicate with other base stations directly through the back-haul network. It can also communicate with other UEs if there is an short distance LOS path from the UE to the MMW BS. It can also try connecting by the use of other UEs if they are willing to act as relays. Similarly, UEs also can communicate with other UEs either directly or using other UEs as relays to ensure it at least has a multi-hop LOS path.

The UE, when trying to communicate with other UEs can take *suggestions* from the LTE BS, regarding the best path/relay to choose. Since these are all comparatively small control messages, even low data rate LTE connections can be used to exchange these messages. Once an UE gets the required information, it tries to establish to the next link in the multi-hop path (i.e., the next link it has been suggested) to reach the desired UE. Here we assume that, at a time any UE can accept at a maximum one incoming and outgoing link. If the relay is established successfully, the packets are sent through this relay. If the suggestions of the LTE BS fails, the UE has to use the LTE BS for communication with a lower data rate.

The figure 3.2 shows one such example of the system model explained above. In the center of the image, there is the LTE BS, and there are 6 other mm wave BS arranged in a hexagon. The sparsely dashed line indicates the control signals that passes between the UEs and the LTE BS, the densely dashed line represents the

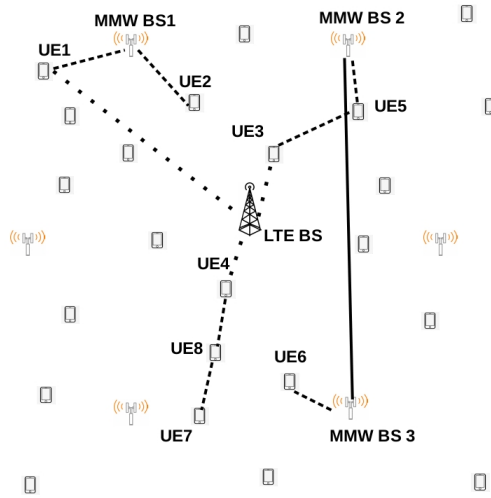


Figure 3.2: Typical system state

actual mm wave D2D communications, and the continuous lines represent the backhaul network transmission. The image shows three types of connections possible in the network. UE1 tries to connect UE2, and it uses MMW BS 1, as a relay. UE3 wants to communicate with UE6, since it doesn't have a direct path or a 1 hop path, it is instructed to reach MMW BS 2, through a one hop pass. UE3 reaches MMW BS 2 through UE5. MMW BS 2 connects to MMW BS 3 through the backhaul network, which in turn communicates to UE6. In the other example, UE4 manages to establish a direct 1 hop path to UE7 without the help of any mm-wave BS.

3.2 Assumptions

3.2.1 Mapping Area

We divide up the area under the LTE BS into square grids of a reasonable size. The smallest possible size of square grids are bounded by the fact that GPS has a finite resolution, below which our method fails to differentiate between separate grids.

Some GPS, provide an accuracy of 30cm. Such level of detail is usually not necessary. Decreasing the size of the grids into finer grids, increases the complexity quadratically, but also increases the accuracy with which the map has been learnt. On the other hand, coarse grids suffer from the lack of reliability in the learnt information, as a large grid can have an obstacle in some place and have free paths. Here we assume that the grid size is 1m x 1m.

3.2.2 User Equipment Behaviour

Similar to [18], we consider only UEs which are used by Pedestrians. Similarly UEs movement is assumed to be *pseudo-stationary*, i.e., UEs are assumed to be occupy the same cell it occupies for a short interval of time. This is not an unreasonable assumption, due to the fact that the speed of a pedestrian and the grid size, allows us to choose a quantum of time, for which this assumption is true.

3.2.3 Obstacle behaviour

We consider two kinds of obstacles, static and dynamic. In our work, both kinds of obstacles are present, but unlike [18], we only model the static obstacles. Dynamic obstacles are included in the work, to show that the algorithm is resilient even in presence of dynamic obstacles.

Static Obstacles: The static obstacles usually involve big building, trees, bridges, towers, signboards etc. We assume all these obstacles block the mmwaves enough to reduce the SINR below the acceptable threshold.

Dynamic Obstacles: Dynamic obstacles constitute humans and vehicles. These obstacles still hamper the transmission. But the obstacles can be expected to be not present in the same cell in subsequent time epochs.

Chapter 4

Evidential Framework

Dempster-Shafer theory (DST) was born from the works of Dempster in 1967 [4] and developed by Shafer in 1976 [19]. DST is also known as the theory of belief functions. It deals with frame of discernment (FoD), basic belief assignments (BBA), belief (Bel), plausibility (Pl) and combination rules. DST is considered as the counterpart of Bayesian updation for belief functions. Evidential framework allows us to work with uncertain, unreliable measurements, while also maintaining the complexity in manageable levels.

4.1 Belief Functions

Let us consider w be a unknown quantity from a possible set of values Ω , also called frame of discernment (FoD). The pieces of evidence are represented as a function $f : 2^\Omega \rightarrow [0, 1]$, such that $m(\phi) = 0$ and $\sum_{A \subseteq \Omega} m(A) = 1$.

In the occupancy grid problem, $2^\Omega = \{\phi, F, O, F \cup O = \Omega\}$, where F represents free, O represents occupied and Ω denotes both of them, which essentially means unknown. A mass function known as *basic belief assignment* (BBA) is calculated.

The BBA is a function which maps every possible subset of Ω to a mass value similar to probability, such that the sum of all these masses add up to 1. The mass of a particular subset denotes the belief in the particular subset. Mass assigned to Ω denotes the uncertainty or unknown. Conflict ϕ is explicitly avoided in some rules like Dempster's rule of combination, whereas in Smets combination rule and in Yager's combination rule [12], they are either kept as is, or is added to the universal set Ω .

Belief functions can be used to model the level of certainty we have in each cell of the occupancy grid. The communications which are used for training the map, provide evidences for occupancy of each cell. These evidences can be combined using the below mentioned combination rules, with the existing belief of these cells, to gradually eradicate the uncertainty from the occupancy map.

4.2 Combination Rules

Combination rules are rules using which two separate pieces of information are combined to produce the final BBA. There are multiple noteworthy rules, which are useful. Let the two information sources be I_1 and I_2 , and the resultant belief after the combination be R .

4.2.1 Dempster's Rule of Combination

This is the widely used combination rule in the evidential framework. This rule considers both the pieces of information as independent to each other. In the case of occupancy grid problem, the set is $2^\Omega = \{\phi, F, O, F \cup O = \Omega\}$.

For any $A \in 2^\Omega$, The combination rule is defined as,

$$m_R(A) = \frac{m_\cap(A)}{1 - m_\cap(\phi)} \quad (4.1)$$

where, $m_R(A)$ corresponds to the resultant belief after combination and m_\cap corresponds to the conjunction rule of combination defined as,

$$m_\cap(A) = \sum_{\substack{B \cap C = A \\ B, C \in 2^\Omega}} m_{I_1}(B).m_{I_2}(C) \quad (4.2)$$

$$m_\cap(\phi) = \sum_{\substack{B \cap C = \phi \\ B, C \in 2^\Omega}} m_{I_1}(B).m_{I_2}(C) \quad (4.3)$$

Here, $m_{I_1}(A)$ and $m_{I_2}(A)$ corresponds to the input beliefs before combination. Here, it can be observed that the conflict masses are removed, and the other masses are normalized to sum up to 1.

