

**IMPROVED PATH LOSS SIMULATION INCORPORATING THREE-  
DIMENSIONAL TERRAIN MODEL USING PARALLEL CO-  
PROCESSORS**

By

**LOO ZHANG BIN**

A dissertation submitted to the Department of Electrical and Electronics  
Engineering,  
Lee Kong Chian Faculty of Engineering and Science,  
Universiti Tunku Abdul Rahman,  
in partial fulfilment of the requirements for the degree of  
Master of Engineering Science  
May 2017

## **ABSTRACT**

### **IMPROVED PATH LOSS SIMULATION INCORPORATING THREE-DIMENSIONAL TERRAIN MODEL USING PARALLEL CO-PROCESSORS**

**LOO ZHANG BIN**

Current network simulators abstract out wireless propagation models due to the high computation requirements for realistic modelling. As such, there is still a large gap between the results obtained from simulators and real world scenario. In this dissertation, we present a framework for improved path loss simulation built on top of existing network simulation software, NS-3. Different from the conventional disc model, the proposed simulation also considers the diffraction loss over multiple knife edges computed using the Epstein and Peterson's approximate technique through the use of actual terrain data to give an accurate estimate of path loss between a transmitter and a receiver. The drawback of high computation requirements is relaxed by offloading the computationally intensive components onto an inexpensive off-the-shelf parallel co-processor, which is a NVIDIA GPU. Three different variant of network simulations are implemented, disc model with conventional CPU architecture, terrain aware model with conventional CPU architecture, and terrain aware model with heterogeneous system architecture to measure the efficiency of the proposed simulator. The effectiveness of the proposed simulations are measured using point-to-point model and modified random walk model. Experiments are

performed using actual terrain elevation data provided from United States Geological Survey. An acceleration of up to 20x to 42x faster using the GPU as a parallel co-processor is achieved in computing the path loss between two nodes using terrain elevation data. The result shows that the path losses between two nodes are greatly affected by the terrain profile between these two nodes. The result also suggests that the common strategy to place the transmitter in the highest position may not always work. Lastly, the proposed simulation framework allows researchers to easily implement different path loss simulation models on NS-3 depending on the terrains they want to simulate by varying the parameters since most of these models rely on similar computations.

## ACKNOWLEDGEMENT

This dissertation would not have been possible without the help of so many people in so many ways. First and foremost, I would like to express my deepest appreciation to my supervisors, Dr. Lee Kim Yee and Dr. Yap Wun She for their guidance on knowledge and practical issues beyond textbooks. Moreover, they have been very supportive all the way through my dissertation revision. I would also like to thank Dr. Chong Poh Kit as my previous supervisor, who was always willing to help and shared a lot of his research experience and continuously to provide me motivation, support and advises. I am very fortunate to have Mr. Yong Keh Kok as my supervisor during my internship in MIMOS which motivated me and introduced me to Dr. Chong, had I not made acquaintance with them, I would not pursued my master's degree. Lastly, I would like to thank my family for the continuous support and encouraging me with their best wishes.

**LEE KONG CHIAN FACULTY OF ENGINEERING AND SCIENCE**

**UNIVERSITI TUNKU ABDUL RAHMAN**

Date: \_\_\_\_\_

**SUBMISSION OF DISSERTATION**

It is hereby certified that **LOO ZHANG BIN** (ID No: **14UEM07867**) has completed this dissertation entitled “IMPROVED PATH LOSS SIMULATION INCORPORATING THREE-DIMENSIONAL TERRAIN MODEL USING PARALLEL CO-PROCESSORS” under the supervision of Dr. Lee Kim Yee (Supervisor) and Dr. Yap Wun She (Co-supervisor) from the Department of Electrical and Electronic Engineering, Lee Kong Chian Faculty of Engineering and Science.

I understand that University will upload softcopy of dissertation in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

\_\_\_\_\_  
(Loo Zhang Bin)

## APPROVAL SHEET

This dissertation entitled “**IMPROVED PATH LOSS SIMULATION INCORPORATING THREE-DIMENSIONAL TERRAIN MODEL USING PARALLEL CO-PROCESSORS**” was prepared by LOO ZHANG BIN and submitted as partial fulfilment of the requirements for the degree of Master of Engineering Science at Universiti Tunku Abdul Rahman.

Approved by:

---

(Dr. LEE KIM YEE)

Date:.....

Supervisor

Department of Electrical and Electronics Engineering  
Lee Kong Chian Faculty of Engineering and Science  
Universiti Tunku Abdul Rahman

---

(Dr. YAP WUN SHE)

Date:.....

Co-supervisor

Department of Electrical and Electronics Engineering  
Lee Kong Chian Faculty of Engineering and Science  
Universiti Tunku Abdul Rahman

## DECLARATION

I hereby declare that the dissertation is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

Name \_\_\_\_\_

Date \_\_\_\_\_

## TABLE OF CONTENTS

	<b>Page</b>
<b>ABSTRACT</b>	<b>ii</b>
<b>ACKNOWLEDGEMENTS</b>	<b>iv</b>
<b>PERMISSION SHEET</b>	<b>v</b>
<b>APPROVAL SHEET</b>	<b>vi</b>
<b>DECLARATION</b>	<b>vii</b>
<b>TABLE OF CONTENTS</b>	<b>viii</b>
<b>LIST OF TABLES</b>	<b>x</b>
<b>LIST OF FIGURES</b>	<b>xi</b>
<b>CHAPTER</b>	
<b>1.0 INTRODUCTION</b>	<b>1</b>
1.1 Problem Statement	4
1.2 Objective	5
1.3 Contributions	7
1.4 Organisation of the Dissertation	7
<b>2.0 LITERATURE REVIEW</b>	<b>8</b>
2.1 Wireless Propagation Modes	8
2.1.1 Line-of-Sight Wireless Transmission	11
2.1.2 Diffraction Loss on Multiple Knife-Edge Obstructions	12
2.1.3 Diffraction Loss Computation using Epstein-Peterson Model	17
2.1.4 Digital Elevation Model (DEM)	19
2.2 Parallel Computing	21
2.2.1 Parallel Co-Processor	22
2.2.2 CUDA	22
2.2.3 Variants of Heterogeneous System Architecture	24
2.2.4 Existing GPU Implementations of Different Wireless Propagation Models	29
2.2.5 NS-3	32
<b>3.0 METHODOLOGY</b>	<b>34</b>
3.1 Propose Path Loss Simulation Framework	34
3.1.1 Disc Model with Conventional CPU Architecture	34
3.1.2 The Terrain Aware Model Implemented using Conventional CPU Architecture	36
3.1.3 The Terrain Aware Model Implemented using Heterogeneous System Architecture	38
3.2 Detailed Data Flow of The Proposed Path Loss Simulation Framework	40
3.2.1 Data Flow of The Disc Model Implemented using Conventional CPU Architecture	40



3.2.2	Data Flow of The Terrain Aware Model Implemented using Conventional CPU Architecture	41
3.2.3	Data Flow of The Terrain Aware Model Implemented using Heterogeneous System Architecture	48
3.3	Experimental Setup	51
3.4	Data Selection and Analysis	52
<b>4.0</b>	<b>RESULTS AND DISCUSSION</b>	<b>57</b>
4.1	Modified Random Walk Mobility Model	57
4.1.1	Preliminary Analysis on Terrain Profile	57
4.1.2	Path Loss Analysis	64
4.1.3	Speed Performance of The Terrain Aware Model Implemented Using CPU Architecture and Heterogeneous System Architecture	66
4.2	Point-to-Point Model	69
4.2.1	Conventional Method: Disc Model	70
4.2.2	Proposed Method: The Terrain Aware Model	72
<b>5.0</b>	<b>CONCLUSION</b>	<b>92</b>
5.1	Conclusion	92
5.2	Future Work	93
	<b>REFERENCES</b>	<b>94</b>
	<b>LIST OF PUBLICATION</b>	<b>104</b>

## LIST OF TABLES

<b>Table</b>		<b>Page</b>
1.1	Pros and cons of wireless communications as compare to wired communications	2
2.1	Comparison between the actual measurements with the diffraction loss computed using different models	16
2.2	Difference of diffraction loss (in terms of dB) between actual FSH-3 measurements and different approximation techniques	17
2.3	Summary of different wireless network simulation	29
3.1	Two types of terrain profile	54
4.1	Terrain elevation range and path type for different terrain profile	59
4.2	Details of the selected terrain profiles	73

## LIST OF FIGURES

<b>Figures</b>		<b>Page</b>
2.1	Three wireless propagation models (Stallings, 2005)	9
2.2	Multipath interference (Stallings, 2005)	12
2.3	The Fresnel zone between a transmitter and a receiver	14
2.4	Epstein-Peterson path geometry method	18
2.5	Sample of DEM data file	21
2.6	CUDA architecture (Gupta, 2013)	23
2.7	Four approaches of GPU-based discrete event simulation (Andelfinger et al., 2011)	25
2.8	Speedup achieved by GPU-based parallelisation of individual signal processing algorithms (Andelfinger et al., 2011)	27
2.9	Cunetsim framework architecture and dependency (Bilel et al., 2012)	27
2.10	Running time (bar) on three different platforms with performance (line) in terms of megacells per second for the FDTD-CPML method (Valcarce et al., 2008)	30
2.11	Execution time of Hoehner channel model in CPU and GPU (Abdelrazak et al., 2009)	31
3.1	Flow chart of disc model using conventional CPU architecture	35
3.2	Flow chart of terrain aware model using conventional CPU architecture	37
3.3	Flow chart of terrain aware model using heterogeneous system architecture	39
3.4	DFD level 0 for the disc model	41
3.5	DFD level 0 for the terrain aware model implemented using conventional CPU architecture	42

3.6	Rotating path profile	45
3.7	Pseudocode for searching obstructing peaks	46
3.8	Pseudocode for calculating the height difference of the obstructing peaks	47
3.9	Pseudocode for computing the diffraction loss	48
3.10	DFD level 0 for the terrain aware model implemented using heterogeneous system architecture	50
3.11	Travelling pattern of a mobile node using 2D random walk mobility model (Camp et al., 2012)	53
3.12 (a)	Low terrain elevation difference profiles (a) Adelanto	54
3.12 (b)	Low terrain elevation difference profiles (b) Amboy Crater	54
3.12 (c)	Low terrain elevation difference profiles (c) Jersey City	55
3.12 (d)	Low terrain elevation difference profiles (d) Amsterdam	55
3.13 (a)	High terrain elevation difference profiles (a) Arrowhead Butte	55
3.13 (b)	High terrain elevation difference profiles (b) Acton	55
3.13 (c)	High terrain elevation difference profiles (a) Caples Lake	56
4.1	Free space path versus blocked path in different terrain profiles	58
4.2 (a)	Low terrain elevation difference profile (a) Adelanto	60
4.2 (b)	Low terrain elevation difference profile (a) Amboy Crater	60
4.2 (c)	Low terrain elevation difference profile (c) Jersey City	61
4.2 (d)	Low terrain elevation difference profile (d) Amsterdam	61

4.3 (a)	High terrain elevation difference profile (a) Arrowhead Butte	62
4.3 (b)	High terrain elevation difference profile (b) Acton	62
4.3 (c)	High terrain elevation difference profile (d) Caples Lake	63
4.4 (a)	Adelanto path profiles (a) Adelanto free path	64
4.4 (b)	Adelanto path profiles (b) Adelanto partial blocked path	64
4.4 (c)	Adelanto path profiles (c) Adelanto blocked path	65
4.5 (a)	Adelanto block path with 5 obstructing peaks (a) Adelanto blocked path 1	66
4.5 (b)	Adelanto block path with 5 obstructing peaks (b) Adelanto blocked path 2	66
4.5 (c)	Adelanto block path with 5 obstructing peaks (c) Adelanto blocked path 3	66
4.6 (a)	Simulation time and speedup for terrain profiles with low terrain elevation difference (a) Simulation time of CPU and HSA	67
4.6 (b)	Simulation time and speedup for terrain profiles with low terrain elevation difference (b) Speedup of HSA over CPU	68
4.7 (a)	Simulation time and speedup for terrain profiles with high terrain elevation difference (a) Simulation time of CPU and HSA	68
4.7 (b)	Simulation time and speedup for terrain profiles with high terrain elevation difference (b) Speedup of HSA over CPU	68
4.8 (a)	Overall simulation time and speedup for all terrain profiles (a) Simulation time of CPU and HSA	69
4.8 (b)	Overall simulation time and speedup for all terrain profiles (b) Speedup of HSA over CPU	69
4.9 (a)	Path loss simulation based on disc model for terrain profiles with low terrain elevation difference (a) Adelanto	70

4.9 (b)	Path loss simulation based on disc model for terrain profiles with low terrain elevation difference (b) Amboy Crater	71
4.9 (c)	Path loss simulation based on disc model for terrain profiles with low terrain elevation difference (c) Jersey City	71
4.9 (d)	Path loss simulation based on disc model for terrain profiles with low terrain elevation difference (d) Amsterdam	71
4.10 (a)	Path loss based on disc model for terrain profiles with high terrain elevation difference (a) Arrowhead Butte	72
4.10 (b)	Path loss based on disc model for terrain profiles with high terrain elevation difference (b) Acton	72
4.10 (c)	Path loss based on disc model for terrain profiles with high terrain elevation difference (c) Caples Lake	72
4.11 (a)	Simulation results on Adelanto terrain profile (a) 2D view of Adelanto terrain profile	74
4.11 (b)	Simulation results on Adelanto terrain profile (b) 3D rotated view of Adelanto terrain profile	75
4.11 (c)	Simulation results on Adelanto terrain profile (c) 2D rotated view of Adelanto's path loss simulation	75
4.11 (d)	Simulation results on Adelanto terrain profile (d) 3D rotate view of Adelanto's path loss simulation	76
4.12 (a)	Simulation results on Amboy Crater terrain profile (a) 2D view of Amboy Crater terrain profile	77
4.12 (b)	Simulation results on Amboy Crater terrain profile (b) 3D rotated view of Amboy Crater terrain profile	77
4.12 (c)	Simulation results on Amboy Crater terrain profile (c) 2D rotated view of Amboy Crater's path loss simulation	78

4.12 (d)	Simulation results on Amboy Crater terrain profile (d) 3D rotate view of Amboy Crater's path loss simulation	78
4.13 (a)	Simulation results on Jersey City terrain profile (a) 2D view of Jersey City terrain profile	79
4.13 (b)	Simulation results on Jersey City terrain profile (b) 3D rotated view of Jersey City terrain profile	80
4.13 (c)	Simulation results on Jersey City terrain profile (c) 2D rotated view of Jersey City's path loss simulation	80
4.13 (d)	Simulation results on Jersey City terrain profile (d) 3D rotate view of Jersey City's path loss simulation	81
4.14 (a)	Simulation results on Amsterdam terrain profile (a) 2D rotated view of Amsterdam terrain profile	82
4.14 (b)	Simulation results on Amsterdam terrain profile (b) 3D rotated view of Amsterdam terrain profile	82
4.14 (c)	Simulation results on Amsterdam terrain profile (c) 2D rotated view of Amsterdam's path loss simulation	83
4.14 (d)	Simulation results on Amsterdam terrain profile (d) 3D rotate view of Amsterdam's path loss simulation	83
4.15 (a)	Simulation results on Arrowhead Butte terrain profile (a) 2D rotated view of Arrowhead Butte terrain profile	84
4.15 (b)	Simulation results on Arrowhead Butte terrain profile (b) 3D rotated view of Arrowhead Butte terrain profile	85
4.15 (c)	Simulation results on Arrowhead Butte terrain profile (c) 2D rotated view of Arrowhead Butte's path loss simulation	85
4.15 (d)	Simulation results on Arrowhead Butte terrain profile (d) 3D rotate view of Arrowhead Butte's path loss simulation	86

4.16 (a)	Simulation results on Acton terrain profile (a) 2D rotated view of Acton terrain profile	87
4.16 (b)	Simulation results on Acton terrain profile (b) 3D rotated view of Acton terrain profile	87
4.16 (c)	Simulation results on Acton terrain profile (c) 2D rotated view of Acton's path loss simulation	88
4.16 (d)	Simulation results on Acton terrain profile (d) 3D rotate view of Acton's path loss simulation	88
4.17 (a)	Simulation results on Caples Lake terrain profile (a) 2D rotated view of Caples Lake terrain profile	89
4.17 (b)	Simulation results on Caples Lake terrain profile (b) 3D rotated view of Caples Lake terrain profile	90
4.17 (c)	Simulation results on Caples Lake terrain profile (c) 2D rotated view of Caples Lake's path loss simulation	90
4.17 (d)	Simulation results on Caples Lake terrain profile (d) 3D rotate view of Caples Lake's path loss simulation	91



## CHAPTER 1

### INTRODUCTION

Communication is the process of exchanging information between two or more entities. A basic communication involves at least two entities named as a transmitter and a receiver.

A communication channel is needed to transmit the information from the transmitter to the receiver. Generally, the communication channel can be categorised into two different types: wired and wireless. Wired channel may include wires that carry the electrical signal or an optical fiber that carries the information on modulated light beam. Meanwhile, wireless channel may include free space which the signal is radiated through an antenna that transmits the information through radio/electromagnetic waves or an underwater ocean channel that transmits the information acoustically (Proakis and Salehi, 2008).

The process of communication involves the transmission of signal. Transmission is the process of sending a signal from the transmitter antenna over a point-to-point or point-to-multipoint through a medium to the receiver antenna, in either wired or wireless. Signal degradation (i.e. the loss of quality of a signal) is an unavoidable problem encountered in the transmission of information over the channel due to interference. Main sources of interference

include amplitude and phase distortions, signal attenuation and multipath distortion.

To model the signal degradation and design a communication system which transmits information through wired or wireless channel, mathematical models are constructed in reflecting the most important characteristics of the transmission channel (Proakis and Salehi, 2008). For example, the additive noise channel (Shannon, 1948), the linear filter channel proposed by Salz (1985) that is applicable to wire line telephone and the linear time-variant filter channel proposed by Bello (1963) that is applicable to underwater acoustic channel and radio channel.

For better understanding the differences between wired and wireless communications, Table 1.1 lists out both the pros and cons of wireless communications as compared to wired communications.

Table 1.1: Pros and cons of wireless communications as compare to wired communications

Pros	Cons
<ul style="list-style-type: none"><li>• Higher flexibility and mobility for roaming</li><li>• Cheaper installation cost</li></ul>	<ul style="list-style-type: none"><li>• Transmission speed is slower and less efficient</li><li>• Susceptible to interference</li><li>• Coverage is limited due to path loss.</li></ul>

This project focuses on wireless communication network as the last decade radio equipment has gotten transportable, smaller, cheaper and

dependable, creating a demand for even greater freedom in the way people use the wireless communication networks.

Radio was the most intensively deployed wireless technology for wireless communications. Radio propagation is realised through the use of antenna that converts electric power into radio/electromagnetic waves. A wireless communication network is formed when a collection of entities or nodes are connected to enable wireless communication between these nodes. These nodes are normally called as base station and mobile station. Another reason for the focus of studies is that as compared to wired communication network, wireless communication network (Mark and Zhuang, 2003) offers a flexible information transmission platform that allows mobile users to roam without suffering significant performance degradation.

As the communication network has become too complex to be analysed using conventional analytical methods for a better understanding of system behaviour, network simulator is used to mimic the behaviour of a communication network. Another factor give rise to the introduction of network simulator is the large coverage of communication network which substantially increases the difficulty in studying the performance of communication network. In the simulation, different entities exist in the communication network are modelled. There exist many popular network simulators in the real world, for examples NS-3, OMNET++, OPNET, Qualnet and SSFNet.

## 1.1 Problem Statement

Generally, network or physical layer oriented simulators have been used to enable the performance studies on communications networks. Network layer simulators mainly focus on the performance and behaviour of the entire communication networks, such as medium access control issues and network layer aspects. On the other hand, physical layer oriented simulators (Hwang et al., 2009) focus on point-to-point link performance under specific channel conditions.

Both types of simulator fail to provide accurate result (Mittag et al., 2011) since network simulators abstract physical layer while physical layer oriented simulators do not consider the characteristics of overall communication network. To bridge the gap between physical and network layer perspectives and enable studies on cross-layer optimization, Mittag et al. (2011) integrated a detailed physical layer simulator into NS-3 network simulator, yet improved accuracy comes at a cost of increased computational effort, hence forth in their future work they aim to optimise the computation of the channel models by making use of parallelisation where possible by using available specialised hardware such as the general-purpose graphics processing unit (GPGPU). Besides, Stojanova et al. (2013) presented a comparison of network simulation performance between NS-2 and its successor NS-3; the result shows that NS-3 has a significant performance improvement due to the improved memory management and the use of aggregation system that prevents unnecessary parameters from being stored.

In radio based wireless communication networks, determining the proper number of deployed antennas (or stations) on the mountainous regions to achieve expected coverage becomes a challenging task. Mittag et al. (2011) did not consider the surface of terrain in evaluating the performance of the entire communication network even though radio transmission in mobile communication systems often takes place over irregular terrain.

The computational complexity of the network simulation will increase substantially when taking terrain profile of into account in computing path loss between two nodes (i.e., the transmitter and the receiver) that affect the coverage. Notice that path loss is the reduction in power density of electromagnetic waves (i.e. signal degradation) as it propagates through free space. Djinevski et al. (2015) also claimed that terrain profiles have major impacts on the optimal deployment of wireless sensor network. In addition, Philips et al. (2013) claimed that including terrain profile gave better predictions but it comes with additional computational complexity especially detailed vector data related to terrain profile is used, such as digital elevation model (DEM).

## **1.2 Objectives**

This dissertation is concerned with the improved path loss computation from the inclusion of terrain model into a wireless propagation model. One of the nodes can be considered as an antenna or a base station while the other

node can be considered as a mobile station or a receiver antenna. The path loss computation is important to determine the coverage of a wireless communication network.

Generally network simulators do not consider the terrain structure of deployed communications networks in their wireless propagation model. By including the terrain's surface (using detailed vector data, DEM) and wireless propagation model (considering free space loss and diffraction loss over multiple knife edge) into account, the computational complexity of a network simulation in computing the path loss between two nodes is increased substantially. Due to these concerns, two objectives of this research project to produce an improved path loss simulation in computing the path loss between two nodes are listed as follow.