4.2.2 Yager's Rule of Combination

One of the main assumptions of Dempster's rule of combination is that the evidences are completely reliable. So in that case, a conflict only occurs in the case of corrupt evidence. So in that case, the corrupt evidence shouldn't be part of the belief at all. So it was removed.

But in cases of non-reliable evidence sources, conflict can also arise due to the inability of the sources to correctly identify the evidences. In this case, Yager proposes to modify the combination rule to include the conflicting mass $m(\phi)$ to the whole set $m(\Omega)$, which implies that the conflict occurs due to the uncertainty of different evidences, rather than the corruption of any evidence source.

For any $A \subseteq \Omega$, The combination rule is defined as,

$$m_R(A) = \sum_{\substack{B \cap C = A \\ B, C \subseteq \Omega}} m_{I_1}(B)m_{I_2}(C) \quad (4.4)$$

$$m_R(\Omega) = m_{I_1}(\Omega)m_{I_2}(\Omega) + \sum_{\substack{B \cap C = \phi \\ B, C \subseteq \Omega}} m_{I_1}(B)m_{I_2}(C) \quad (4.5)$$

where, $m_R(A)$ denotes the resultant combined belief of A, whereas $m_{I_1}(A)$ and $m_{I_2}(A)$ are the input beliefs of A, and Ω denotes the universal set.

Yager's rule of combination is used through the work, because the reliability of evidence is very low in our case.

Chapter 5

Learning based map building algorithm

Using the above architecture and assumptions, an occupancy map can be learnt by the use of continuous updation based on the connections established and broken. After learning a good amount of information, this information can be used to form good visibility graph, as mentioned in [18], which can be expected to help the relay selection algorithm in choosing good relays and links.

The proposed method to learn the occupancies of the grids consists of two parts. The first part of the algorithm is to update the beliefs of the grids by assigning links to UEs and BSs and detecting the SINR difference, to guess whether the links are laden with obstacles or not. The second part of the algorithm updates the beliefs of **student** cells by learning from the surrounding **expert** cells. Expert cells are cells which have a high mass for either *free* or *occupied*, i.e., above a particular predefined threshold t_h . Student cells on the other hand, are cells which have both their beliefs below a specific threshold t_l . The second part of the algorithm takes advantage of the fact that obstacles are not distributed uniformly and independently across the grid,

but distributed as clusters such as buildings, bridges etc.

Below we explain the process to build the map for UE-UE communication. The UE-BS communication also requires a similar process.

5.1 Learning from communication

We propose to improve the algorithm proposed by [18] to learn individual occupancies of each cell in the grid. In the above algorithm, the authors propose that every time a link gets obstructed, i.e., the measured SINR is very less than the expected SINR, the cells in the straight line from the sender to the receiver, are assigned an equal probability for the presence of the obstacle. Then by using the prior for the cells, the probabilities are adjusted.

There are some problems with this kind of updation

- There can be more than 1 obstacle in the straight line from I to J. So considering the cells equiprobable to contain the obstacle is not accurate. Also the updation probabilities of each cell is independent of the other, as each cell here is a separate distribution. So it need not necessarily add up to 1.
- Now considering that there is a possibility of more than 1 obstacle, the first obstacle the wave hits, will mostly be enough to annihilate the SINR. So the presence of the 2nd and the other obstacles make no difference to our outcome of the experiment.

For example: Consider that we have a high probability of an obstacle in a particular cell in the beginning of the straight line from the sender to the receiver.

Let the grid below be the straight line from the sender to the receiver. The numbers in each cell indicate the probability of finding an obstacle in that cell.

0.3	0.8	0.2	0.5	0.2	0	0	0
-----	-----	-----	-----	-----	---	---	---

If we are fairly certain that the obstacle exists in the 2nd cell which has a high probability of an obstacle, then the experiment we conducted doesn't say anything about the cells after the cell. So assigning an equal probability to those cells will be misleading. Our experiment, will have lesser and lesser idea of the status of the cells after the 2nd cell, since the obstacle probability is higher in the 2nd cell. No matter what lies or doesn't lie after the 2nd cell, the experiment's outcome will not change. So the updation probability/beliefs should be lesser and lesser as we move forward in such situations.

By using the evidential framework, we are well placed to learn in the midst of highly uncertain evidence. The algorithm also has two sub parts, explained below. The first part, we can call as the forward updation, will provide an evidential way to update the cells moving forward from the sender's cell moving towards the receiver's cell. In the second part, the backward updation, will try to influence the forward updation's updation probability, if the cells after the particular cell is mostly empty.

5.1.1 Forward Updation

Let us consider 2 hyper parameters for now. Let O_s be the density of static obstacles in the given area, and O_d be the rate of dynamic obstacles in the given area.

We use an algorithm called Bresenham's algorithm [2], which is a well known algorithm in Computer Graphics to draw straight lines avoiding floating point operations in computer screens. Using Bresenham's algorithm, let us consider that we have obtained the sequence of cells, which come in between the sender cell (I) and the receiver cell (J). Each cell has a corresponding tuple attached with it. Let it be

(C^o, C^f, C^d) , where C^o is the mass of Presence of obstacle of cell C, C^f is the mass of absence of obstacle of cell C, C^d is the mass of , or otherwise “Don’t know”. Let the cells in the straight line from I to J, be $C_1, C_2, .., C_n$

Case 1 : Experiment returns no obstacles: In this case, we now clear all these cells of any possibility of having a static obstacle. So we now update this with triples $(0,1,0)$. Using the update rule (Dempster’s Combination rule), if we update for all the cells $C_1, C_2, C_3, ..C_n$, all the cells will then have tuples $(0,1,0), (0,1,0)...$, signifying that we are sure that it doesn’t have any static obstacle. We can do this, because even if we have one obstacle-less communication, it is sure that there are no static obstacles, otherwise we wouldn’t have obtained an LOS even once.

Case 2: Experiment returns an obstacle: In this case, our experiment returned an SINR which is very less than the expected SINR. We have to consider all the possibilities of an obstacle. The drop in SINR can be due to a dynamic obstacle or there could have been a static obstacle somewhere in the straight line.

Let us consider the probability with which ray reaches a cell C_r be P_r . We can calculate P_r by using the below mentioned recursive formula.

$$P_r = P_{r-1}((1 - O_s)C_{r-1(oid)}^d + (1 - O_d)C_{r-1(oid)}^f) \quad (5.1)$$

$$P_1 = 1$$

The equation 5.1 is a heuristic formula. As we argued in Section 5.1, we can learn information about a cell when and only when the mm wave reaches the particular cell. So, when the probability of reaching the cell is low, the evidence we get for saying the cell is occupied, also reduces. For the same reason, we calculate the probability P_r with which the mm wave reaches cell C_r without significant drop in SINR. We consider that with a probability $O_s C_r^d + C_r^o + O_d C_r^f$, where O_s is the hyper parameter indicating

the approximate density of obstacles, and O_d is the rate of dynamic obstacles ($O_d = 1$ indicates there is always a dynamic obstacle, in any free space, $O_d = 0$ indicates there is no dynamic obstacles). Instead of relying on the hyper parameter O_d , one can instead rely on a dynamic obstacle metric, which depends upon the actual probability of dynamic obstacles in the particular area or with an estimated probability of a dynamic obstacle using a method like dynamic obstacle learning method specified in [18]. But that is beyond the scope of this report.

Now using eqn. 5.1 we devise a BBA for the current experiment. Let the belief for the current experiment for cell r be $B_r = (B_r^y, B_r^n, B_r^d)$,

$$B_r^o = P_r O_s, B_r^f = 0 \quad (5.2)$$

$$B_r^d = 1 - B_r^o \quad (5.3)$$

To obtain the posterior, this BBA can be combined with the prior using the Yager's Combination rule. Yager's rule has been chosen here because the uncertainty of the experiment is still very high, so conflicting information can be taken as a source of uncertainty. Since we are using Yager's rule, the cells which have prior beliefs (0,1,0) are corner cases here. Instead of updating these cells, these are left as is, as the updation using the new BBAs can make the cell's belief uncertain again, but in our experiments, we argued in Case 1 in 5.1.1 that (0,1,0) is the only certain state that we can have, since we can clear any cell of an obstacle if it ever has a successful communication. Due to this, only P_r is calculated for these cells and the updation steps are skipped.

$$C_{r(new)}^o = C_{r(old)}^o(B_r^o + B_r^d) + B_r^o C_{r(old)}^d \quad (5.4)$$

$$C_{r(new)}^f = C_{r(old)}^f(B_r^f + B_r^d) + B_r^f C_{r(old)}^d \quad (5.5)$$

$$C_{r(new)}^d = C_{r(old)}^f B_r^o + C_{r(old)}^o B_r^f + B_r^d C_{r(old)}^d \quad (5.6)$$

5.1.2 Backward Updation

In forward updation, we assumed O_s to be a hyperparameter constant. But consider a case where we already know most of the cells after a particular cell C_r , in the straight line from I to J, is empty. In this case, we can deduce that there is an increased chance/evidence of an obstacle being here. Ideally, in such a situation, we can assign a high belief for the cell being Occupied.

Due to this, let us consider $O_{s,r}$ to take values equal to a function of another hyperparameter θ , which now denotes the approximate density of obstacles. Whenever the cells after C_r are not known to be mostly empty, this function will return a value equal to θ . But when those cells are known to be mostly empty, it should take a value almost equal to 1.

Let us consider an skewed/weighted occupancy metric, A_r for the cell r.

$$A_r = \frac{1}{n-r} \sum_{i=r+1}^n (\gamma(C_r^f) + (1+\gamma)(C_r^o + C_r^d)) \quad (5.7)$$

γ is a hyperparameter which takes a positive value near to 0. This is to ensure that A_r doesn't actually reach 0 in any case. The backward updation formula we will propose will have a value equal to 1, when A_r is exactly 0. Whereas in reality we will never be certain about the presence of a static obstacle, even if all the cells after the cell in discussion, is empty, Since there is a finite possibility of a dynamic obstacle.

So γ shifts the left extreme of the axis to some small positive value above 0. Also $(1 + \gamma)$ factor is used to skew the range above 1, to ensure that in cases where there is a small uncertainty about some cells, but all remaining cells are known to be free, the average is not brought down by the majority free cells. i.e. The mere presence of a free cell is less significant in this decision than the presence of an obstructed or an uncertain cell.

Now using this we can introduce a new parameter in place of O_s , by the use of θ , as a weaker hyperparameter.

$$O_s = \begin{cases} \theta + (1 - \theta) \frac{\log(A_r + \epsilon)}{\log \epsilon} & A_r \leq 1 \\ \theta & A_r > 1 \end{cases} \quad (5.8)$$

Here ϵ is yet another hyperparameter, which will determine how fast the curve falls from 1 to θ . The figure 5.1 below showcases the effect of different ϵ .

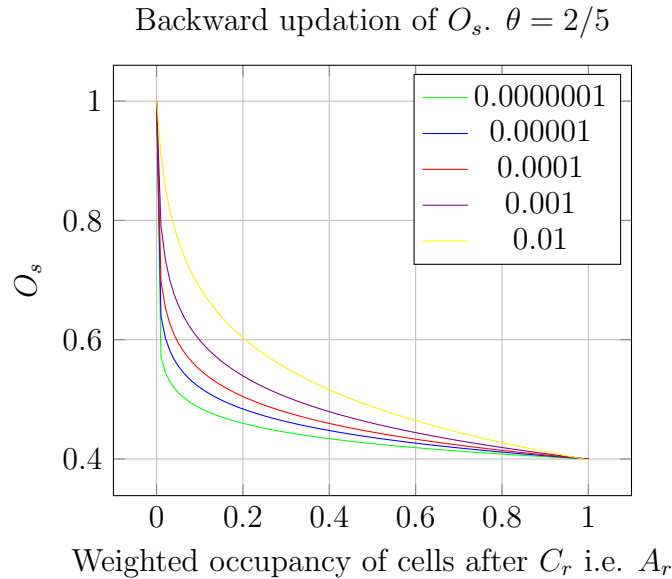


Figure 5.1: O_s for various values of ϵ

This parameter O_s can now be used in (5.1) and (5.2) instead of the constant

hyper parameter.

Algorithm 1, details the implementation of the learning by communication algorithm. G is the entire grid, which contains the triples for all the cells in the entire map. p_1 and p_2 , are the coordinates of I and J . C is an auxiliary array used here to store all the individual elements in the straight line between I and J . Each element of G and C , is a structure which in turn contains three elements o, f and d as shown below. Lines 7 to 10, is the part where updation for cells which are found to be empty happens. Lines 11 to 29, is where the updation for lines which are blocked happens. Lines 13 - 16 is the backward updation. It should be clear now, why it can be called backward updation, since the computation starts from the last cell. Lines 17 - 29 corresponds to the forward updation, where the main learning takes place. A_r from backward updation, is used in Line 18 to obtain O_s . In line 27, the call for learning from surroundings arises, which will be detailed later. After all the learning, we will finally update the cells in the grid.

```
struct belief_mass{
    float o,f,d;
};
```

5.2 Learning from neighbourhood

We can learn a sparse map from the algorithm detailed in Section 5.1. But as we explained in Section 1.4, we also propose to improve the learning process by considering the spatial correlation of the obstacles. This property helps us to predict the occupancy of cells ,whose neighbour cells have high belief. Here we propose an algorithm similar to Gaussian smoothing, to learn from neighbours.