1. To improve the accuracy in computing the path loss between two nodes by incorporating three-dimensional representation of a terrain's surface in a wireless network, i.e. DEM.
2. To offload compute-intensive processes of network simulation through the use of parallel co-processors, i.e. general-purpose graphics processing unit (GPGPU).

### **1.3 Contributions**

This research aims to present an improved path loss simulation in computing path loss between two nodes to determine the coverage issue using GPGPU to accelerate compute-intensive processes and real detailed terrain data expressed using digital elevation model (DEM) data file. The proposed simulation is successfully integrated into a popular network simulator, i.e., NS-3. Thus, such wireless simulator can be exploited by radio-frequency engineers for the planning and deployment of wireless networks in the real world. Most importantly, the technique used in the proposed simulation can be extended to simulate other wireless propagation models to develop a wireless network simulator with higher accuracy as most of these models rely on similar computations.

### **1.4 Organisation of the Dissertation**

The remainder of the dissertation is organised as follows. Chapter 2 provides the background that is required to understand the proposed terrain aware model in computing path loss between two antennas. Chapter 3 presents the proposed terrain aware model and its simulation along with the experimental setup and data selection. Chapter 4 provides the analysis of the simulation results based on two different models (i.e., the point-to-point model and the modified random walk model). Chapter 5 summarises the results obtained in this dissertation and provide suggestions for future research work.

## CHAPTER 2

### LITERATURE REVIEW

In this chapter, the background that is required to understand the proposed terrain aware model in computing path loss between two antennas is provided. In particular, different wireless propagation modes are presented. In order to understand the proposed terrain aware model, more focus is given on line-of-sight transmission and the diffraction loss. Lastly, the related work using different architectures and models are summarised and compared.

#### 2.1 Wireless Propagation Modes

As shown in Figure 2.1, there exist three wireless propagation modes (Stallings, 2005) where the signal is transmitted from a transmit antenna to a receive antenna: ground wave propagation, sky wave propagation and line-of-sight (LOS) propagation.



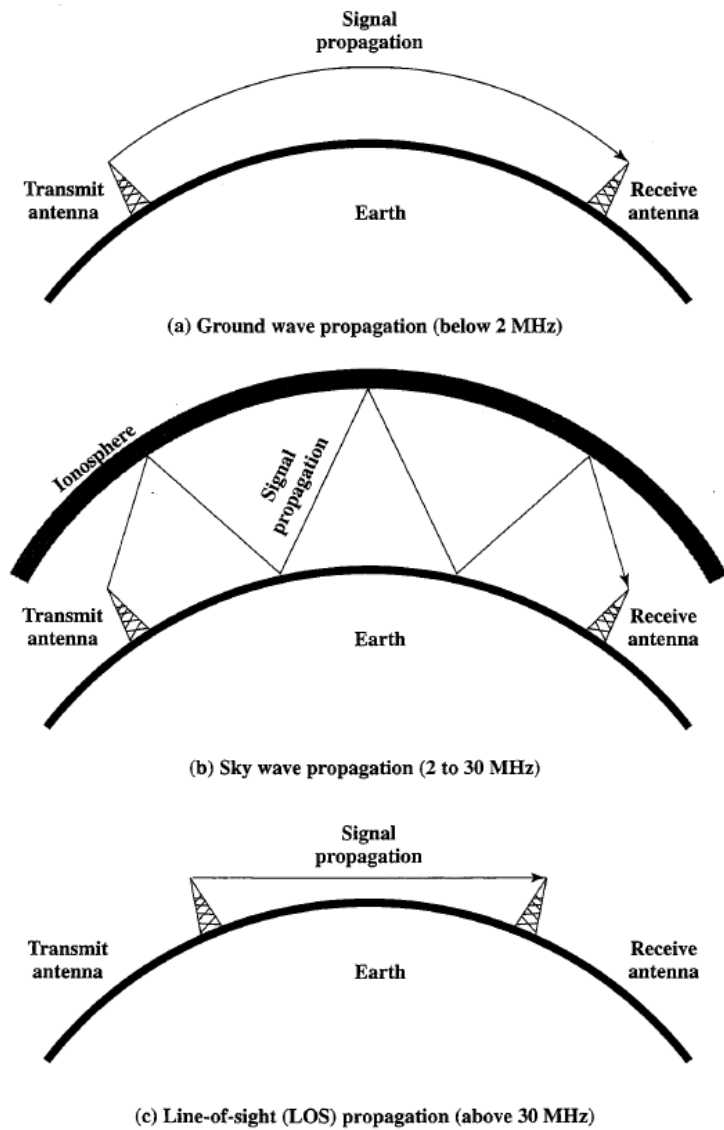


Figure 2.1: Three wireless propagation modes (Stallings, 2005)

Ground wave propagation is the dominant mode of propagation especially in the low and medium frequency portion of the radio spectrum (e.g., frequencies with below 2 MHz used in amplitude modulation broadcasting). Generally, ground wave propagates according to the contour of the earth. Besides, ground wave can propagate a considerable distance over the horizon.

Sky wave is the dominant mode of propagation for frequencies between 2 to 30 MHz (e.g., Amateur radio and Citizens band radio). More precisely, sky wave propagation is a signal from earth-based antenna that travels through number of hops, bouncing back and forth between the ionosphere and the earth surface. A signal can be picked up thousands of kilometres from the transmitter.

Lastly, line-of-sight (LOS) is the dominant propagation mode for frequencies above 30 MHz employed in satellite communication and mobile communication. For satellite communication, the signal is not reflected by the ionosphere and therefore can be transmitted between an earth station and satellite overhead that is not beyond the horizon. For ground-based communication, the transmit and receive antennas must be within a LOS of each other. The signal that is received will differ from the signal that is transmitted in any communications systems, due to various transmission impairments. Mansfield et al. (2016) and Truong et al. (2016) independently showed that the used of LOS in network simulation can provide high accuracy.

The broadcast frequency of 2.4 GHz is commonly used as it had been used since the introduction of the first mass-market radio networking (Desa et al., 2009). Besides, it is an unlicensed international industrial, scientific and medical radio band. For demonstration of our proposed simulation framework, we also use the broadcast frequency of 2.4 GHz and LOS propagation model. However, we remark that our proposed simulation framework is not limited to the broadcast frequency of 2.4 GHz only.

### 2.1.1 Line-Of- Sight Wireless Transmission

In LOS wireless transmission, the most significant impairments are attenuation distortion, free space loss, and multipath.

Attenuation is the strength of a signal falls off over distance for any transmission medium. For wireless transmission, a received signal must have sufficient strength such that the receiver can detect and interpret the signal. Besides, the strength of the transmitted signal must be higher than the noise to achieve error-free transmission. Lastly, attenuation is greater generally if the signal is transmitted with high frequency due to the distortion.

Free space loss (FSL) is the loss in signal strength of an electromagnetic wave resulted from a LOS path through air space (Friis, 1946). The calculation of FSL is carried out using the predefined frequency of propagation wave  $f$  and the distance  $d$  between a transmit antenna and a receive antenna. The FSL is derived from the Friis free space equation (Rappaport, 2002) as follows:

$$FSL = 32.4 + 20\log_{10}(d) + 20\log_{10}(f) \quad (2.1)$$

Notice that FSL is an isotropic radio model implying signal attenuation in all direction with similar magnitude over distance. Thus, the path loss computation between the transmitter and the receiver using FSL is only

accurate if and only if there are no obstacles which will cause the reflection or refraction of electromagnetic waves between the transmitter and the receiver.

### 2.1.2 Diffraction Loss of Multiple Knife-Edge Obstructions

Generally, there are obstacles/obstructions block the LOS between the transmit antenna and the receive antenna. For an example, Figure 2.2 illustrates the multipath interference typically existing in terrestrial fixed microwave and in mobile communications (Stallings, 2005). Due to the obstacles blocking the LOS between the antennas, the signals are reflected and will cause the transmission's delay.

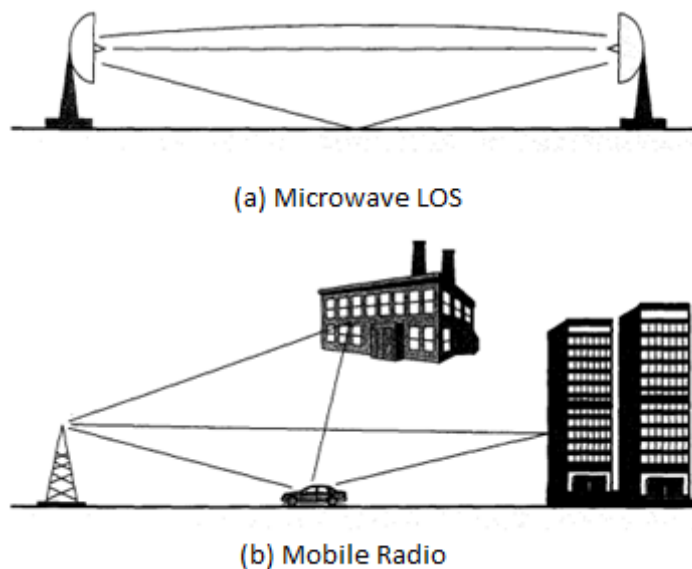


Figure 2.2: Multipath interference (Stallings, 2005)

The path loss simulation is affected by three multipath propagation mechanisms, i.e. reflection, diffraction and scattering. Reflection occurs when

some of the signal power may be reflected back to the transmit antenna rather than being transmitted to the receive antenna. Reflection normally happens when a signal encounters a surface that is large relative to the wavelength of the signal (e.g., surface earth and building walls). Meanwhile, scattering occurs when the signal are deviated from a straight trajectory by non uniformities in the transmission medium. Scattering normally happens when the size of an obstacle is comparable with the wavelength of the transmitted signal (e.g., transmission of signal during rainy day). Lastly, diffraction occurs at the edge of an impenetrable body that is large compared to the wavelength of the electromagnetic wave (e.g. terrain, buildings and vegetation).

This project focuses on diffraction as it is essential to estimate the signal attenuation caused by diffraction of electromagnetic/radio waves over terrains and building to determine the coverage of a wireless communication network. To include the terrain profile into the proposed path loss simulation framework, DEM data are used since it contains the necessary terrain information of the network communication environment in the real world.

When the LOS is obstructed by a single obstacle such as a mountain or a building, such signal attenuation caused by diffraction is called as diffraction loss over knife edge obstruction. Calculation of diffraction loss over knife edge obstruction is important as terrestrial-path propagation modelling requires prediction of the diffraction loss for a general path regardless of LOS or None-LOS (Topcu et al., 2015).

In designing point-to-point communication links, diffraction effects are minimised by ensuring enough clearance distance between the link LOS path and the nearest diffracting element (Durgin, 2009). To facilitate the design of communication link, ellipsoids of constant path difference are drawn around the link where one focus at the transmitter and one focus at the receiver. This geometry is shown in Figure 2.3 and the regions between these constant ellipsoids are called Fresnel zone. The signal transmission between a transmitter and a receiver is considered as free-space propagation if at least 60% of the Fresnel zone is unblocked (Topcu et al., 2015).

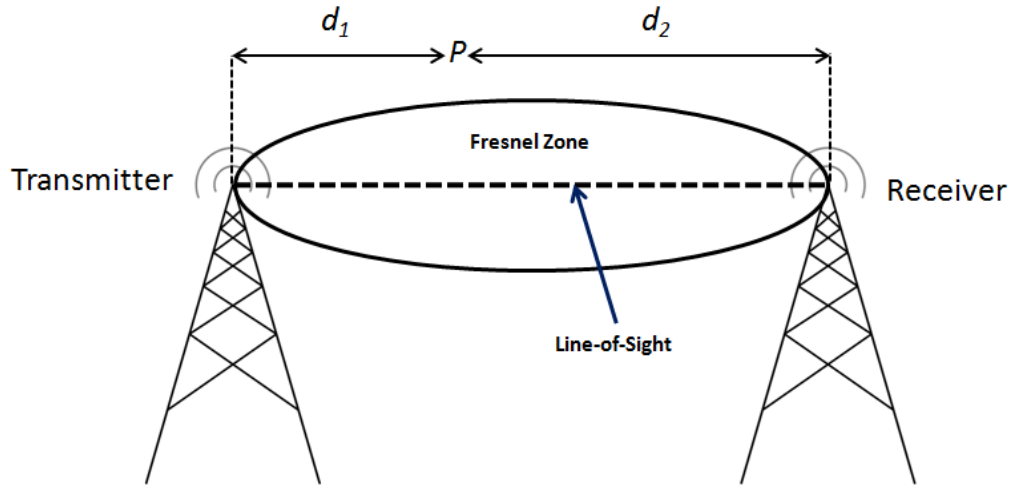


Figure 2.3: The Fresnel zone between a transmitter and a receiver

The general equation for calculating the radius of Fresnel zone at any points  $P$  in between the transmitter and the receiver is as follows:

$$F_n = \sqrt{\frac{n\lambda d_1 d_2}{d_1 + d_2}} \quad (2.2)$$

where  $F_n$  is the  $n$ th radius of Fresnel zone,  $d_1$  is the distance of  $P$  from one end,  $d_2$  is the distance of  $P$  from other end and  $\lambda$  is the wavelength of the transmitted signal. Notice that Equation 2.2 is only valid for ideal terrain height, i.e., both antennas are 90 degrees perpendicular to the LOS.

However, most of the time, the LOS between two antennas are obstructed by more than one obstacle, such signal attenuation caused by diffraction is called as diffraction loss over multiple knife edge obstructions. To compute the diffraction loss over multiple knife edge obstructions, a number of approximation techniques had been introduced. The commonly used approximation techniques were proposed by Bullington (1947), Epstein and Peterson (1953) and Deygout (1966) independently.

The effectiveness of different approximation techniques or models proposed to compute the multiple knife edge diffraction loss is a hot research topic. Ordiales et al. (1994) evaluated the effectiveness of different models with the actual field strength measurements using 120 terrain profiles in Spain. These models include Okumura model (Okumura et al., 1968), Hata model (Hata, 1980), Rec 370 (Ordiales et al., 1994), Giovanelli model (Giovanelli, 1984), Epstein-Peterson model (Epstein and Peterson, 1953), Bullington model (Bullington, 1947) and Deygout model (Deygout, 1966). Notice that Hata model is also known as Okumura-Hata model as it is extended from Okumura model. Table 2.1 summarises the comparison of the actual measurements with the diffraction loss computed using different models in terms of absolute mean, standard deviation and absolute maximum error. The

results suggested that the Epstein-Peterson model is considered as the best prediction model with the lowest absolute maximum error, second lowest standard deviation and absolute mean.

Table 2.1: Comparison between the actual measurements with the diffraction loss computed using different models

<b>Model</b>	<b>Absolute Mean (dB)</b>	<b>Standard Deviation (dB)</b>	<b>Absolute Maximum Error (dB)</b>
Okumura-Hata	7.34	9.42	34.31
Rec 370	6.80	9.00	33.20
Epstein-Peterson	3.15	7.88	24.48
Giovanelli	2.92	7.95	32.68
Bullington	8.45	10.1	36.99
Deygout	3.36	7.69	32.68

Recently, Kasampalis et al. (2015) compared the actual field-strength measurements with the diffraction loss over multiple knife edge obstructions computed using different approximation techniques proposed by Rice et al. (1967), Epstein and Peterson (1953), Deygout (1966) and Giovanelli (1984) in mountainous terrain and in long propagation path for the very high frequency (VHF) and ultra high frequency (UHF) TV broadcasting. More precisely, they placed a transmit antenna at Hortiatis (i.e., longitude of 23.099793, latitude of 40.597648 and altitude of 870 metres) with antenna height of 70 metres and another transmit antenna at Boskija (i.e., longitude of 22.67142, latitude of 41.27108 and altitude of 710 metres) with antenna height of 40 metres. Subsequently, the receive antennas were placed at Kilkis (i.e. a distance of



53.8 kilometres and 333 degrees from the transmit antenna), Evzoni (i.e. a distance of 68.8 kilometres and 321 degrees from the transmit antenna) and Doirani (i.e. a distance of 70.7 kilometres and 293 degrees from the transmit antenna) and Polykastro (i.e. a distance of 63 kilometres and 180 degrees from the transmit antenna). Subsequently, Kasampalis et al. (2015) performed field-strength measurements at these locations using a Rohde & Schwarz FSH-3 portable spectrum analyser. This study aims to investigate the accuracy of different approximation techniques used in computing the diffraction loss over multiple knife edge obstructions. The results are listed in Table 2.2. Notice that more details of setting parameters and results can be found in (Kasampalis et al., 2015). The results obtained by Kasampalis et al. (2015) showed that all the approximation techniques gave satisfactory result in predicting multiple knife edge diffraction loss.

Table 2.2: Difference of diffraction loss (in terms of dB) between actual FSH-3 measurements and different approximation techniques

Location of Transmitter	Location of Receiver	Different Approximation Techniques			
		Rice et al.	Deygout	Epstein and Peterson	Giovaneli
Hortiatis	Kilkis	2.8	-7.2	-7.2	-7.2
	Evzoni	-0.5	-1.9	0.0	0.3
	Doirani	0.9	-3.9	-5.1	-0.9
Boskija	Polykastro	3.8	-0.2	-0.3	-0.3

### 2.1.3 Diffraction Loss Computation using Epstein-Peterson Model

Due to the study conducted by Ordiales et al. (1994) and Kasampalis et al. (2015), the path loss simulation framework proposed in this project is based on the diffraction loss computed over multiple knife edge obstructions using

Epstein-Peterson path geometry method (Epstein and Peterson, 1953), Fresnel-Kirchhoff's theory and free space loss. The Epstein-Peterson method is a well-known model that approximately calculates the path geometry when there are successive shadowing hills. Figure 2.4 illustrates an example where there exist two hills in between a transmitter and a receiver. To calculate the diffraction loss, the height of  $H1$  and  $H2$  must be extracted by drawing the lines connecting transmitter to top of  $H2$ , and  $P1$  to  $P2$ .

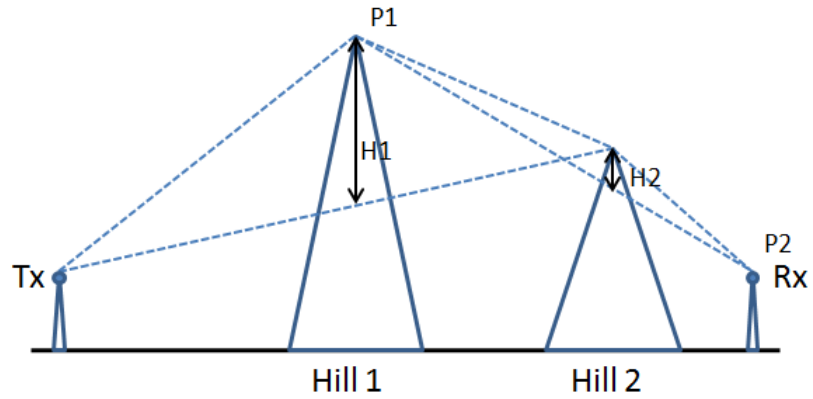


Figure 2.4: Epstein-Peterson path geometry method

Subsequently, the Fresnel-Kirchhoff diffraction parameter  $v$  (Graham et al., 2007) can be computed as follows:

$$v = h \sqrt{\frac{2(d_1+d_2)}{\lambda d_1 d_2}} \quad (2.3)$$

where  $h$  is height of the obstruction (e.g.,  $H2$  from Figure 2.4),  $d_1$  is the distance of hill from one end,  $d_2$  is the distance of hill from other end and  $\lambda$  is the wavelength of the transmitted signal.

Finally, the diffraction loss over multiple knife edge obstructions,  $G_d(v)$ , can be approximated using the Fresnel-Kirchhoff diffraction (Lee, 1997) as follows:

$$G_d(v) = \begin{cases} 0.5 - 0.62v & -0.8 < v < 0 \\ 0.5e^{-0.95v} & 0 < v < 1 \\ 0.4 - \sqrt{0.1184 - (0.38 - 0.1v)^2} & 1 < v < 2.4 \\ \frac{0.225}{v} & v > 2.4 \end{cases} \quad (2.4)$$

#### 2.1.4 Digital Elevation Model (DEM)

In designing and planning wireless communication systems, accurate propagation characteristics must be known to estimate the path loss between two antennas. Path loss simulation helps to determine the coverage of a base station (Philips et al., 2013) and overcome multipath fading (Iskander and Yun, 2002). Han et al. (2013) stated that path loss is the stationary component of the channel model affected by the shadowing in the wireless network. However existing simulators typically use simple disc shape (or free space) path loss model in which path loss increases as a function of a distance. The free space propagation model is practical and fast but terrain unaware. As mobile communication system often takes places over irregular terrain, hence terrain profile must be taken into account when estimating path loss between two antennas (Djinevski et al., 2015). The use of imprecise models can dramatically affect the simulation leading to inaccurate results and false conclusion.

The digital elevation model (DEM) (Li et al., 2004) is a digital model that representing terrain's surface. Liu and Ma (2012), Chong and Kim (2013), Djinevski et al. (2015) and Mansfield et al. (2016) independently showed that using actual terrain elevation data provides higher accuracy for wireless network simulation as wireless propagation is strongly affected by terrain profile regardless of whether it is manmade or natural. However, their work yet to be extended to the well known network simulators like NS-3 platform as such platform does not support the usage of digital terrain model in a trivial way.