Algorithm 1: Learning through Communication

Data: $p_1, p_2, \delta, G, \theta, \epsilon, O_d, \gamma$

Result: G

```
1  $E\_SINR_{p_1,p_2} \leftarrow$  Estimated SINR between  $p_1$  and  $p_2$  assuming LOS using
   Frii's free space equation ;
2  $M\_SINR_{p_1,p_2} \leftarrow$  Measured SINR between  $p_1$  and  $p_2$  ;
3  $B_{p_1,p_2} \leftarrow$  Set of cells lying between  $p_1$  and  $p_2$  using Bresenham's algorithm. ;
4  $P \leftarrow 1$  ;
5  $n \leftarrow len(B_{p_1,p_2})$  /* Length of  $B_{p_1,p_2}$  */;
6  $C \leftarrow$  Belief triplets for cells corresponding to  $B_{p_1,p_2}$  from G /* C is an
   array of a structure with three members, C.o,C.f,C.d */ ;
7 if  $|M\_SINR - E\_SINR| < \delta$  then /* LOS established */
8 |   for  $r \leftarrow 1$  to  $n$  do
9 | |    $C[r] \leftarrow (0, 1, 0)$ 
10 |   end
11 else /* LOS blocked */
12 |    $A \leftarrow$  Array of floats of size  $n-1$ ;
13 |   for  $r \leftarrow n - 1$  to  $1$  do
14 | |    $R\_Sum \leftarrow (1 + \gamma) * (C[r + 1].o + C[r + 1].d) + \gamma * C[r + 1].f$  ;
15 | |    $A[r] \leftarrow \frac{R\_Sum}{n-r}$  ;
16 |   end
17 |   for  $r \leftarrow 1$  to  $n$  do
18 | |   if  $A[r] \leq 1$  then
19 | | |    $O_s \leftarrow \theta + (1 - \theta) * \frac{\log(A[r]+\epsilon)}{\log(\epsilon)}$ ;
20 | | |   else
21 | | |    $O_s \leftarrow \theta$ 
22 | | |   end
23 | |    $P_{new} \leftarrow P * ((1 - O_s) * C[r].d + (1 - O_d) * C[r].f)$ ;
24 | |    $B \leftarrow (P * O_s, 0, 1 - P * O_s)$ ;
25 | |   if  $C[r] \neq (0, 1, 0)$  then
26 | | |    $C[r] \leftarrow YagersCombination(C[r], B)$  ;
27 | | |   end
28 | |    $LearnFromSurroundings(C, B_{p_1,p_2})$  ;
29 | |    $P \leftarrow P_{new}$ 
30 |   end
31 end
32 Update G using updated values from C ;
```

Our claim is taking the spatial correlation into account, makes the learning phase shorter and also since the algorithm relies entirely on random connections, it also helps to learn a better representation of the map within a short period of time and possibly with fewer data points.

5.2.1 Discrete Gaussian Smoothing

Discrete Gaussian Smoothing [3] is a well known operator used in Computer Vision to smooth/blur and to remove noise from an input image.

A 2-D isotropic Gaussian has the form :

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{\sigma^2}} \quad (5.9)$$

A discrete approximation of a Gaussian function is used as the Gaussian filter, and is usually convolved around an image in order to smooth the image or remove noise in the image. The filter in the Figure 5.2 is a discrete approximation of a Gaussian distribution with $\sigma = 1$. The distribution is truncated after 5 cells in either direction, and the entire filter is finally normalized to add up to 1.

	1	4	7	4	1
	4	16	26	16	4
$\frac{1}{273}$	7	26	41	26	7
	4	16	26	16	4
	1	4	7	4	1

Figure 5.2: Discrete gaussian filter of size 5 with $\sigma = 1$

5.2.2 Smoothing based learning

Let us name the cells ,which have high mass i.e. more than a threshold t_h , for either *Free* or *Occupied* state as **experts**. Due to the spatial correlation of the obstacles in the space, each expert gives us some clue into how the surrounding cells will look like.

For cells, which have masses for both *Free* and *Occupied* less than less than a particular threshold t_l , we learn from the neighbouring experts using Smoothing based learning. Let us call them **student** cells.

Let us consider a fixed neighbourhood size of l , throughout the learning phase. The size of the filter for a neighbourhood size l will be $(2l + 1) \times (2l + 1)$.

Let us consider a variance function $\sigma(t)$, where t is the number of the iteration. The purpose of this variance function is to enable the Gaussian filter to shrink over-time, as to make it learn from lesser area, as we learn more about the environment.

$$\sigma(t) = \sigma_0 e^{-\alpha t} \quad (5.10)$$

$\alpha(t)$ can now be used in the place of α for generating the Gaussian distribution.

Let us consider the coordinates of the *expert* cell using which smoothing is going to be applied in its neighbourhood, be x_0 and y_0 .

All the *student* cells in the neighbourhood of (x_0, y_0) , i.e. $x_0 - l \leq x \leq x_0 + l, y_0 - l \leq y \leq y_0 + l$ are all chosen, and let the set be F . If the cell in (x_0, y_0) is a **Free expert**, then for a cell $(x, y) \in F$ in its neighbourhood, the new BBA will be

$$B_{(x,y)}^o = 0, B_{(x,y)}^f = G(x - x_0, y - y_0) \quad (5.11)$$

$$B_{(x,y)}^d = 1 - B_{(x,y)}^f$$

Similarly, if (x_0, y_0) is a **Occupied expert**, then for a cell $(x, y) \in F$ in its neighbourhood, the new BBA will be

$$B_{(x,y)}^o = G(x - x_0, y - y_0), B_{(x,y)}^f = 0 \quad (5.12)$$

$$B_{(x,y)}^d = 1 - B_{(x,y)}^o$$

For all cells in F, Yager's combination rule is used to update the belief assignment.

5.2.3 Implementation Details

The environment even though is correlated to a degree, is not usually very smooth. This kind of smoothing should be applied carefully, as it can make the resulting map too smooth. So everytime, a cell is updated through communication, the updated expert cells, can be used to learn information about their surroundings.

The Algorithm 2 shows how a particular cells in the map, whose belief is less than a threshold, can learn from its *expert* neighbours. Here (x_0, y_0) is the coordinate of the expert cell, whose expertise we are about to spread to its *student* neighbours. T is the iteration number. σ_0 is the initial value of standard deviation. α_t is the time constant, which will determine how fast the standard deviation falls off. G is the occupancy grid that is currently being learnt by the algorithm. l is the neighbourhood size of the filter. t_h and t_l are limits, using which we can identify *student* and *expert* cells. From lines 4 - 17, we check whether the particular cell in question is an expert, on which case, we try to spread its expertise to its neighbours. If the cell is an occupied cell, it increases the belief for *occupied* class for cells in its neighbourhood, defined by l . If it is unoccupied, it does the same but for the other class. For each *expert* cell, this is a $O(1)$ operation since, filter sizes remain a constant throughout. For the each

communication though, it is of the order $O(E)$, where E is the number of expert cells in the straight line after the initial updation.