In this project, the DEM data is used to improve the accuracy in estimating the path loss between a transmitter and a receiver. The resolution of DEM data can affect the accuracy of the measured terrain. The 7.5 minute DEM, that has a 10-meter resolution, is used as it is the most accurate seamless data available from United State Geological Survey (USGS), which covers an area measuring 7.5 minutes of latitude and 7.5 minutes of longitude (also known as 7.5-minute quadrangle maps). Figure 2.5 shows a sample DEM data file from USGS National Elevation Dataset repository which contains a range of different terrain information.

```

ACTON, CA-24000 LAT:: 34 22 30.0000 N LONG:: -118 7 30.0000 W SCALE:: 24000 (SDTS2DEM
v.0.018)
0.000000000000000D+000 0.000000000000000D+000 0.000000000000000D+000 0.000000000000000D+000
0.000000000000000D+000 0.000000000000000D+000 0.000000000000000D+000 0.000000000000000D+000
0.000000000000000D+000 0.000000000000000D+000 0.000000000000000D+000 0.000000000000000D+000
0.000000000000000D+000 0.000000000000000D+000 0.000000000000000D+000 0.000000000000000D+000
0.000000000000000D+000 0.000000000000000D+000 0.000000000000000D+000 0.000000000000000D+000
0.000000000000000D+000 0.000000000000000D+000 0.000000000000000D+000 0.000000000000000D+000
3.804248780075000D+006 3.852400337210000D+005 3.818110330914000D+006 3.967165930010000D+005
3.817975585800000D+006 3.965626610550000D+005 3.804114262520000D+006 7.032000000000001D+002
1.992500000000000D+003 0.000000000000000D+000 01.00000e+0011.00000e+0011.00000e-001 1
1165 2
1
0.000000000000000D+000 1.481000000000000D+003 1.516500000000000D+003 14810 14840 14865 14941 15000 15058
15095 15134
15165
3.850800000000000D+005 3.804250000000000D+006 0.000000000000000D+000 1.281800000000000D+003
1.538000000000000D+003 14805 14845 14881 14960 15018 15064 15100 15140 15170 15200 15223 15265 15315 15354
15379 15365 15380 15346 15284 15231 15184 15151 15106 15078 15034 14998 14940 14885 14828 14770 14708
14636 14576 14505 14435 14365 14320 14264 14204 14148 14058 14009 13866 13821 13740 13686 13640 13611
13578 13536 13474 13390 13318 13245 13201 13126 13050 12970 12895 12845 12818 12839 12861 12878 12900
12890 12909 12939 12961 13003 13026 13050 13088 13156 13198 13230 13264 13295 13331 13380 13414 13440
13440 13458 13458 13428 13415 13351 13285
13240
1 2 90 1

```

Figure 2.5: Sample of DEM data file

However, by including the terrain’s surface (using detailed vector data, DEM) and wireless propagation model (considering free space loss and diffraction loss over multiple knife edge) into account, the computational complexity of a path loss simulation in computing the path loss between two nodes over a large scale mobile communication network is increased substantially. Thus, one may offload compute-intensive processes of path loss simulation through the use of parallel computing technique.

## 2.2 Parallel Computing

Parallel computing is the simultaneous use of multiple compute resources to solve a computational problem. A problem is broken down into discrete parts that can be solved concurrently. These parts are further broken down to a series of instructions in order to be executed on different processors.

In our work, the parallel co-processor is used to accelerate the compute-intensive processes through parallel computing technique.

### **2.2.1 Parallel Co-Processor**

A co-processor is a microprocessor distinct from the central processing unit (CPU) that offloads specialised processing operations from the CPU. CPU is the primary component of the computer that handles all the given instructions. Parallel co-processor executes the specialised processing operations simultaneously thereby allowing it to work at optimum speed. There are a few types of parallel co-processor available in the market such as the NVIDIA graphic processor unit (GPU), Intel Xeon Phi, field-programmable gate array (FPGA) and etc.

### **2.2.2 CUDA**

Compute unified device architecture (CUDA) is a programming model and a general-purpose parallel computing platform that utilises the parallel execution capacities of NVIDIA GPU (Li et al., 2014). Meanwhile, GPU is a hardware accelerator that offloads computational tasks from the central processing unit to multi-processor platform (Fujimoto, 2015).

CUDA supports various standard programming languages such as C, C++, Java, FORTRAN, and Python. Its advantages include processing a large

volume of parallel calculations and utilising its full functions simply through the operation of NVIDIA's built-in devices (Kim et al., 2015).

A standard CUDA program requires memory to be reallocated from host memory to device memory for the kernel functions to be carried out on the GPU as shown in Figure 2.6.

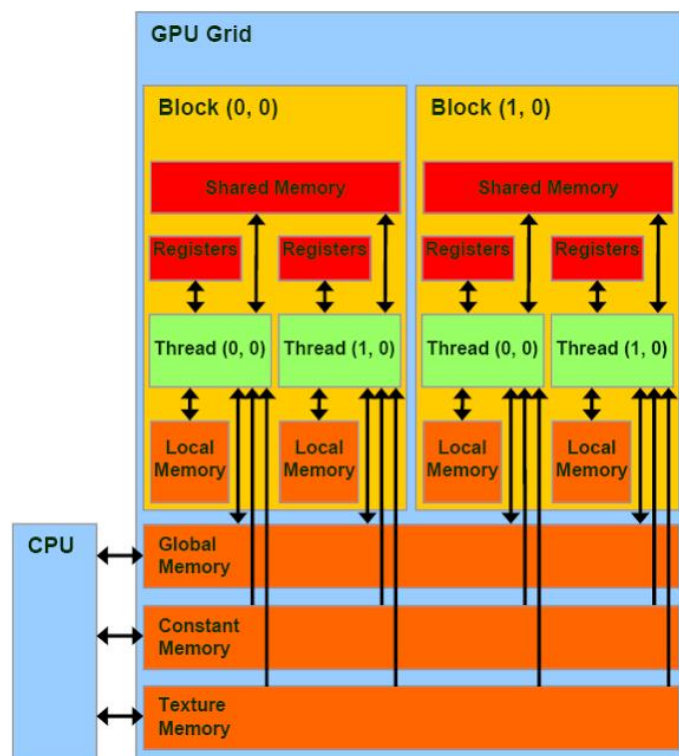


Figure 2.6: CUDA architecture (Gupta, 2013)

Cook (2013) stated that each GPU consists of a number of streaming multiprocessors that are similar to CPU cores. Each streaming multiprocessor is attached by eight or more stream processors known as CUDA core. Stream processors work in parallel of up to 32 units. CUDA splits the working load into grids of blocks. The blocks are allocated from the grid of blocks to any

streaming multiprocessor that has free slot. Each block is split into a warp or 32 threads.

Threads running on the GPU in the CUDA programming model have access to several memory regions such as the global memory, local memory, constant memory, texture memory and shared memory. Different types of memory are optimised for different memory uses. Threads within a same block can share data through shared memory (Chu et al., 2015) while the read-only constant cache memory and the texture cache memory are shared by all processors. The texture cache optimised for two-dimensional spatial locality can perform hardware interpolation and is equipped with the support of boundary value calculation (Ryoo et al., 2008).

### **2.2.3 Variants of Heterogeneous System Architecture**

Heterogeneous system architecture refers to a system that utilises more than one kind of processor. Normally, both CPU and GPU are used in this architecture to give the best of both worlds. GPU processing can perform mathematically intensive computations on substantial amount of data sets parallelly, while CPUs run the operating system and perform traditional serial tasks.

There are different types of heterogeneous system architecture that can be used to exploit GPU-based processes, such as the four different GPU-based



architectures presented by Andelfinger et al. (2011) for discrete event-based simulations as shown in Figure 2.7.

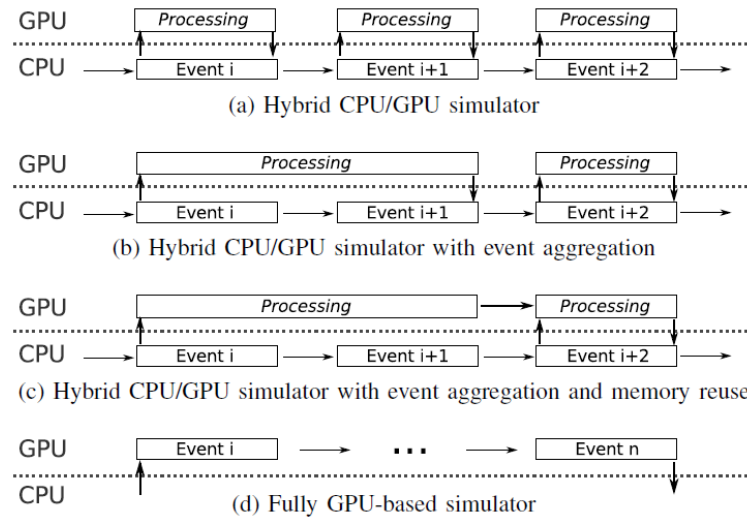


Figure 2.7: Four approaches of GPU-based discrete event simulation (Andelfinger et al., 2011)

The simplest approach of GPU-based discrete event simulation processes events sequentially in CPU as shown in Figure 2.7 (a). The input data from the compute-intensive processes is transferred into the GPU memory for parallel processing. Once the task is completed, the output data from GPU is transferred back to the main memory of host computers. A more efficient approach can be achieved, by utilising aggregation and parallel execution of identical tasks that belong to different but independent events. Multiple data transfers and context switch are reduced to only one transfer and one context switch with this approach as shown in Figure 2.7 (b). Further optimisation can be achieved when the output of one event accommodates as the input of the subsequent event which operates on the same input data. Additional data transfers can then be preserved, by reusing memory and data that has been transferred to the GPU memory as shown in Figure 2.7 (c). A

pure GPU-based simulation is envisioned, where all simulation processes reside on the GPU to reduce data transfers to the beginning and end of the simulation as shown in Figure 2.7 (d).

In order to evaluate the performance of these four architectures, a discrete event-based simulation was developed from scratch by Andelfinger et al. (2011). In the simulation, a node (transmitter) sent a packet to multiple nodes (receiver). The simplest approach showed a speedup of 1.5x independent of number of receivers, while additional event aggregation showed an overall speedup of 30.9x for 100 receivers. Lastly, by reusing memory, a speedup of 69.6x for 100 receivers was observed.

Andelfinger et al. (2011) has ported three compute-intensive signal processing algorithms (i.e., Rayleigh fading, frame synchronization and Viterbi decoding) into GPU for accurate simulation of wireless network communication. The simulations were separately implemented using an ATI Radeon HD 5870 graphic card with 1600 cores and an AMD Phenom II X6 1035T CPU with single core. Figure 2.8 shows speedup achievement of all three algorithms for the number of packets processed in parallel from 1 to 100. This finding was further applied to the NS-3 physical layer. Since transparent memory management on GPUs is not yet supported, the applied hybrid approach with event aggregation only achieved a speedup factor of 4.3x.

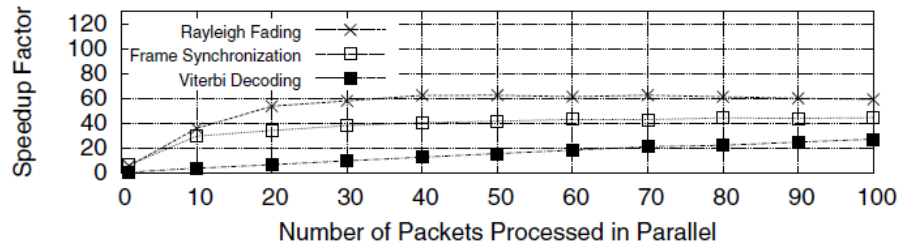


Figure 2.8: Speedup achieved by GPU-based parallelisation of individual signal processing algorithms (Andelfinger et al., 2011)

Bilel et al. (2012) presented another approach, where the CPU controls the experiment and synchronisation while the simulation is fully executed over GPU. They named their new framework as Cunetsim which enables efficient packet-level simulation for large scale scenario. This framework exploits the CPU-GPU co-simulation and provides hybrid synchronisation which maximises the efficiency as shown in Figure 2.9.

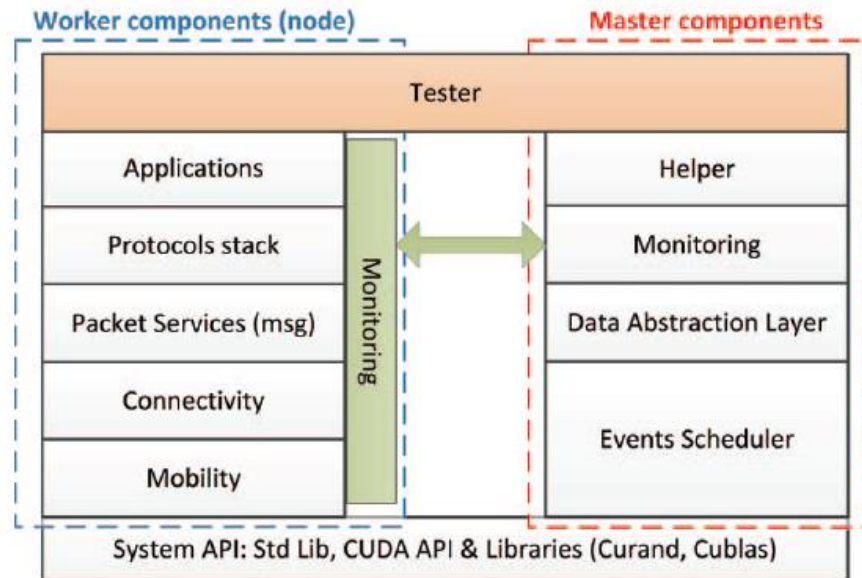


Figure 2.9: Cunetsim framework architecture and dependency (Bilel et al., 2012)

They evaluated the performance of Cunetsim using simple PCs equipped with an INTEL i7 940 CPU (4 cores with hyperthreading), 6GB of DDR3 and one GeForce 460 1GB (336 cores for GPGPU computing). Performance results showed that the execution time could be radically improved when GPU parallelism is used to carry out the simulation. In particular, Cunetsim achieved up to 260x faster as compared to existing simulators for large scale scenario. Their results also suggested that the existing simulator could be further improved through multi-core parallelism.

Ivey et al. (2016) incorporated the ability to call NVIDIA CUDA kernels from within simulated NS-3 nodes by extending the functionality of the Direct Code Execution framework of NS-3 that allows real-world network applications to be executed within the stimulated NS-3 environment. They studied three different options available for communicating between the stimulated nodes and the GPU in their framework. One examined the simple native redirection of CUDA calls in Direct Code Execution directly to the CUDA driver library. The other two exploited cloud computing with the gVirtuS backend framework to virtualise the usage of CUDA API calls, which allows an instanced virtual machine to access GPUs in a transparent way, using the Unix-based file descriptor that handles the messages or through tap bridges that act as real network interfaces.

## 2.2.4 Existing GPU Implementations of Different Wireless Propagation Models

Different features used in different wireless network simulations are summarised in Table 2.3 where GPU, DEM and SF denote graphical processing unit, digital elevation model and simulation framework respectively. Notices that the models proposed by Djinevski et al. (2015) and in this project are the only work that exploit GPU, DEM and SF simultaneously. Since some of the work has been discussed in the previous sections, this section focuses on the GPU implementations of different wireless propagation models.

Table 2.3: Summary of different wireless network simulations

No	Source	GPU	DEM	SF
1	Valcarce et al., 2008	x	x	
2	Abdelrazek et al., 2009	x		x
3	Andelfinger et al., 2011	x		x
4	Mittag et al., 2011			x
5	Bilel et al., 2012	x		x
6	Liu and Ma, 2012		x	
7	Chong and Kim, 2013		x	
8	Djinevski et al., 2015	x	x	x
<b>9</b>	<b>Proposed Work</b>	<b>x</b>	<b>x</b>	<b>x</b>

Valcarce et al. (2008) performed finite-difference time-domain (FDTD) analysis of propagation in a city using GPU. The COST 213 Munich city test environment was selected in the simulation. As FDTD tried to simulate an unbounded region in an area with restricted size, an absorbing boundary condition (ABC) was applied by using the convolutional perfectly matched

layer (CPML) to avoid reflections on the border. The FDTD-CPML method was simulated using an AMD Athlon 64 X2 Dual Core processors 4600++ at 2.41GHz and different GPUs respectively. As compared to the simulation conducted using CPU with dual core processors and MATLAB, a speedup of 100x and 540x were achieved using a standard off-the-shelf GPU (i.e., GeForce 8600M GT) and a high performance computing card (i.e., TESLA C870) respectively as shown in Figure 2.10. The results indicate that the CUDA technology realised the simulation with high complexity of time. However, proper calibration and configuration of different FDTDs are needed for accurate result. Hence, DEM data based realistic simulation without additional configuration and calibration is preferred.

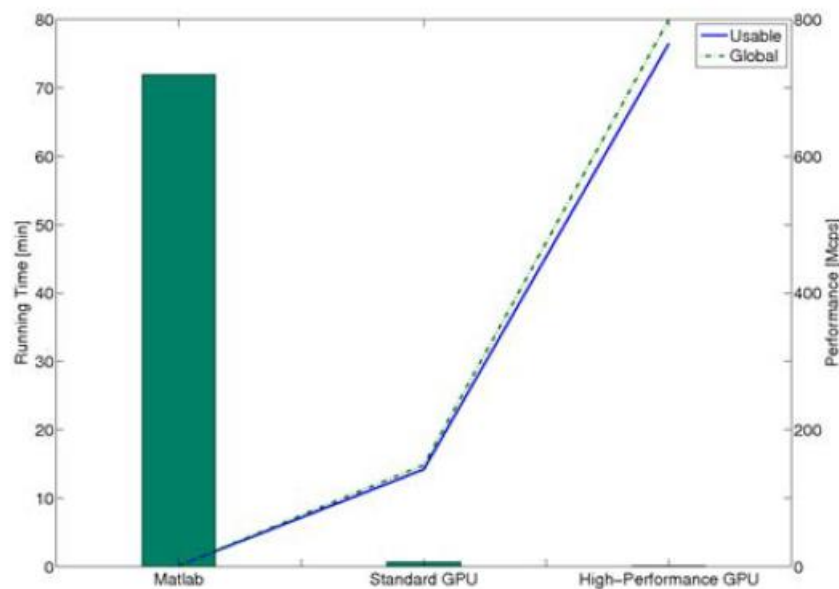


Figure 2.10: Running time (bar) on three different platforms with performance (line) in terms of megacells per second for the FDTD-CPML method (Valcarce et al., 2008)

Abdelrazek et al. (2009) presented an approach that offloads the wireless multi-path fading channel computation to an inexpensive massive

parallel GPU. These computations were embedded into a simple event-driven network simulation using IKR-Simlib and NVIDIA GeForce 8800GTS (500MHz core clock, 96 stream processors, and 320MB of memory). Their work focuses on a particular statistical discrete-time model (Hoeher, 1992). Their proposed system showed a speedup of about 30x as compared to a regular PC hardware implementation as shown in Figure 2.11. They integrated their model into a simulator framework. Different with their proposed model, a deterministic model is used in this project to produce results with higher accuracy by considering consider the obstacles blocking the propagation path.

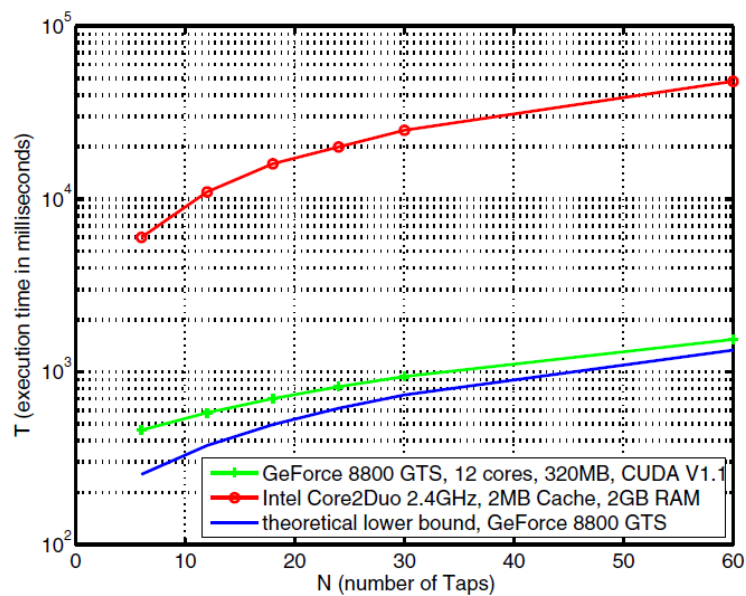


Figure 2.11: Execution time of Hoeher channel model in CPU and GPU (Abdelrazek et al., 2009)

Djinevski et al. (2015) presented a parallel implementation of the Durkins radio propagation algorithm using TESLA C2070 GPU. Their proposed system showed a speedup of 118x as compared to a sequential implementation. They extended the system to NS-2 simulator and a speedup of 67x was achieved with a cache structure that gather all possible propagation

pairs for each transmitters within the terrain. In their simulation, free space propagation model is used when the Fresnel zone of the T-R pair is unblocked; otherwise, their model compute the path loss based on single diffraction edge [Vuckovik et al., 2011]. As compared to Djinevski et al. (2015) work, this project considers the path loss based on multiple diffraction edges using Epstein and Peterson method to produce path loss computation with higher accuracy.

### **2.2.5 NS-3**

NS-3 is a discrete-event network simulator that is widely used free software licensed under the GNU GPLv2 license. Similar with its predecessor NS-2, C++ programming language is used in both NS-2 and NS-3 to perform the simulation. However, NS-3 does not use oTcl scripts to control simulations (Chengetanai and O'Reilly, 2015) and thus is free from the problems introduced by the combination use of C++ and oTcl in NS-2 (Amoretti et al., 2013).

NS-3 is suitable to accommodate parallel computing as its modular design allows users to add, modify, or replace modules based on requirements (Yip and Asaduzzaman, 2014). Stojanova et al. (2013) proposed a NS-3 simulation of Durkin's terrain aware radio wave propagation model. Such simulation showed a significant performance increase as compared to the NS-2 simulation due to the improved memory management, the use of aggregation



system which prevents unneeded parameters from being stored and the eliminations of overhead introduced from oTcl scripting.

## CHAPTER 3

### METHODOLOGY

The proposed path loss simulation framework is explained in three different phases for ease of understanding. Each process involved in the path loss simulation framework is explained in details using the data flow diagram. Lastly, the experimental setup and data selection for the proposed simulation are presented.

#### 3.1 Proposed Path Loss Simulation Framework

The proposed path loss simulation framework is divided into three phases denoted as A1, A2 and A3 respectively as follows:

- i. A1: Disc model with conventional CPU architecture
- ii. A2: Terrain aware model with conventional CPU architecture
- iii. A3: Terrain aware model with heterogeneous system architecture

These phases are developed to capture the changes in computational complexity of the proposed path loss simulation framework.

##### 3.1.1 A1: Disc Model with Conventional CPU Architecture

Disc model is also known as free space loss model where the path loss simulation between two antennas is a function of distance. More precisely,

free space loss model is an isotropic radio model that implies signal attenuates in all directions with similar magnitude over distance which does not hold well in comparison to real measurement (Zhou et al., 2004). Even though free space loss model does not consider terrain profiles in path loss computation between two antennas, the free space loss model is still implemented to determine the differences in terms of computation complexity and the path loss between two antennas as compared to the terrain aware model.