Algorithm 2: Learning through Smoothing

Data: $x_0, y_0, T, G, \sigma_0, \alpha_t, l, t_h, t_l$
Result: G

```

1  $\sigma_T = \sigma_0 e^{-\alpha_t T}$  ;
2 Gauss  $\leftarrow$  Gaussian filter of size  $(2l + 1) \times (2l + 1)$  with  $\sigma = \sigma_T$  and
   normalized to 1;
3  $S \leftarrow$  Coordinates of students in the range
    $x_0 - l \leq x \leq x_0 + l, y_0 - l \leq y \leq y_0 + l$  ;
4 if  $G_{x_0, y_0}^o \geq t_h$  OR  $G_{x_0, y_0}^f \geq t_h$  then
5   for  $(x_s, y_s) \in S$  do
6      $B_{(x_s, y_s)} \leftarrow (0, 0, 1)$  ;
7     if  $G_{x_0, y_0}^o \geq t_h$  then ;                               /* Occupied expert */
8
9     |    $B_{(x_s, y_s)}^o = Gauss(x_e - x_0, y_e - y_0)$  ;
10    |    $B_{(x_s, y_s)}^d = 1 - B_{(x_s, y_s)}^o$  ;
11    else ;                               /* Free expert */
12
13    |    $B_{(x_s, y_s)}^f = Gauss(x_e - x_0, y_e - y_0)$  ;
14    |    $B_{(x_s, y_s)}^d = 1 - B_{(x_s, y_s)}^f$  ;
15    end
16     $G_{(x_s, y_s)} \leftarrow Yagers\_Combination( G_{(x_s, y_s)}, B_{(x_s, y_s)} )$  ;
17  end
18 end

```

5.3 Building Visibility graph and Relay Selection

Algorithm

We are also suggesting improvements to the visibility graph and the relay selection algorithm proposed in [18].

5.3.1 Visibility Graph

In [18], the visibility graph involved all the mm wave BS's to build their own visibility graphs with only the nodes inside the coverage area of kD , where k is the number of hops allowed and D is the distance beyond which SINR drops too low to get any significant advantage in data rate. Also at every time step, the visibility graph gets reconstructed based on their updated locations.

This has two potential problems.

- Expecting the devices to update the location at every time step, especially when the time step considered here is 0.25s is unrealistic. Also when the number of UEs in the zone gets high, the overhead induced by an updation like this, will become unmanageable very quickly.
- When each mm wave BS construct their own visibility graphs, with a horizon of kD , a lot of potential D2D links will be missed which cross the horizon of the mm wave BS's coverage. This problem will only worsen when the available UEs within the zone increases. So some sort of global visibility graph has to be maintained within the entire zone to ensure those links aren't missed. But with the current setup, constructing the global visibility graph will impose a heavy computational load on the BS, especially considering the time limit of a time-step being as small as 0.25s

Due to the above mentioned problems, we propose that the visibility graph to be constructed in the LTE BS instead of the mm wave BS, as the coverage of any mm wave BS is limited when compared to the LTE BS. Also since the LTE BS can reach any device within the zone, it constructs a global visibility graph for that zone.

Addressing the problem of increasing computational complexity, we propose that instead of constructing the visibility graph at the start of every iteration from scratch,

the changes in locations can be received as updates from the UEs, which will then be updated in the visibility graph. Changes in the location of an UE will mean that it will have new neighbours. To find out the neighbour a KD-Tree [1] will be maintained alongside the visibility graph which can efficiently compute the neighbours of any other node in $O(\sqrt{n} + k)$ time, where k is the number of neighbours of a cell, and n being the total nodes in the KD-Tree. Also calculating and adding the new node in the visibility graph will be $O(k)$ using a visibility graph similar to the one proposed in [18]. The deletion of a node in the KD-Tree takes another $O(\sqrt{n})$ time.

One drawback of this method is, since we are maintaining the visibility graph without reconstructing every iteration, the weights of the visibility graph which will encode the information learnt in the visibility map, will remain outdated. This should not be a problem in a place where the UEs keep moving fast. But in such a situation, the overhead by changing the nodes will overtake the overhead of constructing the visibility graph every iteration. In a zone, where the UEs don't move fast, the weights of the visibility graph also does not get adapted fast.

So to overcome this problem, the entire graph can be reconstructed once every n iterations once, to ensure that the graph the algorithm is not working with very outdated information.

The visibility graph's weights are decided by the use of the visibility map constructed through learning in Section 5.1. Consider two nodes in the graph corresponding to individual devices at positions p_1 and p_2 , the weight of the edge connecting these two nodes is calculated by multiplying the *Free* and *Uncertain* belief masses of all the intermediate cells between p_1 and p_2 . *Uncertain* belief mass is also taken into account to encourage the algorithm to suggest uncertain links for exploration implicitly.

$$w(p_1, p_2) = \prod_{b \in B_{p_1, p_2}} (1 - C_b^o) \quad (5.13)$$

In the case where, a separate pre-processing time can be allocated, for example if it is possible that while setting up such a network, a group of dummy UEs are allowed to roam around the area, then an alternate type of visibility graph can be used for such a phase, which we can call **exploratory phase**. During the exploratory phase, the aim of the experiment is to learn as much as possible about the map/zone. So since the communication quality is not of importance here, we can slightly tweak the visibility graph weight allocation procedure to prefer uncertain links rather than best LOS links.

In the exploratory phase, instead of the above mentioned weight initialization formula, the below formula can be used.

$$w(p_1, p_2) = \sum_{b \in B_{p_1, p_2}} C_b^d \quad (5.14)$$

By using the uncertainty alone during the link creation, while relay selection, links with the highest cumulative uncertainty can be preferred over well known performing links. This will aid the faster learning of the graph and reduce the number of iterations required for convergence of the graph.

Algorithm 3, details how the individual functions of the visibility graph should work. The function *CreateVGraph* creates a visibility graph from the set of nodes initially. *EditVGraph* is the function which will be called when the UE updates its location change to the LTE BS. This function updates the Visibility Graph in such a way that the information related to the updated UE is up to date. *AddToVGraph* is a helper function, whereas *DeleteFromGraph* is called when the UE moves updates

the BS that it is going to disappear.

5.3.2 Relay Selection

Using the visibility graph, if a request for data transmission from p_1 to p_2 arrives, the distance between them are checked. If they are within the same mm wave BS's coverage and within kT distance, the mm wave BS will arrange a k hop limited path from p_1 to p_2 . But on the other hand, if they don't satisfy either one of these constraints, the mm wave BS arranges a path to their nearest mm wave BS, which can act as a data portal.

When a request for a link from p_1 to p_2 arrives, the candidate links are chosen from the visibility graph in such a way that the expected data rate of the link chosen is maximized. The maximum data rate of a particular link can be easily computed using Shannon's capacity theorem and by the use of Frii's transmission equation, as mentioned in [18].

For each hop, in the path, we have the knowledge of probability of obstruction in the form of the weight of the corresponding edge in the visibility graph. Using this information, we can easily compute the expected data rate using that link. For a k hop link, the minimum of the expected data rates is the expected data rate of the entire k hop link, as the weakest link will determine the rate of information through the k hop link. There will be usually multiple candidate links, which are suggested by the visibility graph. The K-hop limited link, which has maximum of the expected data rate is finally chosen as the best relay, and is communicated to the involved devices and relays.

Exploration Phase: If an exploration phase is possible, then the relay selection algorithm also has to be tweaked accordingly to use the modified visibility graph. As

Algorithm 3: Visibility Graph constructed by LTE BS

Data: $G, U, R, m, T, k, mode$
Result: $VG = (V, E), Tr = KDTree(U \cup R \cup m)$

```
1 Function CreateVGraph( $G, U, R, m, T, k, mode$ ):  
2    $V \leftarrow U \cup R \cup m$   
3    $Tr = KDTree(V)$   
4   for  $p_1 \in V$  do  
5      $AddToVGraph(p_1, V, Tr, G, mode)$   
6   end  
7   return  $(V, E)$   
8 EndFunction  
9 Function EditVGraph( $G, D, T, mode$ ):  
10  Remove  $D$  from  $V$   
11  Remove  $D$  from  $Tr$   
12  Add new location of  $D$  in  $Tr$   
13   $AddToVGraph(D, V, Tr, G, mode)$   
14 EndFunction  
15 Function AddToVGraph( $p_1, V, Tr, G, mode$ ):  
16   $P \leftarrow Tr.GetNeighbours(p_1, T)$   
17  for  $p_2 \in P$  do  
18     $B_{p_1, p_2} \leftarrow$  Set of points between  $p_1$  and  $p_2$  using bresenham's algorithm  
19    if  $mode == "explore"$  then  
20       $E(p_1, p_2) \leftarrow \sum_{b \in B_{p_1, p_2}} G_b^d$   
21    else  
22       $E(p_1, p_2) \leftarrow \prod_{b \in B_{p_1, p_2}} (G_b^f + G_b^d)$   
23    end  
24  end  
25 EndFunction  
26 Function DeleteFromVGraph( $p_1, V, Tr$ ):  
27  Remove  $p_1$  from  $V$   
28  Remove  $p_1$  from  $Tr$   
29 EndFunction
```

the modified visibility graph contains the uncertainty in each hop, the score of each link can be calculated by using the sum of all the weights of each edge in the k-hop link suggested by the visibility graph, and the link which has the maximum score is finally used for the communication.

The procedure in Algorithm 4 is the relay selection algorithm in detail. The K hop limited BFS in line 4 is $O(N)$, where N is the number of neighbours in k hop radius. The As we can see, in line 3, the score is calculated differently when compared to line 5, where the relay selection mode is *normal*. Finally the algorithm returns the path with the best score, which corresponds to the path with the highest data rate in *normal* mode, or the path with the highest uncertainty in *explore* mode. We can easily see here, that the link performance, will be progressively worse in the *explore* mode, as it obsessively tries to form links through areas, where not much is known.

Algorithm 4: Relay selection by LTE BS

Data: $p_s, p_d, G, k, mode, VG$
Result: Suggested path

- 1 $P \leftarrow VG.hopLimitedBFS(p_s, p_d, G, k)$
- 2 **for** $p \in P$ **do**
- 3 **if** $mode == "explore"$ **then**
- 4 $S_p \leftarrow \sum_{e \in p} w(e)$
- 5 **else**
- 6 $w \leftarrow \prod_{e \in p} w(e)$
- 7 $S_p \leftarrow \min_{e \in p} (W \log_2(1 + MAX_SINR_e))$
- 8 $S_p \leftarrow w * S_p$
- 9 **end**
- 10 **end**
- 11 **return** $(p^* : S_p^* = \max_{p \in P} R'_p)$

Chapter 6

Simulation and Results

6.1 Simulation Setup

For simulating the algorithm, we have assumed some system parameters. We have considered a square area of $1500\text{m} \times 1500\text{m}$, with an LTE BS in the center of the area, surrounded by 6 mm wave BS' in the form of a regular hexagon, equally placed so as to cover maximal area. For the purpose of comparing the efficacy of the algorithm under various kinds of obstacles density, we have assumed two different sorts of obstacle placement. Sparse obstacles and dense obstacles. In a Sparse map, we have assumed 100 15×15 obstacles, 5 20×20 obstacles and 6 100×100 obstacles. In the dense map, we have considered 100 15×15 obstacles, 50 20×20 obstacles and 20 100×100 obstacles, an almost 3 times increase in the obstacle density.

In addition to the static obstacles, there are 300 3×3 dynamic obstacles constantly moving around the map, using the random waypoint model, with thinking period randomly chosen in the range $(0\text{s}, 10\text{s})$. Eventhough, the above algorithm does not model dynamic obstacles, those are still deployed to test how both algorithms fare in the presence of dynamic obstacles.

There are 1300 UEs moving around the map, continuously in the same random waypoint model, with thinking period in the range of (0s,10s). The maximum speed of a UE is limited to 4 kmph. Thinking period adds some realism to the movement, as the users generally dont keep continuously moving around. Also, thinking periods, allow us to use the computational benefits of not having to update the location of UEs in every 0.25s or in each iteration.

The coverage of a UE or mm wave BS is chosen to be 150m and hops are capped at 3. The time period of each iteration is chosen to be 0.25s.

We assume that half the UEs, chosen randomly, are willing to act as a relay at any point of time. And in the remaining half, half of them will act as senders and the rest will act as receivers.

Due to the pseudo-stationary assumption, we also assume that all the movement happens between the iterations and not during a single iteration. This is a reasonable assumption to make since the block sizes are 1m x 1m, even if the UEs or obstacles are in motion during the time interval, if they remain in the same cell, the assumption still holds.

We assume that if there is an obstacle in the path of a link, there will be significant attenuation at that epoch, so much so that, it will pull the measured SINR below the acceptable threshold.

The existing learning algorithm in [18], assumes initially that the entire map is free. In our algorithm, we consider the entire map's belief in entirely uncertain, i.e. the belief masses of each of the cell in the entire map is (0,0,1) initially.

6.2 Results

Here we compare the proposed procedure with the learning based approach proposed by [18]. In [18], the updation rule used, is very similar to yager’s combination rule, if we consider the probability as belief mass of *occupied* class. So we used the yager’s updation rule for the method proposed in [18], to make both the maps easily comparable. Also, the existing model, learns information about dynamic obstacles in addition to learning about the static map. Since in this report, we are only modelling static obstacles, learning of the dynamic obstacle was not done for the existing algorithm, in order to maintain a level playing field.

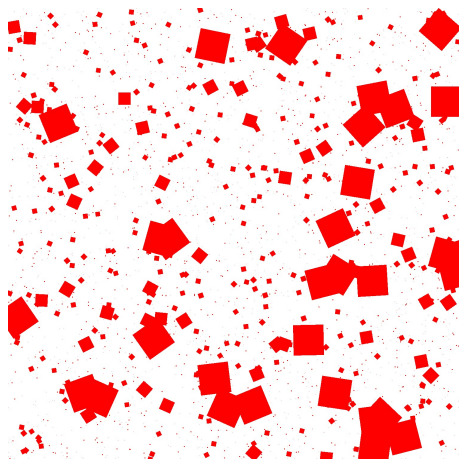


Figure 6.1: Actual simulation area

We calculated the accuracy , precision and recall of the estimated map with the actual map, at the end of each iteration, by considering that if the belief mass of a particular cell being occupied is greater than or equal to the mass of the cell being empty, then we consider the particular cell occupied, and vice versa. Accuracy, precision and recall was calculated using the formula below. The figure 6.2, shows what True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) correspond to.