Figure 3.1 shows the flow chart of free space loss model implemented using conventional CPU architecture framework. Computation of free space loss is carried out whereby the distance between a transmitter (T) and a receiver (R) is calculated based on the T-R pair coordinates and the frequency of a transmitted signal is pre-fixed at the beginning of the simulation.

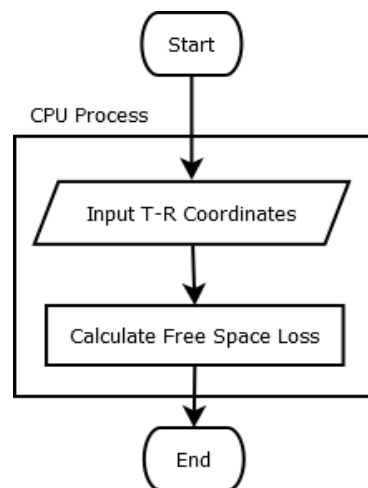


Figure 3.1: Flow chart of free space loss model using conventional CPU architecture

### **3.1.2 A2: The Terrain Aware Model Implemented using Conventional CPU Architecture**

Free space loss model is only ideal when there are no obstacles blocking the path between the T-R pair as the obstacles may cause reflection or refraction of the transmitted signal. To provide a more realistic real-world path loss computation between two antennas, a terrain aware model based on Epstein-Peterson method, Fresnel-Kirchhoff's diffraction theory and free space loss is incorporated. More details of the terrain aware model can be referred to Section 2.1.3.

To obtain the terrain profile between the T-R pair, the proposed terrain aware model needs to analyse, extract and form the terrain/path profile between the T-R pair from the USGS DEM data. Additionally the proposed terrain aware model is ported to NS-3 for enabling future studies on cross-layer optimisation. NS-3 is chosen as it is one of the most popular open source network simulators. Besides, NS-3 is compatible with CUDA platform (which will be described in Section 3.1.3) which utilises C++ as well.

Figure 3.2 shows the flow chart of terrain aware model implemented using conventional CPU architecture framework. First, the terrain information between two antennas is extracted from the USGS DEM data file where the coordinates of two antennas (i.e. the T-R pair) are provided. Subsequently, the extracted terrain information can be used to form the terrain/path profile and the Fresnel zone between the T-R pair. If 60% of the Fresnel zone between the

T-R pair is unblocked, the terrain aware model will compute the path loss using free space loss model (as stated in Equation 2.1); otherwise, the terrain aware model will proceed to analyse and search the terrain peaks (i.e. multiple obstacles) sitting in between the T-R pairs that block the line-of-sight between the T-R pair. Finally, the path loss computation is performed as the addition of two different losses, i.e., Fresnel-Kirchhoff diffraction loss and free space loss.

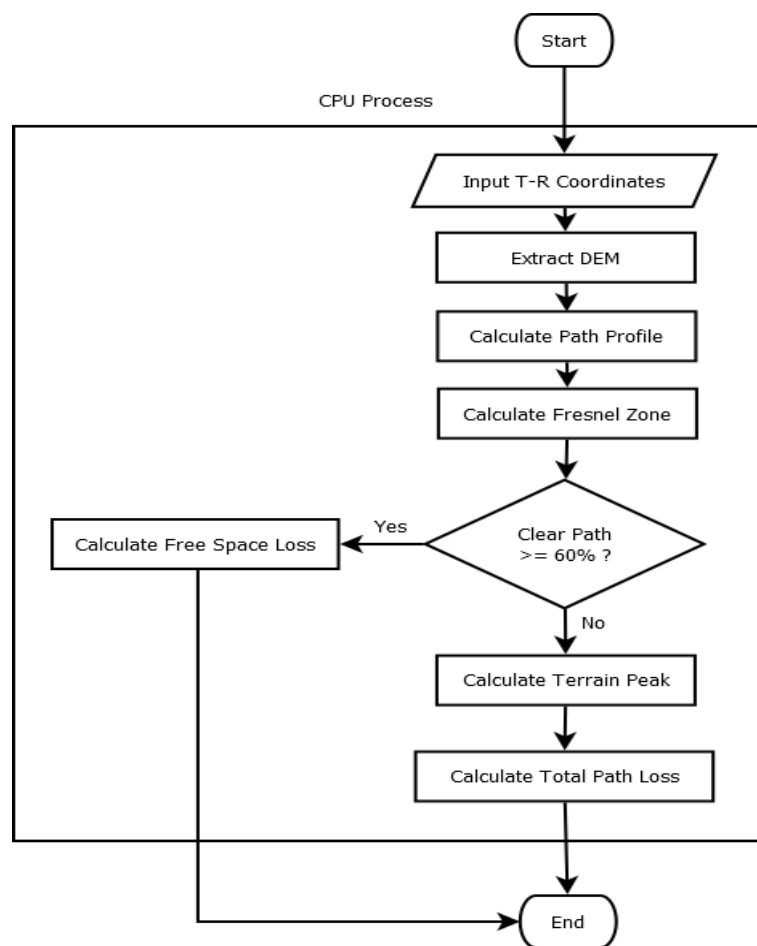


Figure 3.2: Flow chart of terrain aware model using conventional CPU architecture

### **3.1.3 A3: The Terrain Aware Model Implemented using Heterogeneous System Architecture**

The terrain aware model provides a more realistic path loss result at the cost of higher time complexity in extracting terrain information for large scale communication network. To solve the aforementioned issue, some compute-intensive processes of the terrain aware model are parallelised and offloaded to general-purpose graphics processing unit (GPU).

Figure 3.3 shows the flow chart of terrain aware model implemented using heterogeneous system architecture. As compared to the terrain aware model implemented using conventional CPU architecture framework, two compute-intensive processes are offloaded to GPU, i.e., terrain mapping and path profile calculation.

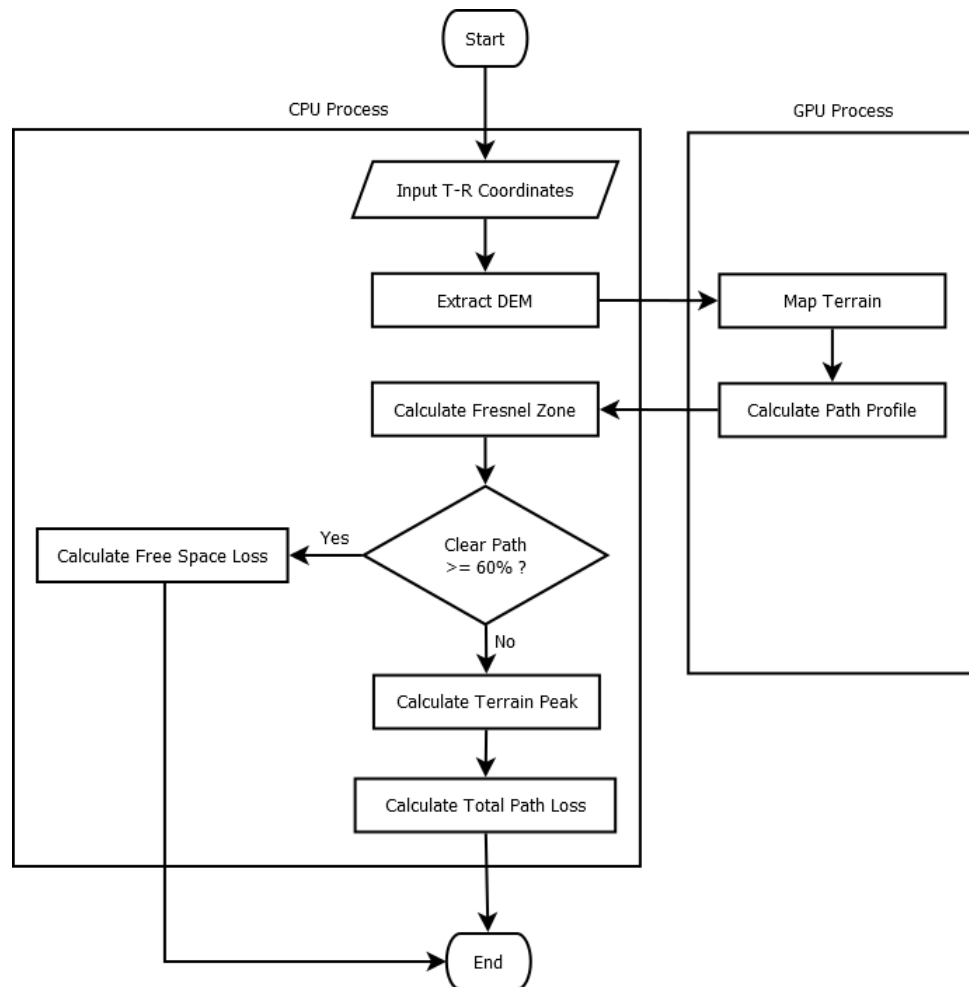


Figure 3.3: Flow chart of terrain aware model using heterogeneous system architecture

Terrain mapping is a process in which the system ports the extracted terrain data into GPU texture cache memory. The reasons for using GPU texture memory are two-fold. First, the data reading time is faster since the texture memory is a cache memory. Second, the simulation achieves faster speed by exploiting the additional feature of the texture memory where the linear interpolation of data which is needed in the previous CPU architecture can be computed easily. Subsequently, with the terrain data extracted into GPU texture cache memory, path profile calculation can be parallelly executed by different co-processors of GPU. These techniques can be easily extended to

any other terrain aware model, such as the models proposed by Deyout, Giovanelli and Bullington.

### **3.2 Detailed Data Flow of The Proposed Path Loss Simulation Framework**

A level 0 data flow diagram (DFD) shows high level details of how does the data is moved from one process to another in the proposed simulation and explains how does the input data is transformed and manipulated through a sequence of functional transformations. The underlying details of each process and its data processing are described in the following sub sections.

#### **3.2.1 Data Flow of The Disc Model Implemented using Conventional CPU Architecture**

Based on the flow chart of disc model implemented using conventional CPU architecture, Figure 3.4 shows the details of free space loss computation. This process only requires one input which is the T-R coordinates input from the user.



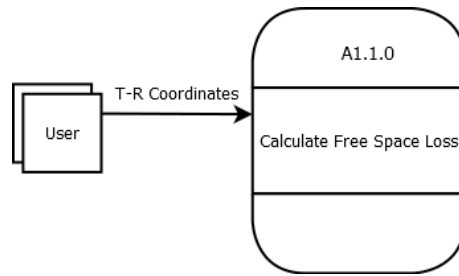


Figure 3.4: DFD level 0 for the disc model

**a) Process A1.1.0: Calculate Free Space Loss**

The system first calculates the antenna distance between the T-R pair through Pythagoras' theorem (Maor, 2007) and T-R pair coordinates. Subsequently, with the knowledge of the antenna distance, the free space loss can be computed using Equation 2.1 as 2.4 GHz frequency is selected in the beginning of the simulation.

**3.2.2 Data Flow of The Terrain Aware Model Implemented using Conventional CPU Architecture**

To provide a more realistic real-world path loss computation between two antennas, a terrain aware model based on Epstein-Peterson method, Fresnel-Kirchhoff's diffraction theory and free space loss is incorporated. Moreover, DEM data with high resolution to model terrain' surface is used throughout this simulation.

Generally, there are two main flows of data in the system, which are the pre-processing DEM data process and the terrain aware path loss computation. The data flow of this simulation is shown in Figure 3.5.

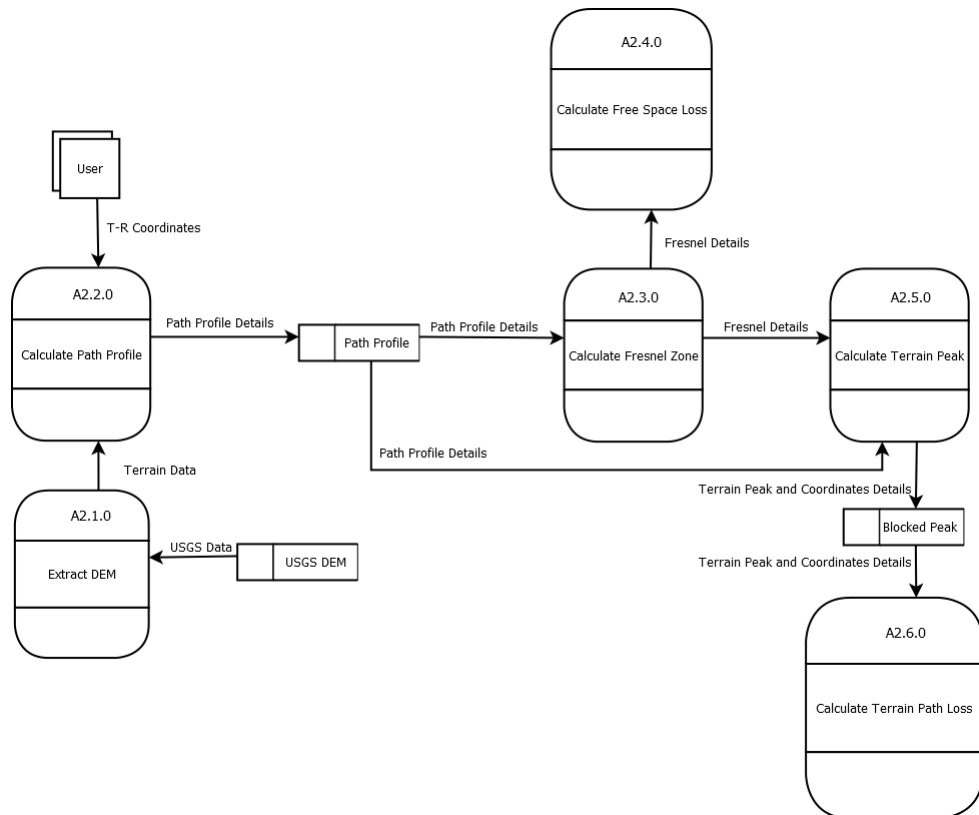


Figure 3.5: DFD level 0 for the terrain aware model implemented using conventional CPU architecture

**a) Process A2.1.0: Extract DEM**

Process A2.1.0 is a DEM extraction process to extract out the specific details of a terrain's surface, i.e. coordinates ( $x$ ,  $y$ ) and height elevation ( $z$ ) from the USGS DEM data using the Geospatial Data Abstraction Library (GDAL) that reads and writes raster and vector geospatial data formats.

**b) Process A2.2.0: Calculate Path Profile**

To generate a path profile for a T-R pair in Process A2.2.0, the process first determine the antenna distance between the T-R pair using Pythagoras' theorem and T-R pair coordinates selected by the user. Subsequently, the path profile is calculated based on the computed antenna distance and the general equation of straight line as shown in Equation 3.1 where the gradient  $m$  and the constant  $c$  (i.e. the  $y$ -intercept) that are computed using a pair of two coordinates.

$$y = mx + c \quad (3.1)$$

However, the points given in DEM data are only expressed in  $x$ - and  $y$ -direction. Besides, not all the points that fall on a path profile between the T-R pair are given in DEM data. Thus, interpolation must be performed to extract the height of the terrain that falls on the path profile between T-R pair in completing the path profile in  $x$ -,  $y$ - and  $z$ -direction. More precisely, given a point that falls on the path profile but is not in the DEM data, the developed interpolation function first located the four nearest points that can be found in the DEM data. Based on these four nearest points, the terrain height of the given point can be extracted.

**c) Process A2.3.0: Calculate Fresnel Zone**

After computing the path profile, the path/terrain profile between the T-R pair is analysed in Process A2.3.0 to check whether there exist any obstruction that may block the line-of-sight between the T-R pair.

The signal transmission between a transmitter and a receiver is considered as free-space propagation if at least 60% of the Fresnel zone is unblocked. Thus, the Fresnel zone between the T-R pair must first be drawn. However, the Fresnel zone equation (i.e. Equation 2.2) is only valid for ideal terrain height as explained in Section 2.1.2, i.e., both antennas are 90 degrees perpendicular to the line-of-sight between the T-R pair.

In the real world, both antennas are not likely to be located at the same terrain height. To solve this issue, the proposed simulation framework performs the matrix rotation such that both antennas are virtually located at the same height as shown in the Figure 3.6. More precisely, to rotate the path profile clockwise when the height of the left antenna is lower or rotate counter-clockwise when the left node antenna is higher.

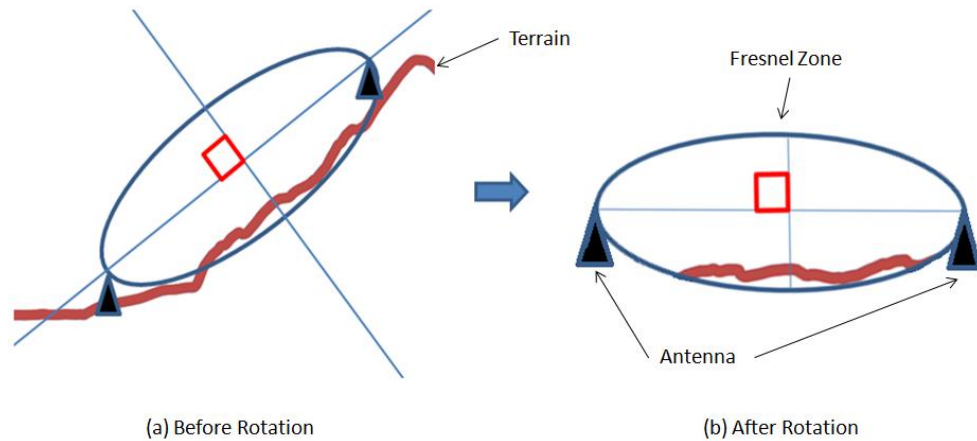


Figure 3.6: Rotating path profile

Subsequently, Process A2.3.0 performs a search to check whether any terrain blocks the line-of-sight between the T-R pair using the DEM data. If at least 60% of the Fresnel zone is unblocked, the simulation will proceed to the free space loss computation; otherwise, the simulation will proceed to further analyse the path profile between the T-R pair.

**d) Process A2.4.0: Calculate Free Space Loss**

This process is the same process explained in Process A1.1.0, thus the details are ignored here.

**e) Process A2.5.0: Calculate Terrain Peak**

Generally, there exist many peaks and troughs between the transmitter and the receiver. However, not all these peaks and troughs will affect the path loss computation between the T-R pair.

Firstly, Process A2.5.0 locates the highest peak existing in the path profile between the T-R pair. The coordinate of such highest peak is then stored. By using the highest peak as the reference point, the process performs the search both to the left and right directions to find out the other peaks that block the line-of-sight between one of the antennas and the highest peak. If such a peak is found, then this peak will be considered as the new highest peak of that section. Similarly, the coordinate of such new highest peak is stored. This process is iterated until all the peaks that affect the line-of-sight propagation are identified. For an example, Figure 3.7 shows the detailed pseudocode in searching any obstructing peaks to the left direction.

```

REPEAT-UNTIL pathProfileClear is TRUE
    SET pathProfileClear is TRUE
    FOR each currentLocation of the total from startLocation to endLocation
        COMPUTE terrain elevation of ideal line-of-sight as LOS
        COMPUTE terrain elevation of currentLocation as currentTerrainElevation
        IF currentTerrainElevation is higher than LOS and higher than highestPeak THEN
            SET currentTerrainElevation as highestPeak
            SET currentLocation as endLocation
            SET pathProfileClear is FALSE
        INCREMENT totalPeak
    STORE endLocation and highestPeak
RETURN totalPEAK

```

Figure 3.7: Pseudocode for searching obstructing peaks.

After all the obstructing peaks from different sections of the path profile is detected, all the stored coordinates and terrain information are further passed into a smoothing function to eliminate shallow irregularities where the height differences that are smaller than the wavelength (i.e.

wavelength = speed of light/frequency =  $3 \times 10^8 \text{ ms}^{-1} / 2.4 \text{ GHz} = 0.125 \text{ m}$ ) will be ignored as the negligible height differences will not affect the path loss between the T-R pair.

**f) Process A2.6.0: Calculate Terrain Path Loss**

The final phase of the simulation is to calculate the total path loss using the terrain aware model. There are four steps in calculating the total path loss in this terrain aware model.

The first step is to calculate the path geometry using the Epstein and Peterson method described in Section 2.1.3 based on the obstructing peaks identified in Process A2.5.0, Figure 3.8 shows the detailed pseudocode in obtaining all the height difference of the obstructing peaks.

```
FOR each currentPeak of the blockPeaks
    COMPUTE terrain elevation of ideal line-of-sight as LOS
    COMPUTE terrain elevation of currentPeak as currentTerrainElevation
    COMPUTE currentTerrainElevation subtract LOS as currentHeightDifference
    STORE currentHeightDifference
```

Figure 3.8: Pseudocode for calculate the height difference of the obstructing peaks.

Secondly, the Fresnel-Kirchhoff diffraction parameter  $\nu$  (i.e. Equation 2.3) can be computed based on the aforementioned height differences. Then, the diffraction loss  $G_d(\nu)$  is approximated using the Fresnel-Kirchhoff diffraction (i.e. Equation 2.4). Figure 3.9 shows the pseudocode that calculates the diffraction loss based on obstructing peaks.

```

FOR each currentPeak of the blockPeaks
    COMPUTE Fresnel-Kirchhoff parameter as FresnelKirchhoffParameter
    IF FresnelKirchhoffParameter is more than -0.8 and more than or equal 0 THEN
        diffractionGain = 0.5-(0.62*FresnelKirchhoffParameter)
    ELSE IF FresnelKirchhoffParameter is more than 0 and more than or equal 1 THEN
        diffractionGain = 0.5*(exp(-0.95*FresnelKirchhoffParameter))
    ELSE IF FresnelKirchhoffParameter is more than 1 and more than or equal 2.4 THEN
        diffractionGain = 0.4-sqrt(0.1184-pow(0.38-(0.1*FresnelKirchhoffParameter),2))
    ELSE IF FresnelKirchhoffParameter is more than 2.4 THEN
        diffractionGain = 0.225/FresnelKirchhoffParameter
    ELSE FresnelKirchhoffParameter
        diffractionGain = 0
    COMPUTE total path loss

```

Figure 3.9: Pseudocode for computing the diffraction loss.

Thirdly, free space loss (i.e. Equation 2.1) is calculated. Finally the total path loss is computed by summing up the diffraction loss and the free space loss.

### 3.2.3 Data Flow of The Terrain Aware Model Implemented using Heterogeneous System Architecture

The computational complexity of the simulation increases substantially when taking terrain profile into account in computing path loss between a transmitter and a receiver. This situation becomes worse when a suitable spot to locate the receiver must be found for a large-scale communication network given a transmitter located at a fixed location.



To counter the higher time complexity introduced in the terrain aware model with the use of DEM data with high resolution, the final phase of the proposed simulation is to swift the CPU architecture to heterogeneous system architecture that allows compute-intense processes to be offloaded to GPU for parallel processing as shown in Figure 3.10.

The most important task in designing the terrain aware model using heterogeneous system architecture is to identify and parallelise compute-intense processes. Notice that not all the processes can be parallelised and are not suitably ported to GPU. Besides, one may consider the extra introduced time complexity in transferring the data from CPU to GPU as GPU can process data stored in GPU memory only. Due to these reasons, the process “Map Terrain” is added to transfer the data from CPU memory to GPU texture memory while the process “Calculate Path Profile” is parallelised and loaded to GPU. Since the remaining processes are similar with those describe in Section 3.2.2, the remaining processes will be ignored in this section.

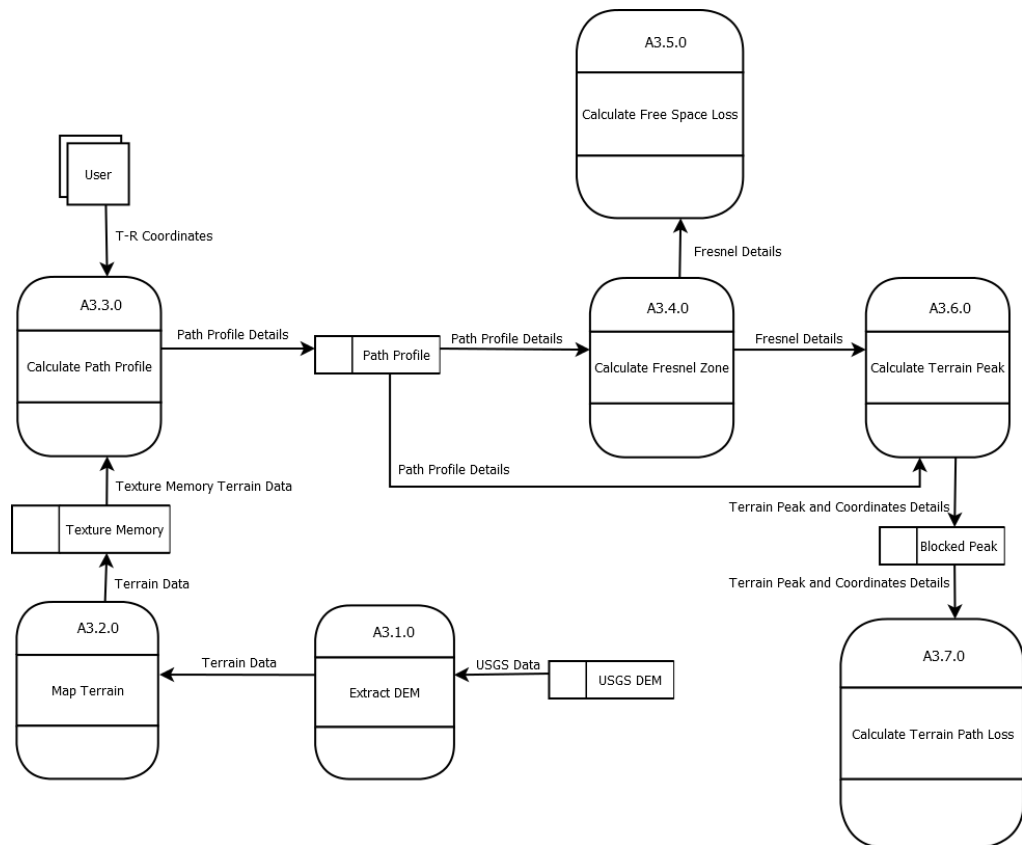


Figure 3.10: DFD level 0 for the terrain ware model implemented using heterogeneous system architecture

a) **Process A3.2.0: Map Terrain**

Process A3.2.0 stores the extracted terrain data into GPU texture memory for faster data accessing. Besides, GPU equips interpolation feature which helps in extrapolating data beyond specific point locations. Thus, the developed interpolation function in conventional CPU architecture is no longer needed in the terrain aware model implemented using heterogeneous system architecture. Remark that the DEM data extraction and terrain mapping are only needed to be performed once if the same data are used throughout the simulation.

### **b) Process A3.3.0: Calculate Path Profile**

Since interpolation feature is available for GPU, thus the developed interpolation function in Section 3.2.2 is no longer needed. The use of GPU texture memory and parallelised path profile calculation reduce the overall time in the simulation. The number of blocks used depends on the distance between the T-R pair with each CUDA thread performs calculation of one metre point. For example, assume that each CUDA block consists of 128 CUDA threads, then a total of eight blocks is needed to calculate the path profile with the length of one kilometre (i.e.,  $1000/128 = 7.8 \approx 8$ ).

### **3.3 Experimental Setup**

The conventional CPU architecture system is developed using an Intel® Core™ i7-4500U CPU @ 1.80GHz with 4GB RAM, while the heterogeneous system architecture uses a consumer grade GPU (GT 720M) besides the aforementioned CPU. All the simulations are written in C++ programming language using Microsoft Visual Studio 2012 as the integrated development environments to integrate into NS-3. Meanwhile, GPU-accelerated processes are built using CUDA Toolkit 6.5. Besides, GDAL and the geographic information software Global Mapper 16 are used to extract and study DEM data respectively. Lastly, MATLAB and Microsoft Excel are used to analyse and present the results.

### **3.4 Data Selection and Analysis**

Both point-to-point model and mobility model are used to check the performance of our simulations in different scenarios. In point-to-point model, connection link are established between two fixed nodes representing a transmitter and a receiver respectively; in mobility model, one of the nodes is treated as a mobile node that keep moving from one position to another position. More precisely, modified random walk model is used in the proposed simulations to realistically mimic the behaviours of mobile nodes by means of mathematical model.

The random walk mobility model was developed to mimic erratic movement as many entities move in unpredictable ways naturally (Davies, 2000). Each movement in the random mobility model used occurs with constant distance or time interval where a new direction and speed are calculated in the end of travelling (Camp et al., 2012). If the simulated mobile node reaches the simulation boundary, it will bounces off the simulation border.

In this project, the modified random mobility model is used where it excludes the speed variable. Notice that the speed of mobile nodes will not affect the judgement whether a path between the transmitter and a receiver is free from obstacles. Figure 3.11 shows the travelling pattern of a mobile node using a 2D random walk mobility model.

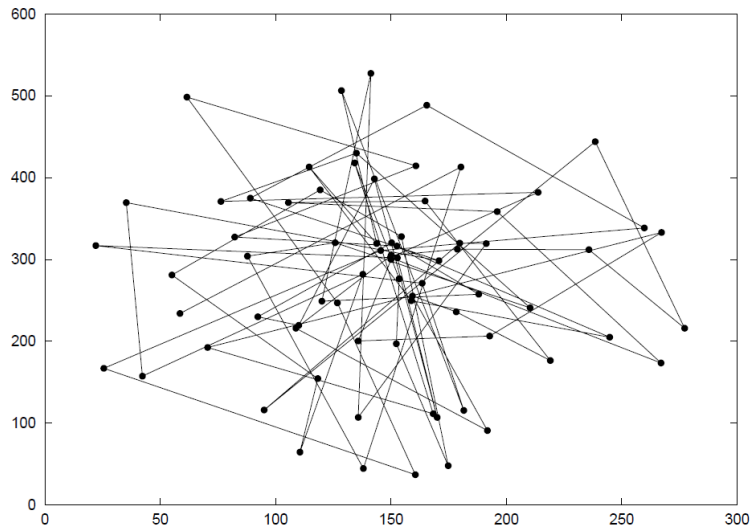


Figure 3.11: Travelling pattern of a mobile node using 2D random walk mobility model (Camp et al., 2012).

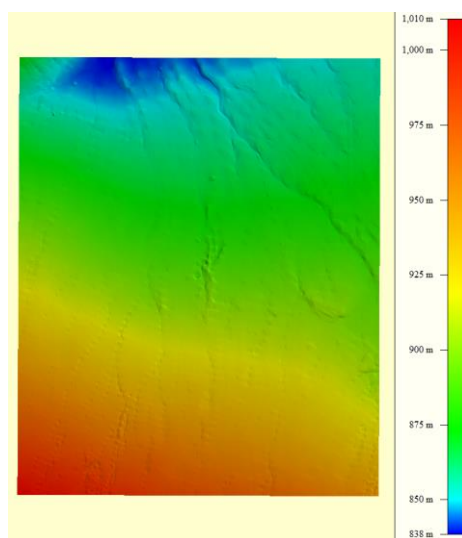
To show the effectiveness of the proposed simulations, seven different DEM data representing seven different areas are chosen, i.e., Acton, Adelanto, Amboy Crater, Amsterdam, Arrowhead Butte, Caples Lake and Jersey City. These seven 7.5-minute DEM data with a 10-meter resolution are used as it is the most accurate seamless data available from USGS. These seven chosen DEM data (as visualised in Figure 3.12 and Figure 3.13) consist of flat terrains and mountainous terrain that cannot be mapped with mathematical model. Extensive experiments are conducted using these seven DEM data to check whether different types of terrain may affect the computation of path loss between a transmitter and a receiver.

Two different types of terrain are categorised based on the terrain elevation difference (i.e the difference between the highest point and the lowest point): low terrain elevation difference that is lesser than 500m and high terrain elevation difference that is more than 500.

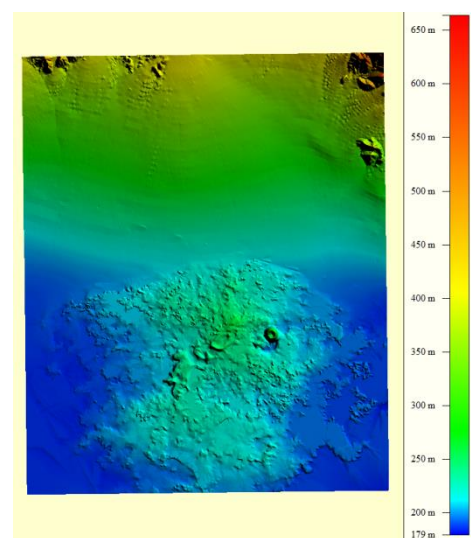
We have basically categorised two types of terrain which is the terrain that has low terrain elevation difference less than 500m and the terrain with high terrain elevation difference more than 500m. Table 3.1 shows that four locations (i.e. Adelanto, Amboy Crater, Amsterdam and Jersey City) are of low terrain elevation difference and the remaining three locations (i.e. Acton, Arrowhead Butte and Caples Lake) are of high terrain elevation difference.

Table 3.1: Two types of terrain profile

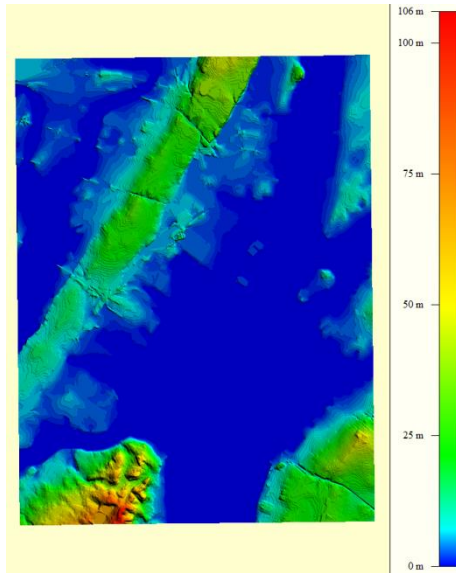
Terrain Profile		Terrain Elevation (m)		
		Highest	Lowest	Difference
<b>Low</b>	Adelanto	1010	838	173
	Amboy Crater	664	179	485
	Jersey City	106	0	106
	Amsterdam	319	73	246
<b>High</b>	Arrowhead Butte	6405	5709	696
	Acton	1992	703	1289
	Caples Lake	10382	6576	3806



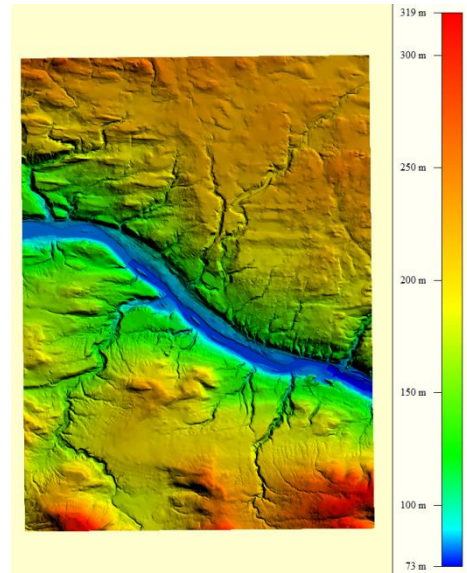
(a) Adelanto



(b) Amboy Crater

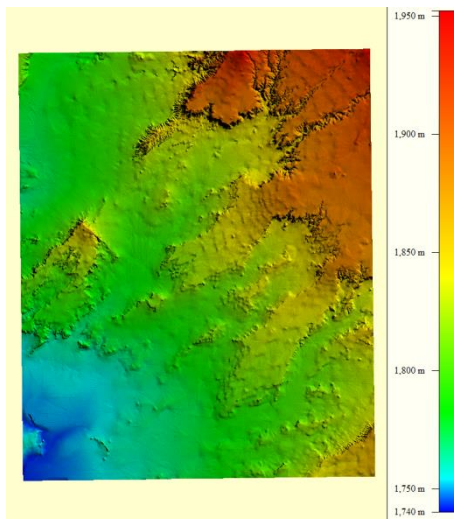


(c) Jersey City

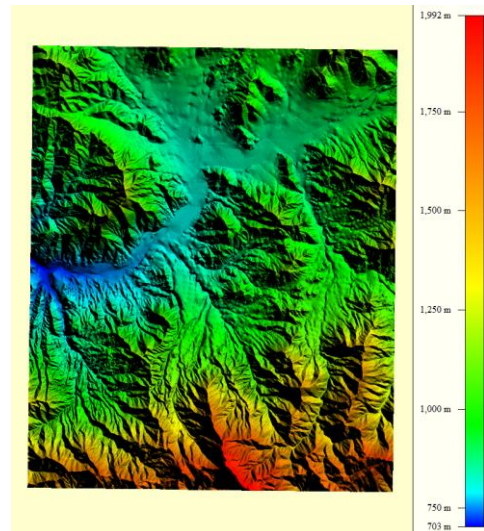


(d) Amsterdam

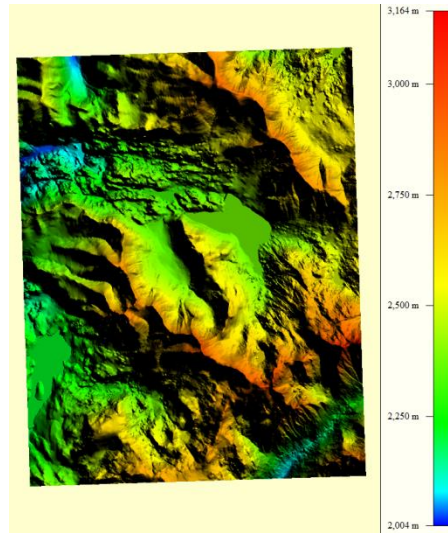
Figure 3.12: Low terrain elevation difference profiles



(a) Arrowhead Butte



(b) Acton



(c) Caples Lake

Figure 3.13: High terrain elevation difference profiles



## CHAPTER 4

### RESULTS AND DISCUSSION

To highlight the importance of terrain profile in affecting the path loss between a transmitter and a receiver, different analyses based on two different models, i.e. the modified random walk model and point-to-point model are conducted.

#### 4.1 Modified Random Walk Mobility Model

This section analysis is based on a modified random walk mobility model on our proposed terrain model which takes 100 simulations for 9 different linear distances ranging from lowest 1km to 9 km difference, for a total of 900 simulations run per terrain profile.

##### 4.1.1 Preliminary Analysis on Terrain Profiles

Firstly, terrain profile analysis is conducted by running 100 simulations for nine different linear distance ranging from 1km to 9km which result a totally of 900 simulations per one terrain profile. Figure 4.1 shows the percentage of free space path and blocked path for each selected terrain profile. The results indicate that Amboy Crater, Jersey City and Adelanto have high percentage of free space path ranging from 26% to 56%. Meanwhile, Acton,

Caples Lake and Arrowhead Butte have low percentage of free space path ranging from 7% to 8% only.

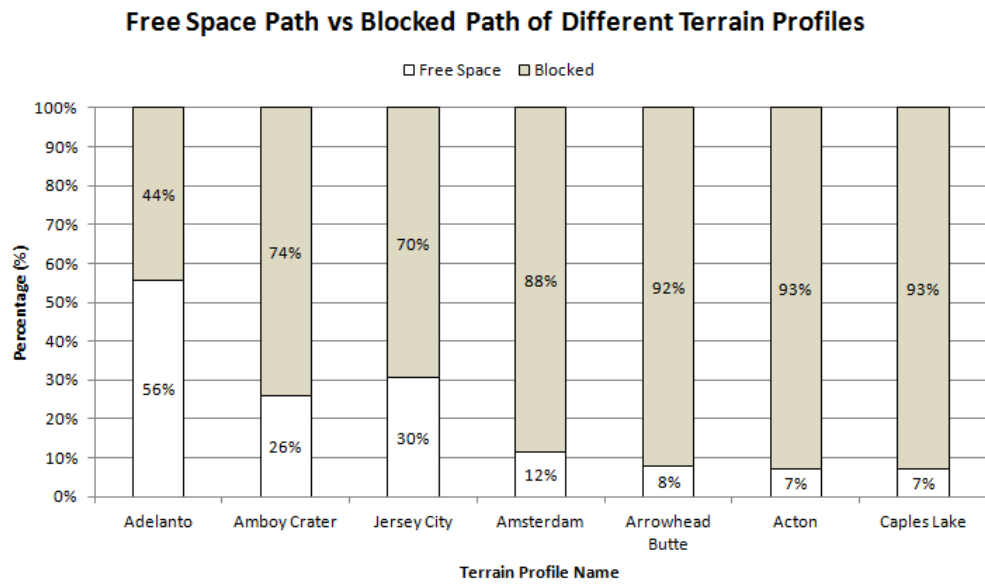


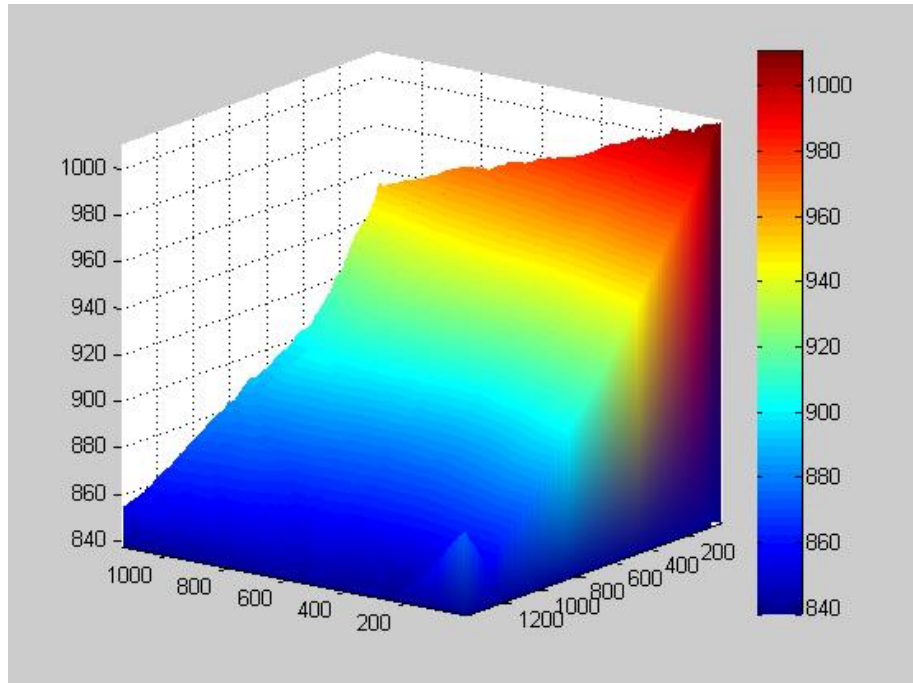
Figure 4.1: Free space path versus blocked path in different terrain profiles

To investigate whether the percentage of detected free space path is directly affected by the terrain elevation difference of terrain profile, the terrain elevation difference, mean and standard deviation of each terrain profile are calculated and summarised in Table 4.1. Higher terrain elevation difference and standard deviation generally contributes to lower percentage of detected free space path; however such observations do not hold on certain cases, such as Jersey City which has the lowest terrain elevation difference and standard deviation is not with the highest percentage of detected free space path.