		<i>Actual Occupancy</i>	
		Occupied	Free
<i>Predicted Occupancy</i>	Occupied	True Positive (TP)	False Positive (FP)
	Free	False Negative (FN)	True Negative (TN)

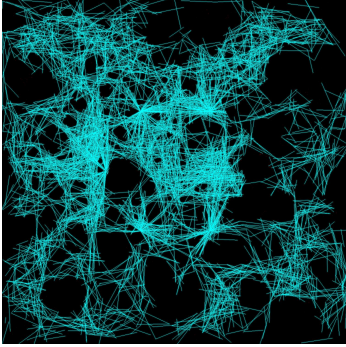
Figure 6.2: Confusion Matrix

$$\begin{aligned}
 Accuracy &= \frac{\text{Number of correctly predicted grids}}{\text{Total Number of grids}} \\
 &= \frac{TP + TN}{TP + TN + FP + FN}
 \end{aligned} \tag{6.1}$$

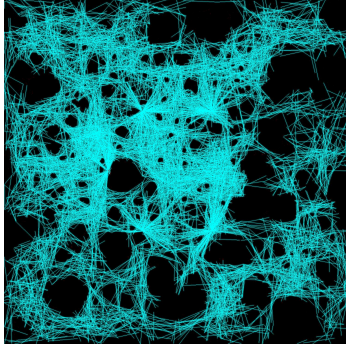
$$\begin{aligned}
 Precision &= \frac{\text{Number of correctly predicted occupied grids}}{\text{Number of grids predicted as } \textit{occupied}} \\
 &= \frac{TP}{TP + FP}
 \end{aligned} \tag{6.2}$$

$$\begin{aligned}
 Recall &= \frac{\text{Number of correctly predicted occupied grids}}{\text{Number of actually occupied grids}} \\
 &= \frac{TP}{TP + FN}
 \end{aligned} \tag{6.3}$$

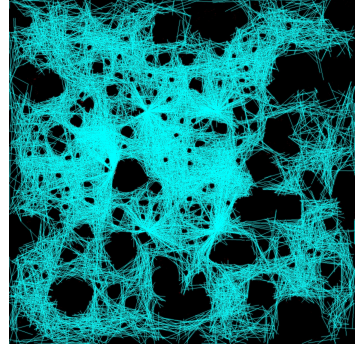
Accuracy measures how much cells, the algorithm tags correctly in general. Pre-



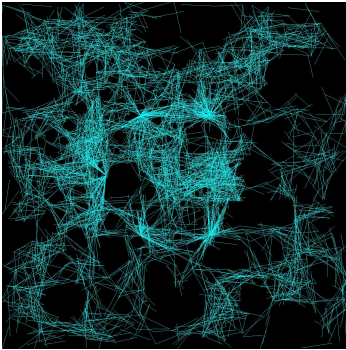
(a) At the end of 100 iterations



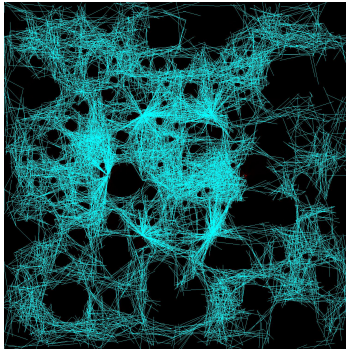
(b) At the end of 200 iterations



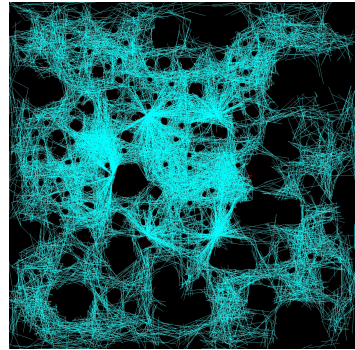
(c) At the end of 300 iterations



(d) At the end of 100 iterations



(e) At the end of 200 iterations



(f) At the end of 300 iterations

Figure 6.3: The maps after different stages of learning. The top row corresponds to the proposed model, and the bottom row corresponds to the existing model

	Dense map	Sparse map
Static Obstacle sizes	15,20,100	15,20,100
Static Obstacle nos.	100,50,20	100,5,6
Dynamic Obstacle nos.	300	100
Dynamic Obstacles size	3	3
θ	0.4	0.25
O_d	0.003	0.001
ϵ	0.01	0.01
γ	0.05	0.05
σ_0	100	100
α_t	0.001	0.001
l	3	3
t	(0.45, 0.55)	(0.45, 0.55)

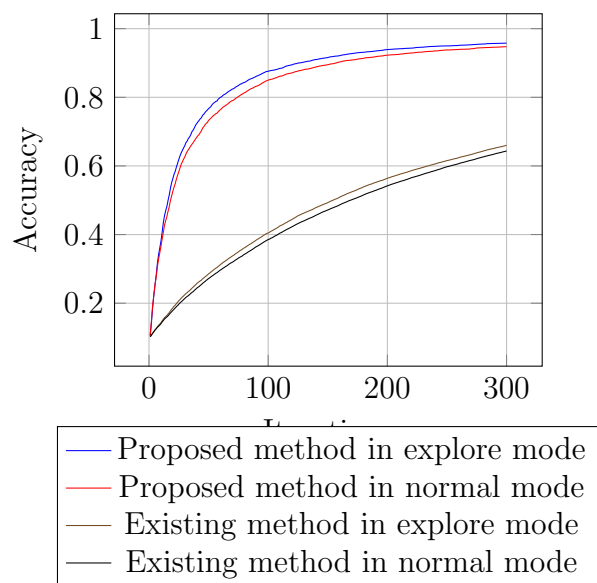
Table 6.1: Hyperparameters chosen

cision measures how correct the predictions of a particular class is. In this case, we are measuring how precise our *occupied* predictions are. Recall is the ratio which measures how many occupied cells we are successfully finding out from the set of actually occupied cells. Precision measures how specifically we are tagging a grid as an obstacle, whereas recall measures how good the algorithm is at finding an obstacle. We need to compare all three metrics, because the number of unoccupied cells far outweigh occupied cells, to come to a conclusion only with accuracy.

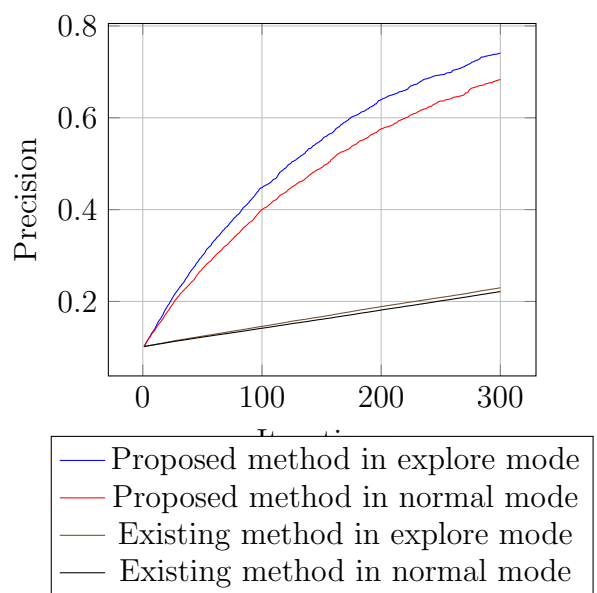
For the above conditions and using the hyperparameters mentioned in Table 6.1, both the algorithms were tested for 300 iterations, with each iteration consisting of 325 communications. Also, the visibility graph and relay selection proposed by this report was used for the comparisons. Also, both the algorithms were tested for the same amount of time, when the relay selection mode was *exploratory*. The map in Figure 6.1 is one such dense maps, in which the simulation took place. The results of the learning can be seen in Figure 6.3. The cyan zones correspond to the free zones, and the black zones correspond to obstacles.