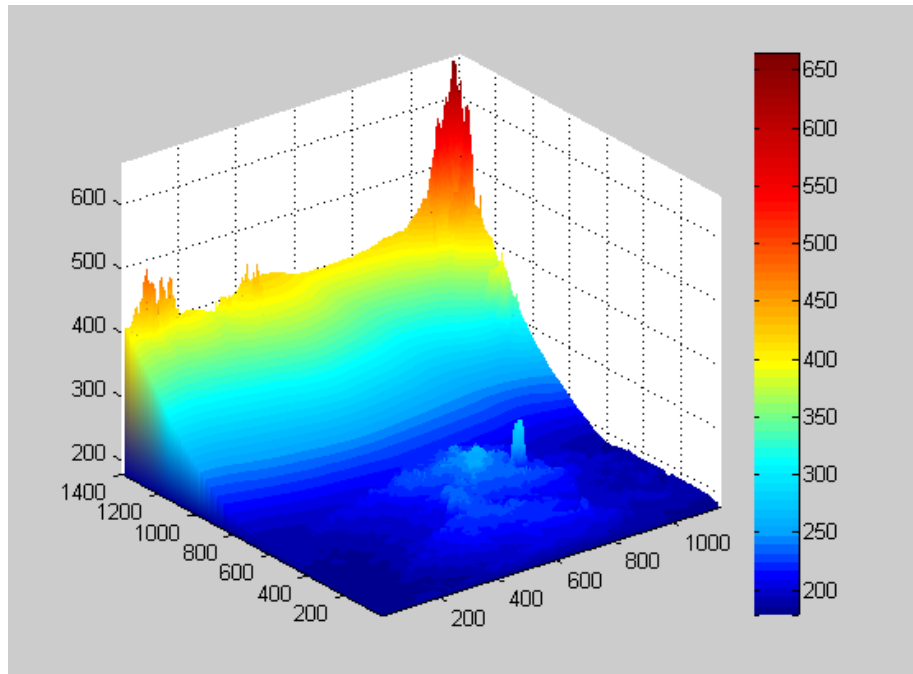
Table 4.1: Terrain elevation range and path type for different terrain profile

Terrain Profile	Terrain Elevation (m)					Path (%)	
	Highest	Lowest	Difference	Mean	Standard Deviation	Free Space	Block
Adelanto	1010	838	173	901.3	40.123	56	44
Amboy Crater	664	179	485	247.6	66.830	26	74
Jersey City	106	0	106	6.5	12.293	30	70
Amsterdam	319	73	246	183.9	50.396	12	88
Arrowhead Butte	6405	5709	696	5953.7	149.899	8	92
Acton	1992	703	1289	1067.2	266.062	7	93
Caples Lake	10382	6576	3806	8109.5	729.164	7	93

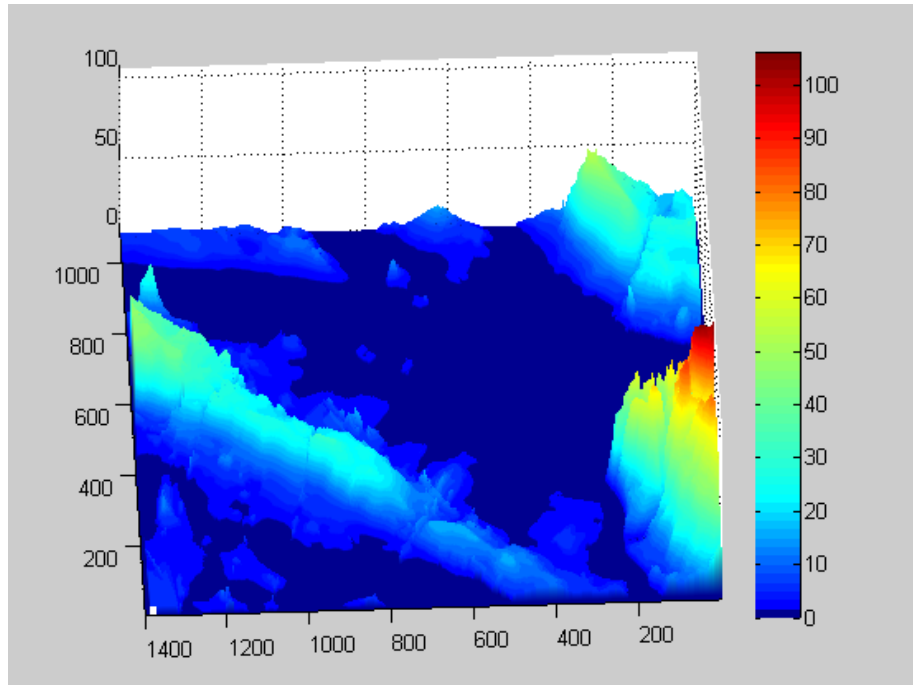
To obtain more insight on how a terrain profile affecting the percentage of detected free space path, a three-dimensional (3D) view of each terrain profile is plotted and shown in Figure 4.2 and Figure 4.3.



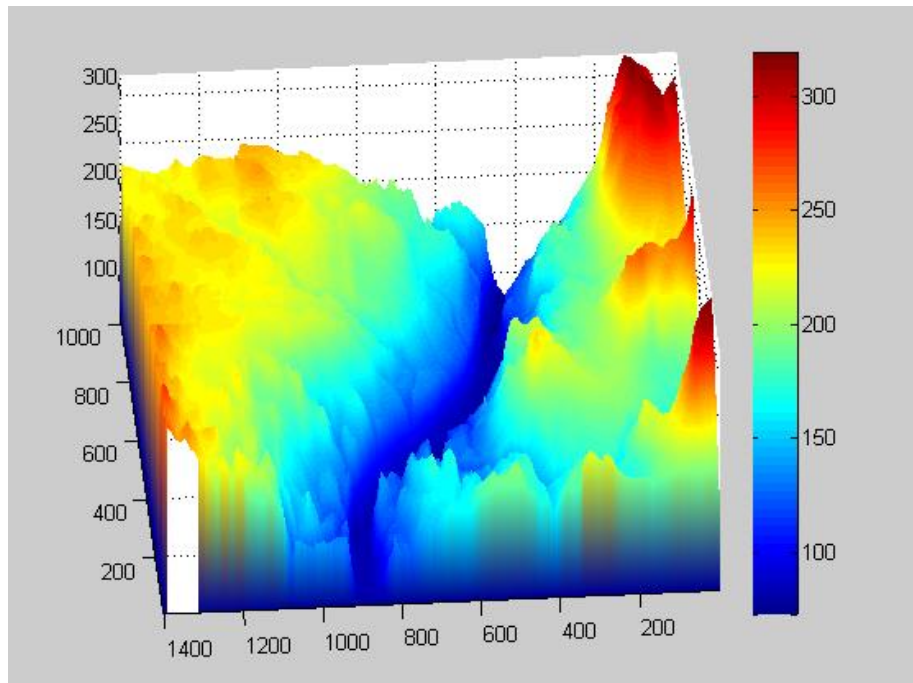
(a) Adelanto



(b) Amboy Crater

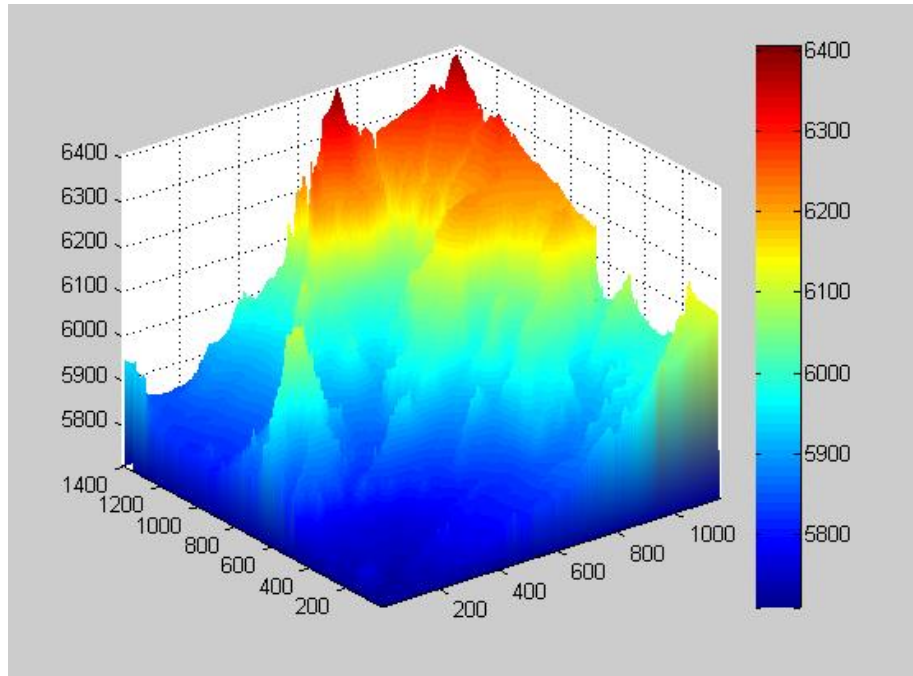


(c) Jersey City

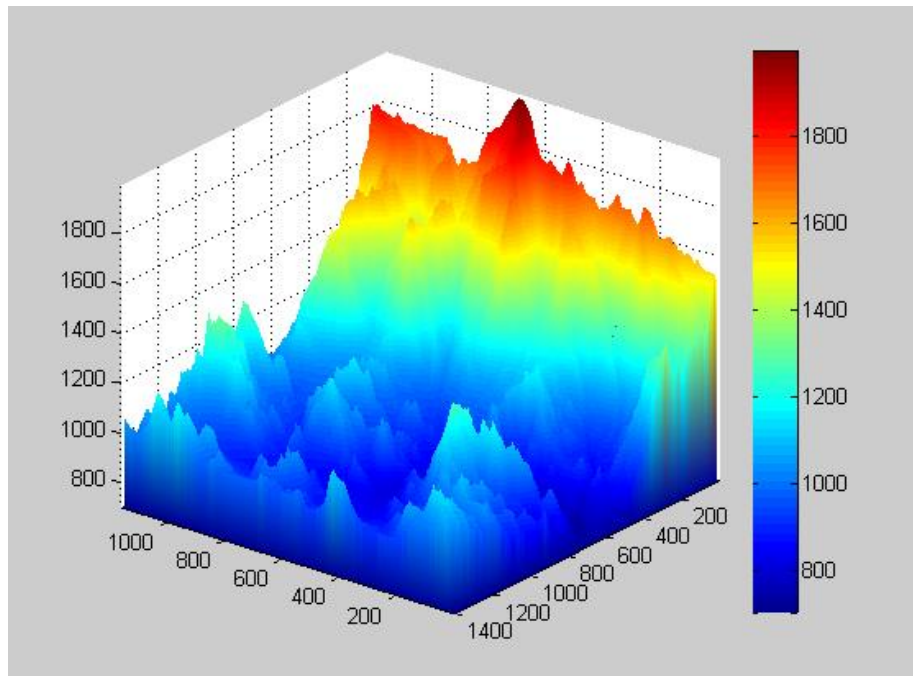


(d) Amsterdam

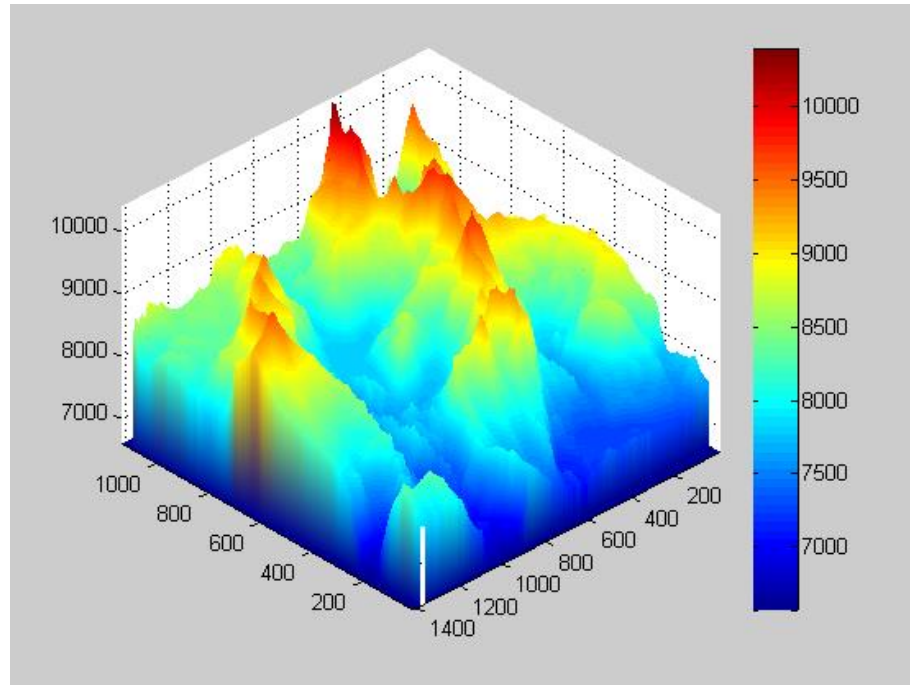
Figure 4.2: Low terrain elevation difference profiles



(a) Arrowhead Butte



(b) Acton



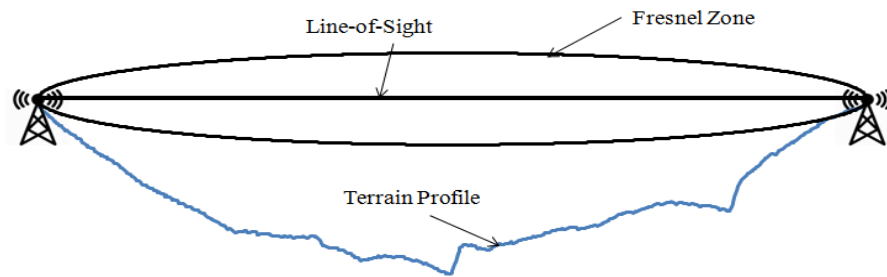
(c) Caples Lake

Figure 4.3: High terrain elevation difference profiles

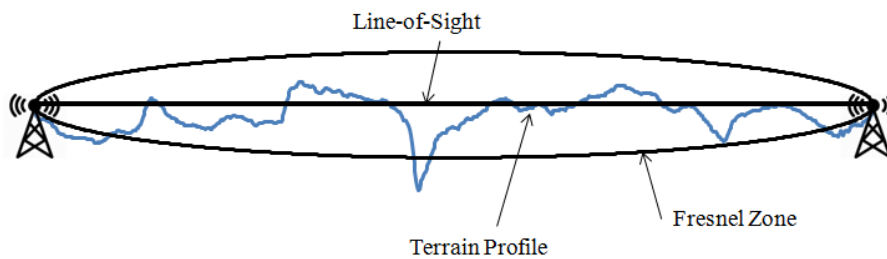
From Figure 4.2 and Figure 4.3, it shows that the percentage of detected free space path is mainly dependent on the placement of the transmitter and the receiver (or the terrain surface). The main reason contributing to high percentage of free space path in Adelanto is due to the whole terrain is a slope with even surface. Even though Jersey City has the lowest terrain elevation difference and standard deviation, from the 3D view shown in Figure 4.2 (c), Jersey City is composed of multiple islands with different heights. Thus, the chance for a signal being blocked by obstacles is higher in Jersey City as compared to Adelanto. As shown in Figure 4.3, the mountainous regions with uneven terrain surface contribute to the high percentage of detected free space path to Arrowhead Butte, Acton and Caples Lake. As a conclusion, the terrain profile should be considered in computing the path pass loss between two antennas.

### 4.1.2 Path Loss Analysis

To study the effect of terrain profile on the path loss between two antennas, a few examples of 8 kilometres long path profile for Adelanto are chosen and illustrated in Figure 4.4. A path between transmitter-receiver pair is considered as a free path if no obstacles are found in the Fresnel Zone; otherwise, the path is treated as a blocked path. The path profiles for other maps are ignored as it follows the same analysis for Adelanto. Some cases of path loss simulation based on the terrain aware model are studied to highlight some important observations as follows:

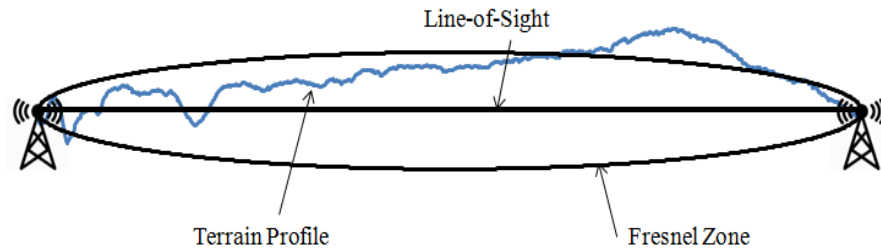


(a) Adelanto free path



(b) Adelanto partial blocked path





(c) Adelanto blocked path

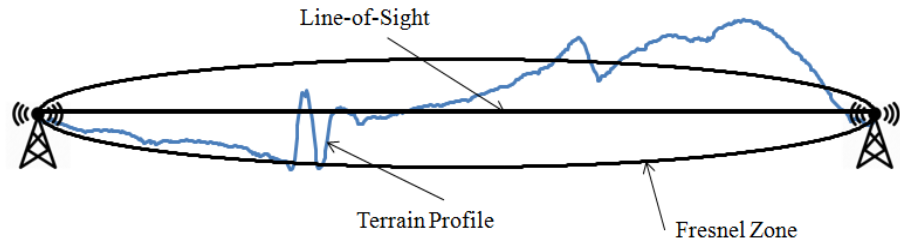
Figure 4.4: Adelanto path profiles

**Observation 1:** The higher number of obstructing peaks between the T-R pair yields the higher path loss.

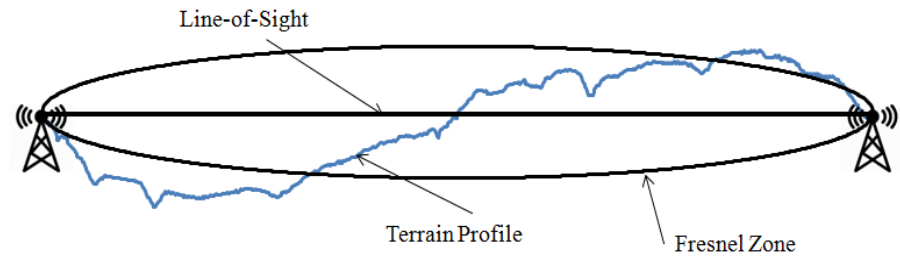
Examples: Figure 4.4 (a) shows that a free space loss of 118.07 dB is attained when the path profile between two antennas is unobstructed. Meanwhile, Figure 4.4 (b) shows that a partial blocked path with four peaks in between the T-R pair has a path loss of 142.139 dB. Lastly, Figure 4.4 (c) shows that a blocked path with 122 peaks in between the T-R pair has a path loss of 839.558 dB.

**Observation 2:** Other than the number of obstructing peaks, the height of the obstruction affects the path loss between two antennas.

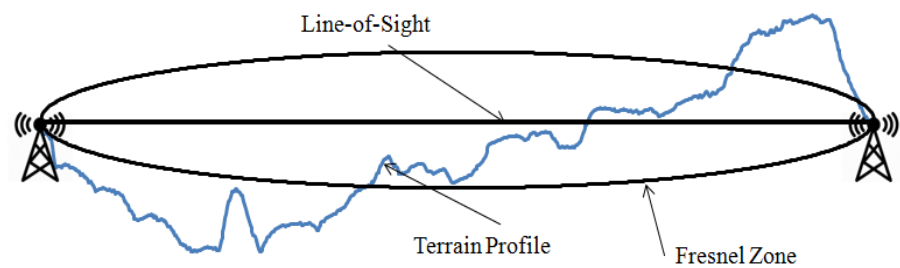
Examples: Figure 4.5 shows three different path profiles that consist of five obstructing peaks in between two antennas. These three path profiles have different path loss of (a) 155.004 dB, (b) 153.164 dB, and (c) 150.863 dB. The total path loss is affected by the height of the obstruction in the path profile. This observation tallies with Equation 2.3 and Equation 2.4.



(a) Adelanto blocked path 1



(b) Adelanto blocked path 2



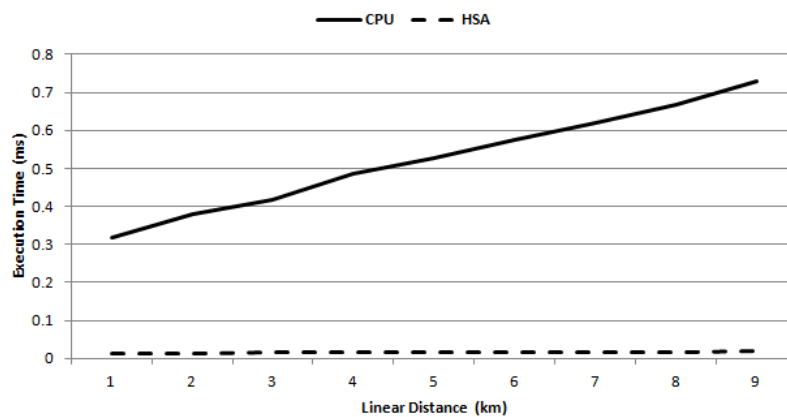
(c) Adelanto blocked path 3

Figure 4.5: Adelanto block path with 5 obstructing peaks

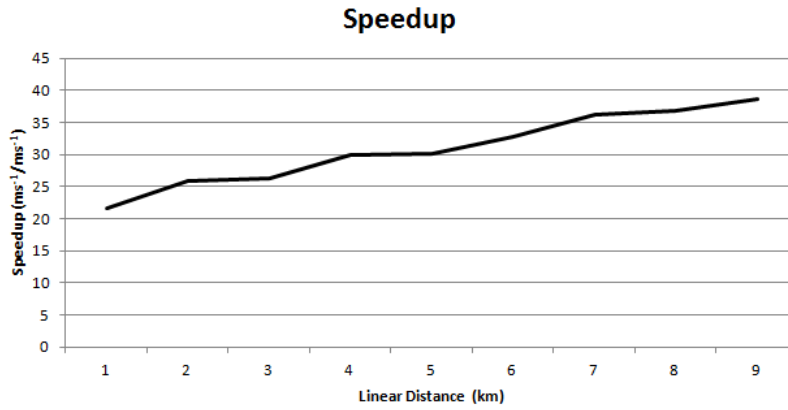
### 4.1.3 Speed Performance of The Terrain Aware Model Implemented Using CPU Architecture and Heterogeneous System Architecture

Figure 4.6 to Figure 4.8 shows the execution time and speedup comparison of the terrain aware model implemented using CPU architecture and heterogeneous system architecture on different terrain profiles. Figure 4.6 (a), Figure 4.7 (a) and Figure 4.8 (a) show the execution time (measured in

milliseconds) for CPU architecture increases proportionally against the linear distance between two antennas, whereas the execution time for HSA architecture remains almost the same regardless the linear distance between two antennas. Meanwhile, Figure 4.6 (b), Figure 4.7 (b) and Figure 4.8 (b) show the ratio of execution times for HSA over CPU against the linear distance between two antennas after including the path file in path loss computation. Overall, a simulation speedup of 20x to 42x is achieved for all seven selected maps by just porting two of the most compute-intensive processes into GPU for parallel processing to form the required path profile. This proves the successfulness in offloading the compute-intensive processes of network simulation through the use of parallel co-processors.

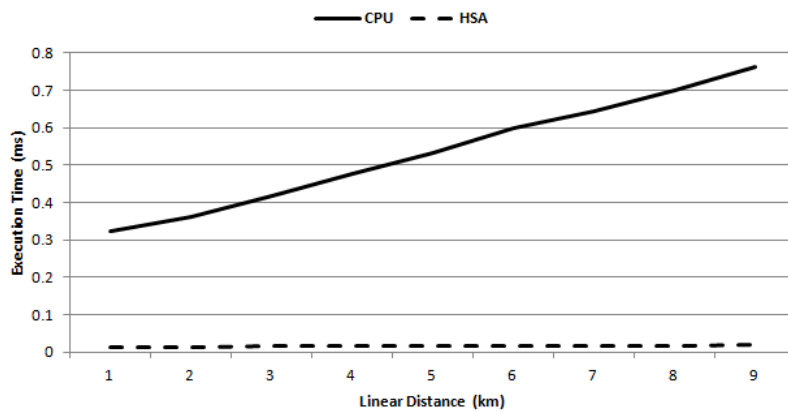


(a) Simulation time of CPU and HSA

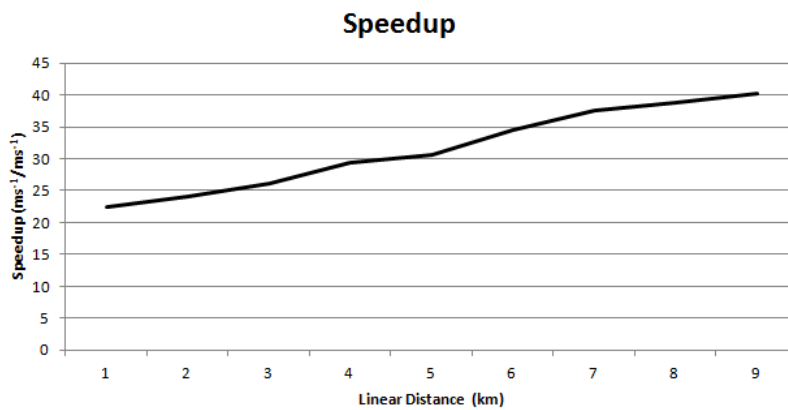


(b) Speedup of HSA over CPU

Figure 4.6: Simulation time and speedup for terrain profiles with low terrain elevation difference

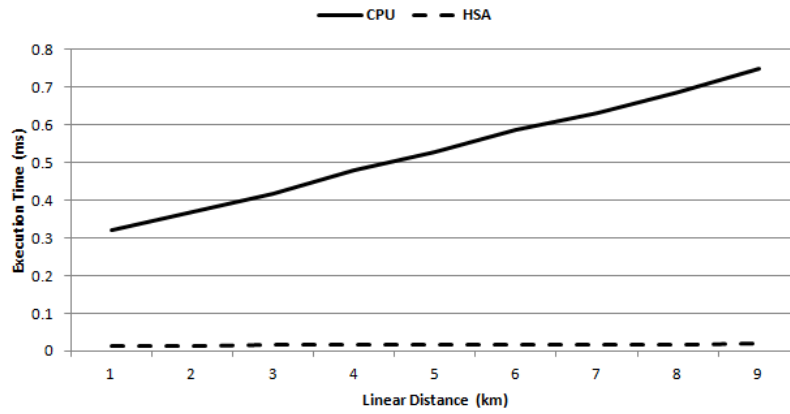


(a) Simulation time of CPU and HSA

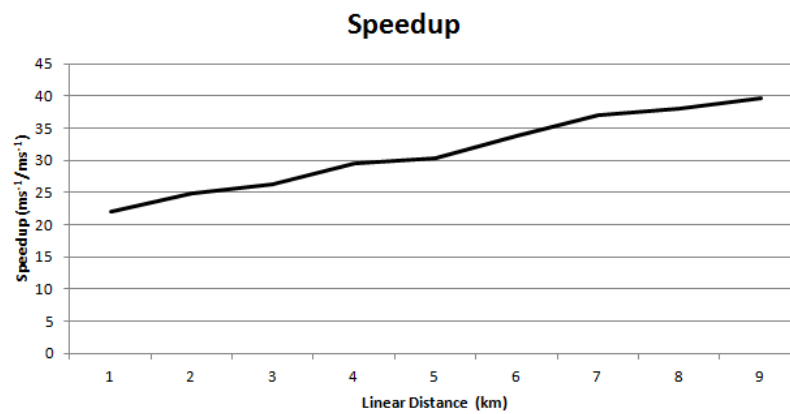


(b) Speedup of HSA over CPU

Figure 4.7: Simulation time and speedup for terrain profiles with high terrain elevation difference



(a) Simulation time of CPU and HSA



(b) Speedup of HSA over CPU

Figure 4.8: Overall simulation time and speedup for all terrain profiles

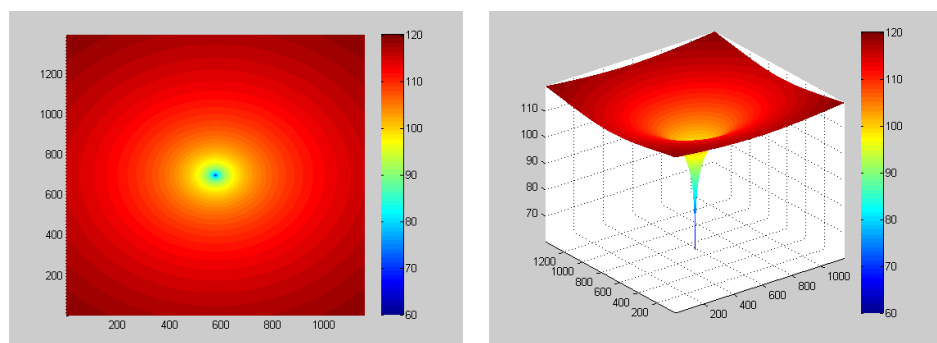
## 4.2 Point-to-Point Model

The heterogeneous based path loss simulation using terrain aware model and point-to-point model is performed. The location of a transmitter is fixed at the center of a map and the path loss between the transmitter and a receiver that is located at every possible location in the same map is computed. For comparison, the CPU path loss simulation using disc model and point-to-point model is also performed. For terrain aware model, additional terrain

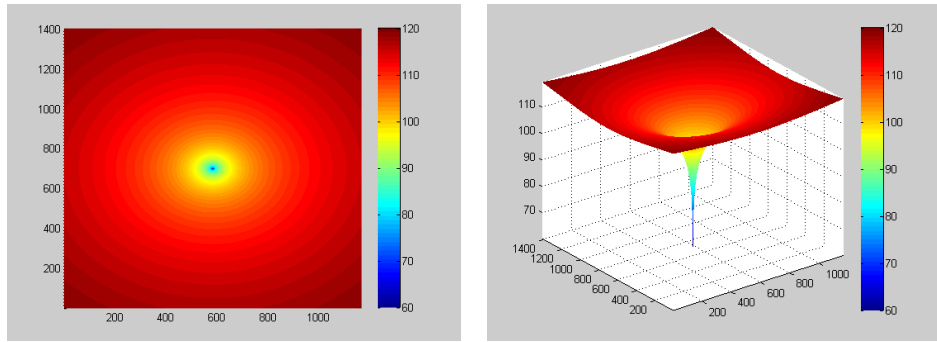
profiles between a transmitter and a receiver are extracted from the DEM data provided by USGS. Even though throughout the simulation, the transmitter is located at the center of a map, the position of transmitter can be easily relocated to any other position.

#### 4.2.1 Conventional Method: Disc Model

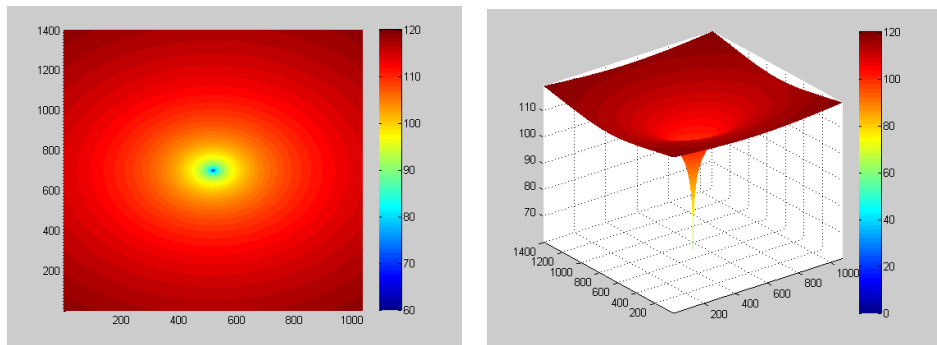
The disc model used in this simulation is implemented using conventional CPU architecture. Figure 4.9 to Figure 4.10 shows the 2D and 3D visualised image of the path loss simulation calculated based on the disc model for all seven selected maps. All path loss simulations calculated based on the disc model on seven different maps share the same result as the path loss simulation based on the disc model solely depends on the distance between a transmitter and a receiver without considering the terrain profiles between these two nodes. This pattern can be easily spotted through the ring shapes where the path loss simulations from the transmitter to the receiver with the same distance are always same regardless of the terrain profile between the transmitter and the receiver.



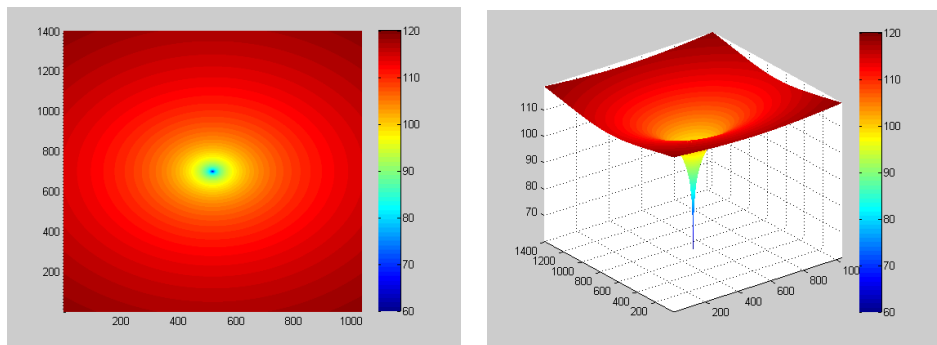
(a) Adelanto



(b) Amboy Crater

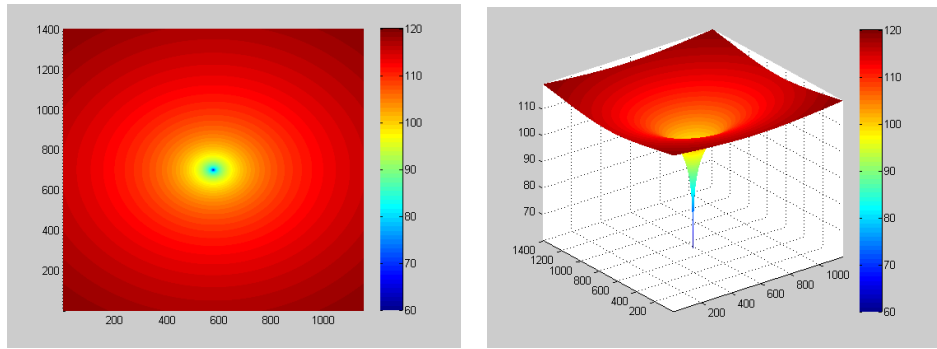


(c) Jersey City

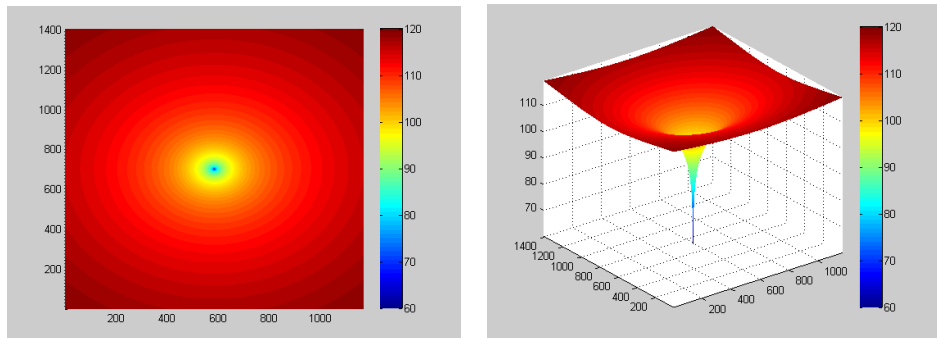


(d) Amsterdam

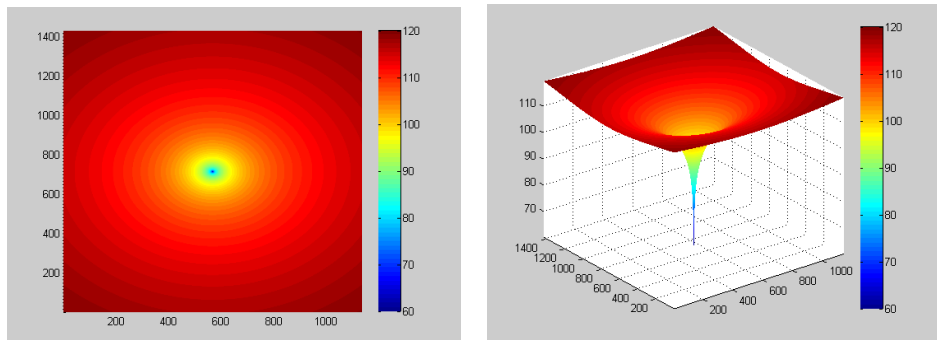
Figure 4.9: Path loss simulation based on disc model for terrain profiles with low terrain elevation difference



(a) Arrowhead Butte



(b) Acton



(c) Caples Lake

Figure 4.10: Path loss simulation based on disc model for terrain profiles with high terrain elevation difference

#### 4.2.2 Proposed Method: The Terrain Aware Model

The terrain aware model used in this simulation is implemented using heterogeneous system architecture. Throughout the experiments, the transmitter indicated by a red dot is placed in the middle of the selected maps.



Besides, the receiver is placed at every possible position available in the map. Table 4.2 shows details of different selected terrain profiles. The antenna height is the height of the transmitter location, which is the total of the terrain elevation obtained from DEM data and the 0.5m height of the antenna.

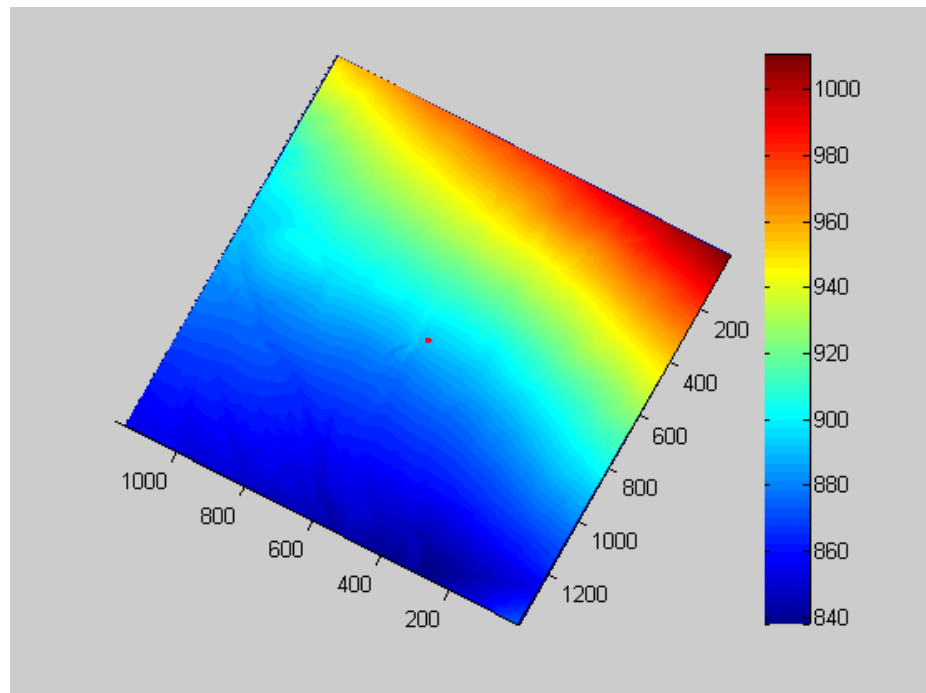
Table 4.2: Details of the selected terrain profiles

Terrain Profile	Terrain Elevation (m)		Antenna Height (m)
	Highest	Lowest	
Adelanto	1010	838	890.7
Amboy Crater	664	179	219.92
Jersey City	106	0	0.5
Amsterdam	319	73	107.93
Arrowhead Butte	6405	5709	5980.5
Acton	1992	703	841.09
Caples Lake	10382	6576	7920.4

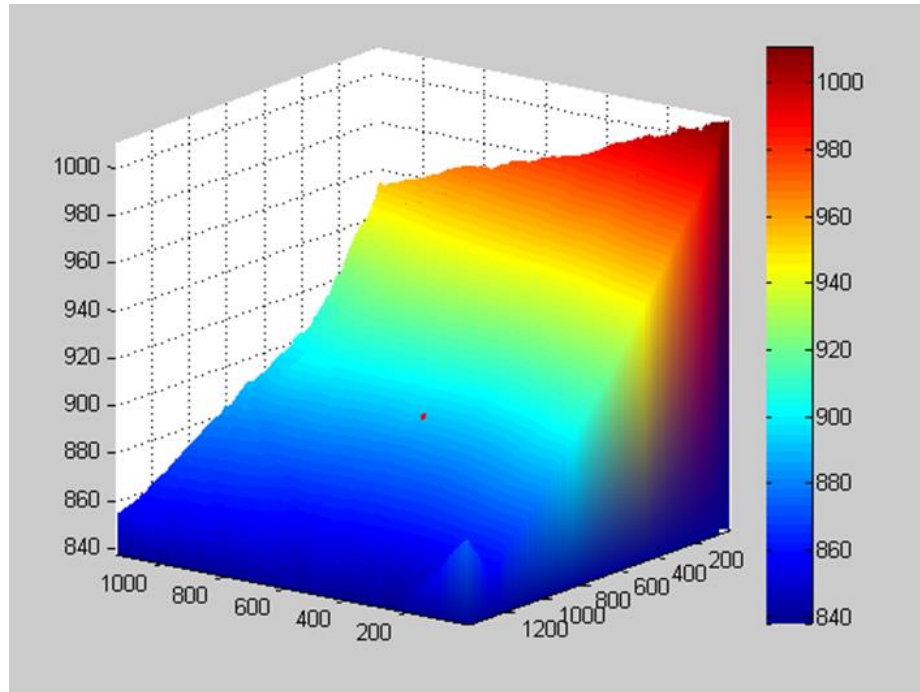
Figure 4.11 to Figure 4.17 show the path loss simulation calculated based on the terrain aware model in the form of 2D and 3D visualised images. For a better comparison, the map of each terrain profile is included and the images are rotated accordingly. Notice that all path loss simulations based on the terrain aware model on seven different maps are all different as the terrain aware model depends on both distance and terrain profile between a transmitter and a receiver. Notice that the darker red line indicates higher path loss between the T-R pair. Besides, the maximum path loss between the T-R pair is limited to 120 dB to present a clear view with standardised colour bar for all seven maps due to there being only a few points in the maps that have path loss above 120 dB. The results on seven different maps are explained in details as follows:

- a. Adelanto: Adelanto is one of the terrain profiles with low terrain elevation difference. The transmitter is located at a smooth hill with the

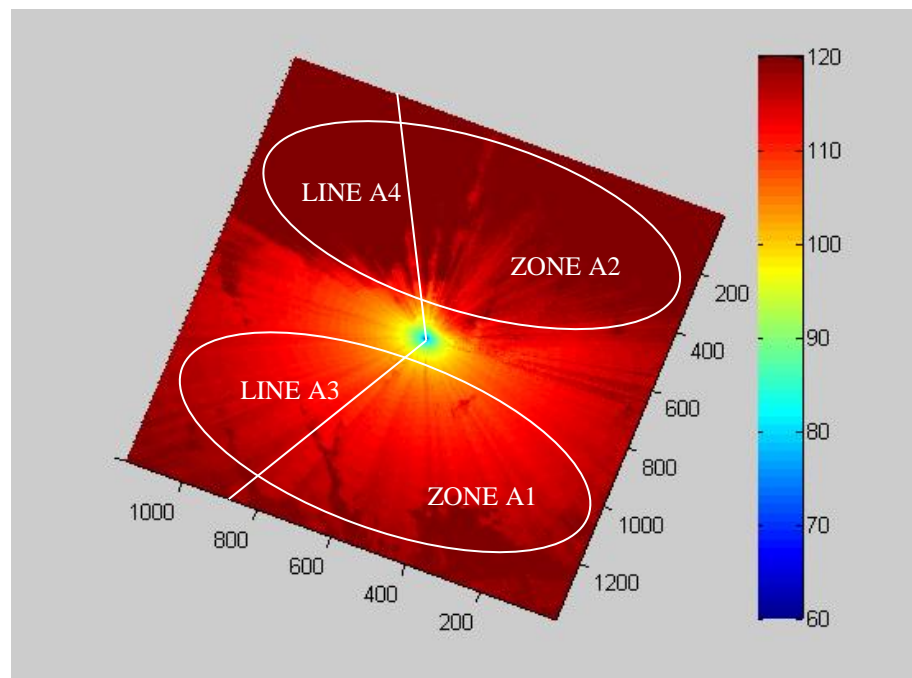
height of 890.7m as shown in Figure 4.11 (a) and Figure 4.11 (b). Figure 4.11 (c) shows clearly the placement of the receiver affects the path loss between the T-R pair after considering the terrain profile. For an illustration, LINE A3 and LINE A4 drawn in Figure 4.11 (c) shows different magnitudes of path loss even the T-R pair is of equal distance. Similarly, as shown in Figure 4.11 (c), the placement of the transmitter in the middle of Adelanto can cover major areas in ZONE A1; however, major areas in ZONE A2 are with weaker strength of signal.