The accuracy in Figure 6.4a and precision in Figure 6.4b of the proposed method

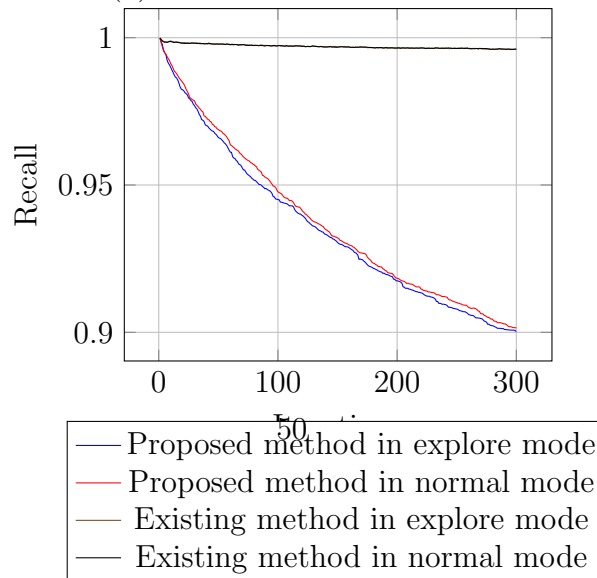
raises much faster than that of the existing method regardless of the mode of relay selection used. The recall in 6.4c of the proposed method though falls with iterations and later stabilises at a certain point. The above behaviour suggests that existing method tags occupied spaces much more easily when compared to the proposed method, as the precision of the existing method rises slower when compared to the proposed method. This is due to the fact that if an obstacle is found, the existing method tags the entire straight line from the sender to the receiver, as an obstacle with a probability inversely proportional to the length. The proposed method considers the fact that, the edges of the obstacle can only be observed by such an experiment and cautiously avoids making assumptions about the interiors of the obstacles. Due to this, the insides of a large obstacles can be seen as a hugely uncertain area in the proposed method, since it was actually never observed at all.



(a) Accuracy w.r.t iterations



(b) Precision w.r.t iterations



Chapter 7

Conclusion and Future Work

In this report, we proposed a evidential framework based learning algorithm which also takes advantage of the spatial correlation, and showed that it learns faster than the other learning algorithm for relay selection. This does not need any sort of prior data such as satellite terrain data. Also, we proposed a visibility graph algorithm which reduces the overhead of the number of location updates. We also proposed a relay selection mode called *exploration* mode, using which the map can be learnt faster than the normal mode.

Future work in this area can concentrate on the relay selection algorithm here, which is still expensive, as it involves a BFS every time, and this becomes unmanageable once the number of hops increase. An efficient relay selection algorithm with a more efficient data structure can be looked into to heavily improve the learning algorithm along with the system in place.

Bibliography

- [1] Bentley, J.L.: Multidimensional binary search trees used for associative searching. *Commun. ACM* 18(9), 509–517 (Sep 1975), <https://doi.org/10.1145/361002.361007>
- [2] Bresenham, J.E.: Algorithm for computer control of a digital plotter. *IBM Systems Journal* 4(1), 25–30 (1965)
- [3] Davies, E.R.: *Machine Vision: Theory, Algorithms, Practicalities*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2004)
- [4] Dempster, A., et al.: Upper and lower probabilities induced by a multivalued mapping. *The Annals of Mathematical Statistics* 38(2), 325–339 (1967)
- [5] Dezert, J., Moras, J., Pannetier, B.: Environment perception using grid occupancy estimation with belief functions. In: 2015 18th International Conference on Information Fusion (Fusion). pp. 1070–1077 (2015)
- [6] Dhiman, V., Kundu, A., Dellaert, F., Corso, J.J.: Modern map inference methods for accurate and fast occupancy grid mapping on higher order factor graphs. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). pp. 2037–2044 (2014)

- [7] Elfes, A.: Using occupancy grids for mobile robot perception and navigation. *Computer* 22(6), 46–57 (1989)
- [8] Irish, A.T., Isaacs, J.T., Quitin, F., Hespanha, J.P., Madhow, U.: Belief propagation based localization and mapping using sparsely sampled gnss snr measurements. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). pp. 1977–1982 (2014)
- [9] J. Qiao, X. S. Shen, J.W.M.Q.S.Y.H., Lei, L.: Enabling device-to-device communications in millimeter-wave 5g cellular networks. *IEEE Communications Magazine* 53, 209–215 (2015)
- [10] Kim, K., Summet, J., Starner, T., Ashbrook, D., Kapade, M., Essa, I.: Localization and 3d reconstruction of urban scenes using gps. pp. 11 – 14 (11 2008)
- [11] Kim, S., Kim, J.: Recursive bayesian updates for occupancy mapping and surface reconstruction. In: ICRA 2014 (2014)
- [12] Lefevre, E., Colot, O., Vannoorenberghe, P.: Belief function combination and conflict management. *Information Fusion* 3(2), 149 – 162 (2002)
- [13] Mouhagir, H., Cherfaoui, V., Talj, R., Aioun, F., Guillemard, F.: Using evidential occupancy grid for vehicle trajectory planning under uncertainty with tentacles. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). pp. 1–7 (2017)
- [14] O’Callaghan, S.T., Ramos, F.T.: Gaussian process occupancy maps. *The International Journal of Robotics Research* 31(1), 42–62 (2012)

- [15] Pagac, D., Nebot, E., Durrant-Whyte, H.: An evidential approach to map-building for autonomous vehicles. *Robotics and Automation, IEEE Transactions on* 14, 623 – 629 (09 1998)
- [16] Paskin, M., Thrun, S.: Robotic mapping with polygonal random fields. *CoRR* abs/1207.1399 (2012)
- [17] Pi, Z., Khan, F.: An introduction to millimeter-wave mobile broadband systems. *IEEE Communications Magazine* 49(6), 101–107 (2011)
- [18] Sarkar, S., Ghosh, S.C.: Relay selection in millimeter wave d2d communications through obstacle learning. In: 2020 International Conference on COMMunication Systems NETworkS (COMSNETS). pp. 468–475 (2020)
- [19] Shafer, G.: *A mathematical theory of evidence* (1976)
- [20] Sun, K., Saulnier, K., Atanasov, N., Pappas, G.J., Kumar, V.: Dense 3-d mapping with spatial correlation via gaussian filtering. *CoRR* abs/1801.07380 (2018)
- [21] Thrun, S.: Learning occupancy grids with forward models. In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium* (Cat. No.01CH37180). vol. 3, pp. 1676–1681 vol.3 (2001)
- [22] Weissman, A., Ben-Moshe, B., Levi, H., Yozevitch, R.: 2.5d mapping using gnss signal analysis. In: 2013 10th Workshop on Positioning, Navigation and Communication (WPNC). pp. 1–6 (2013)
- [23] Yassin, A., Nasser, Y., Al-Dubai, A.Y., Awad, M.: Mosaic: Simultaneous localization and environment mapping using mmwave without a-priori knowledge. *IEEE Access* 6, 68932–68947 (2018)