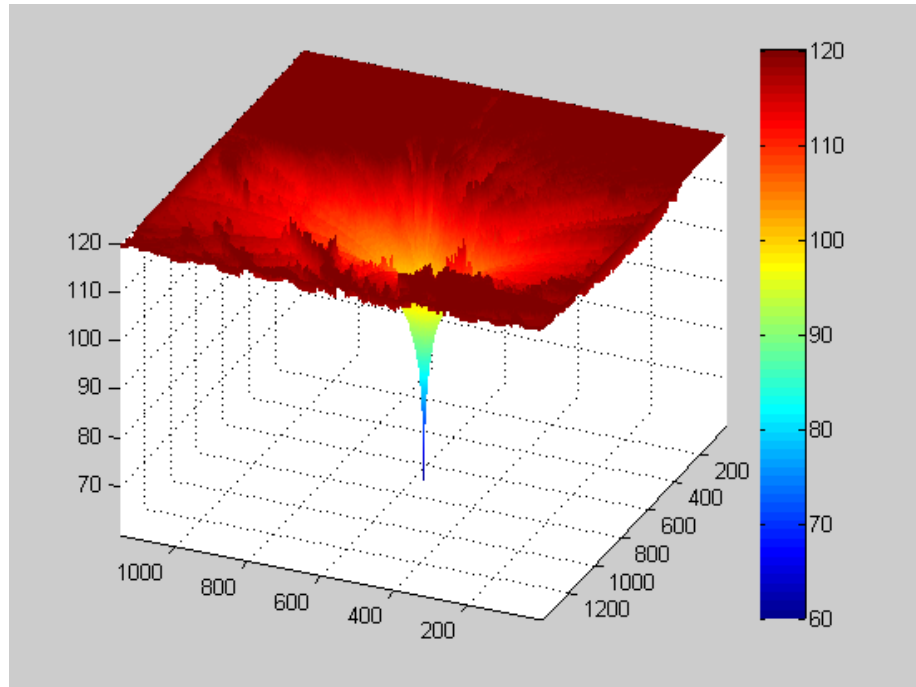
(a) 2D view of Adelanto terrain profile



(b) 3D rotated view of Adelanto terrain profile



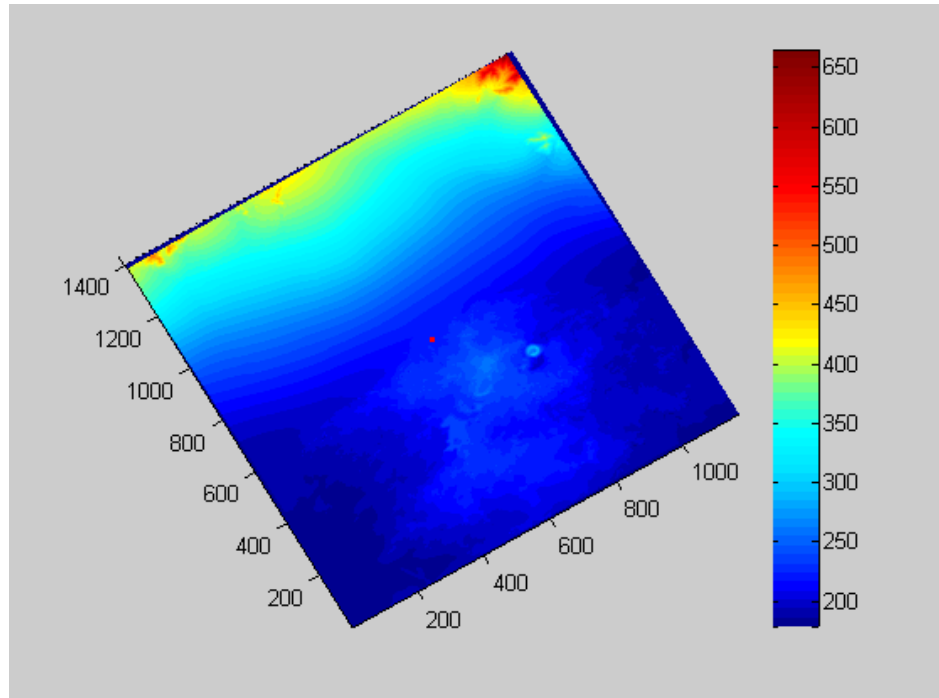
(c) 2D rotated view of Adelanto's path loss simulation



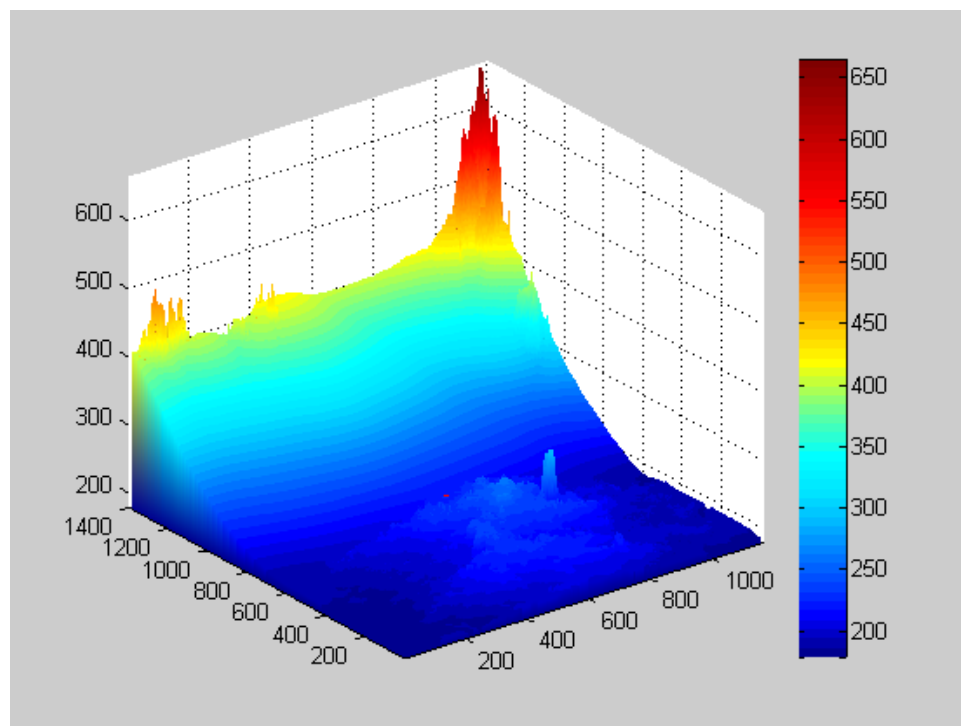
(d) 3D rotated view of Adelanto's path loss simulation

Figure 4.11: Simulation results on Adelanto terrain profile

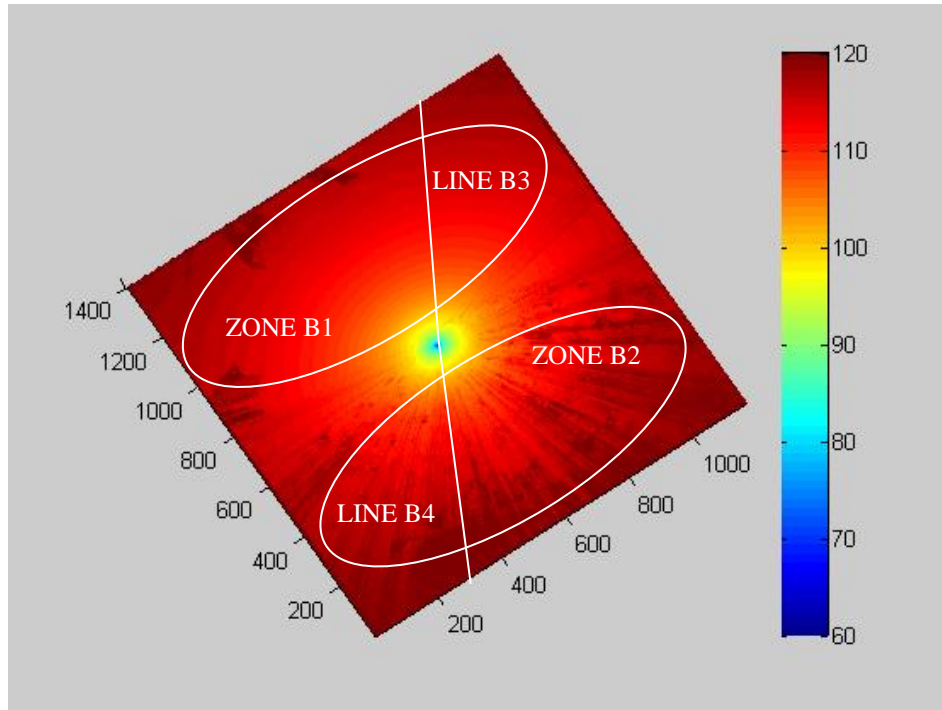
b. Amboy Crater: Amboy Crater is one of the terrain profiles with low terrain elevation difference. The transmitter is located at the base of a hill and at the edge of an uneven slope with the height of 219.92m as shown in Figure 4.12 (a) and Figure 4.12 (b) As shown in Figure 4.12 (c) and Figure 4.12 (d), it is interesting to observe that the disc model is suitable to be used in ZONE B1 as no obvious obstacles existing in this zone while the terrain aware model is more suitable to be used in ZONE B2 due to the existing of multiple obstacles in this zone. Similarly, LINE B3 shows that the path loss along the direction solely depends on the distance between the T-R pair while LINE B4 shows different magnitudes of path loss along the direction.



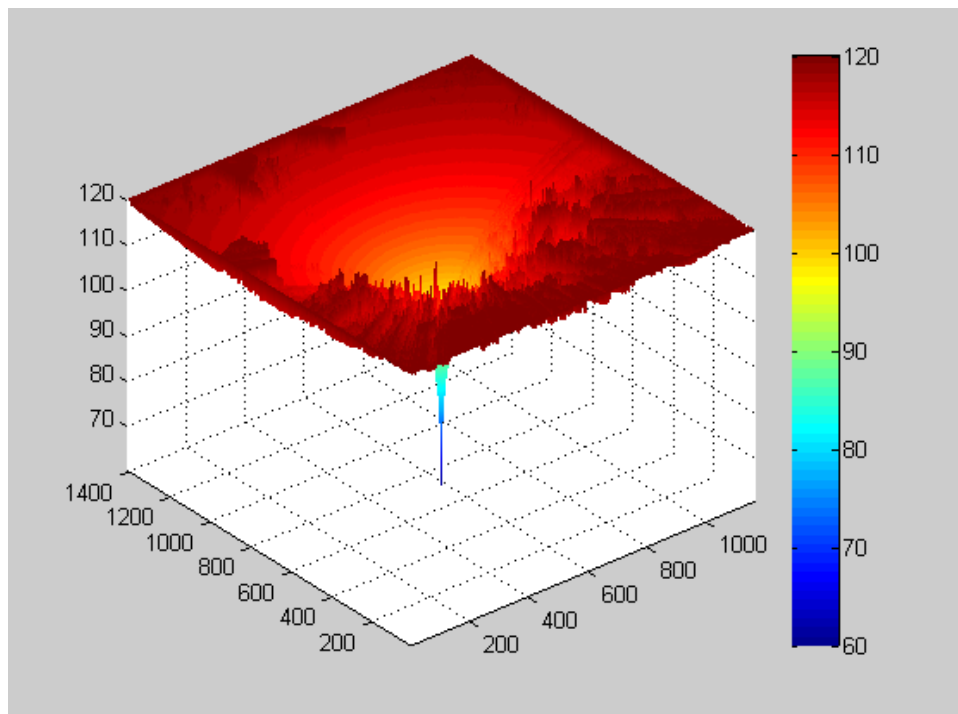
(a) 2D view of Amboy Crater terrain profile



(b) 3D rotated view of Amboy Crater terrain profile



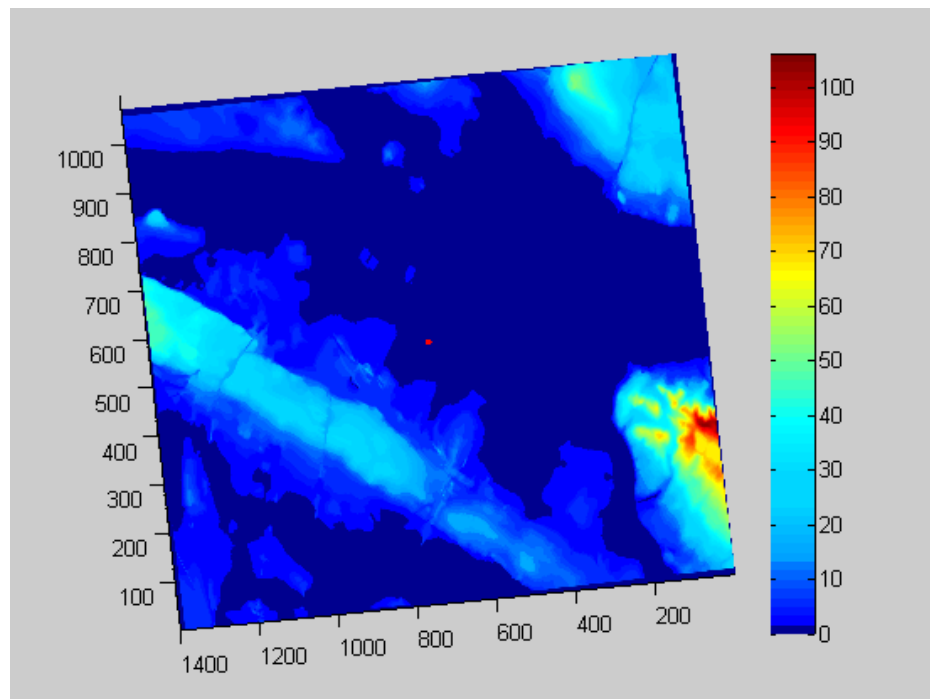
(c) 2D rotated view of Amboy Crater's path loss simulation



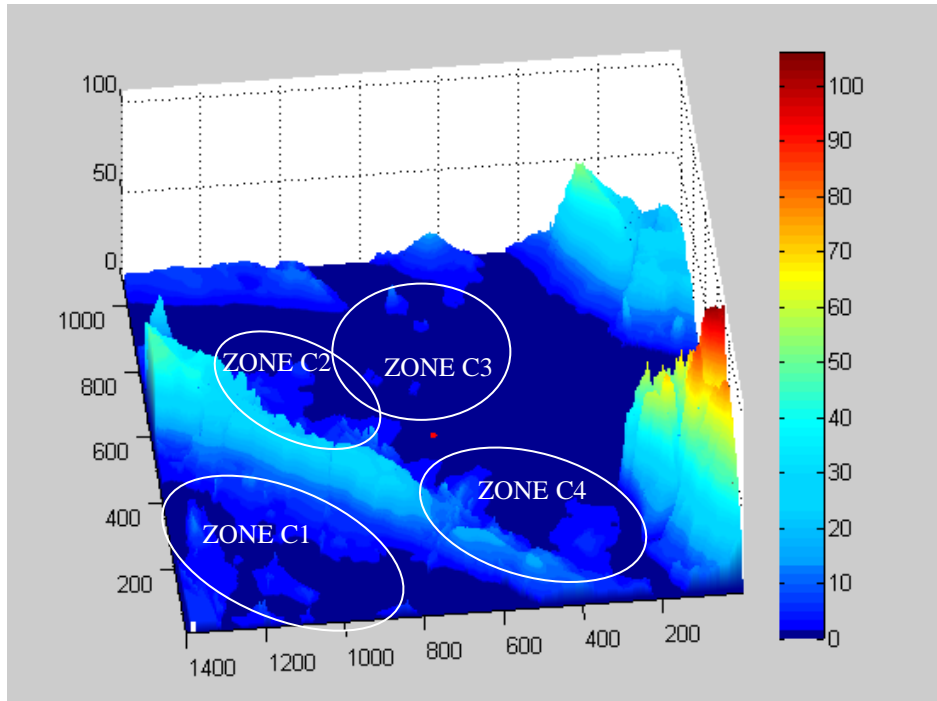
(d) 3D view of Amboy Crater's path loss simulation

Figure 4.12: Simulation results on Amboy Crater terrain profile

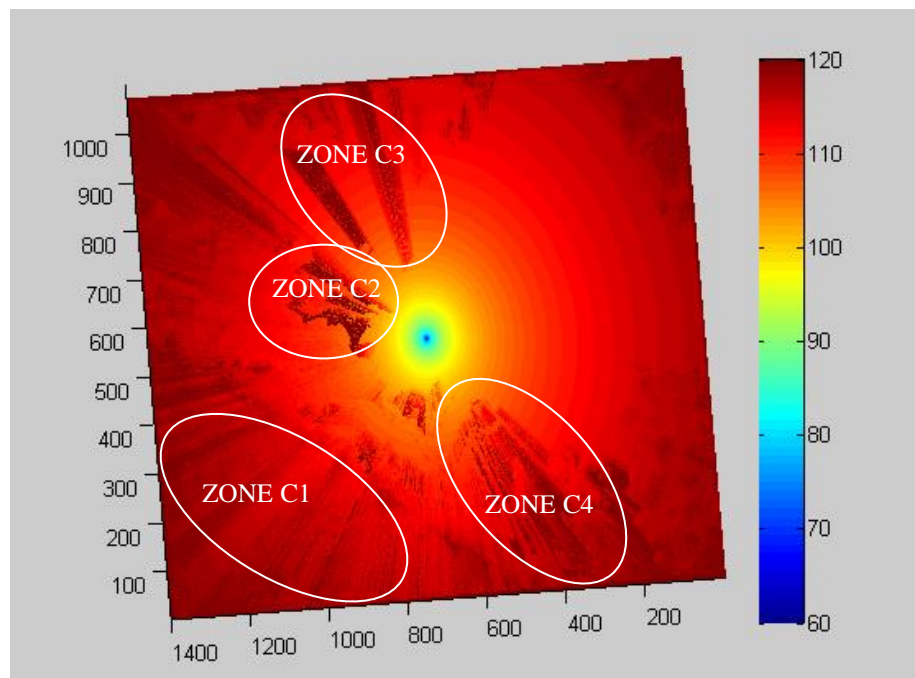
- c. Jersey City: Jersey City is one of the terrain profiles with low terrain elevation difference. The transmitter is located at sea level, the lowest point with only a 0.5m height extension from the antenna as shown in Figure 4.13 (a) and Figure 4.13 (b). As shown in Figure 4.13 (c) and Figure 4.13 (d), ZONE C1, ZONE C2, ZONE C3 and ZONE C4 show higher path loss between the T-R pair due to the blocking of multiple islands in between the T-R pair, especially the transmitter is placed at the lowest position. Thus, the placement of transmitter at higher position is important to make sure bigger area can be covered.



(a) 2D view of Jersey City terrain profile

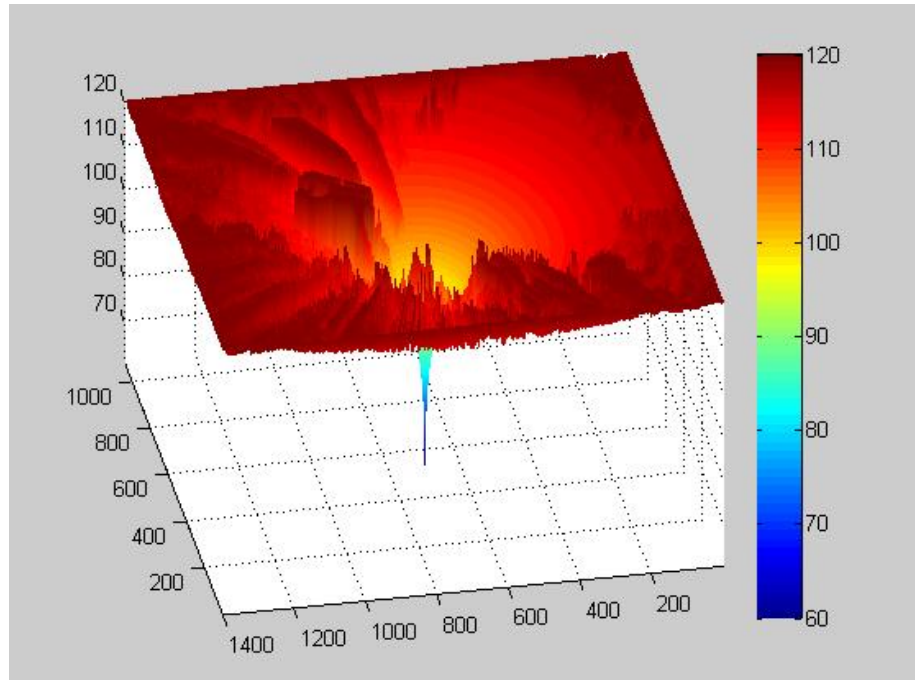


(b) 3D rotated view of Jersey City terrain profile



(c) 2D rotated view of Jersey City's path loss simulation

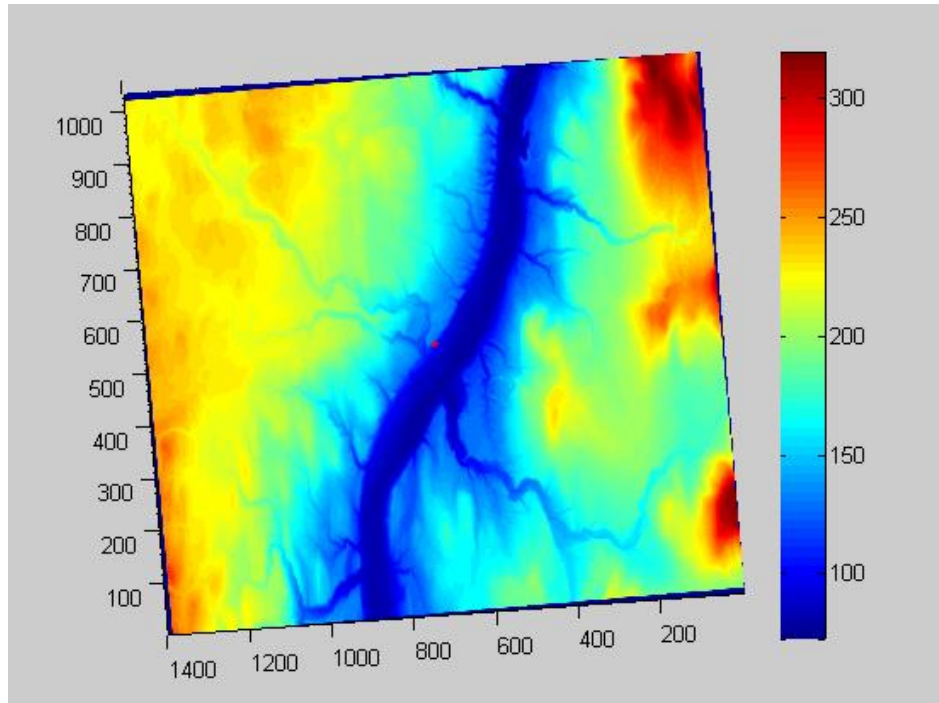




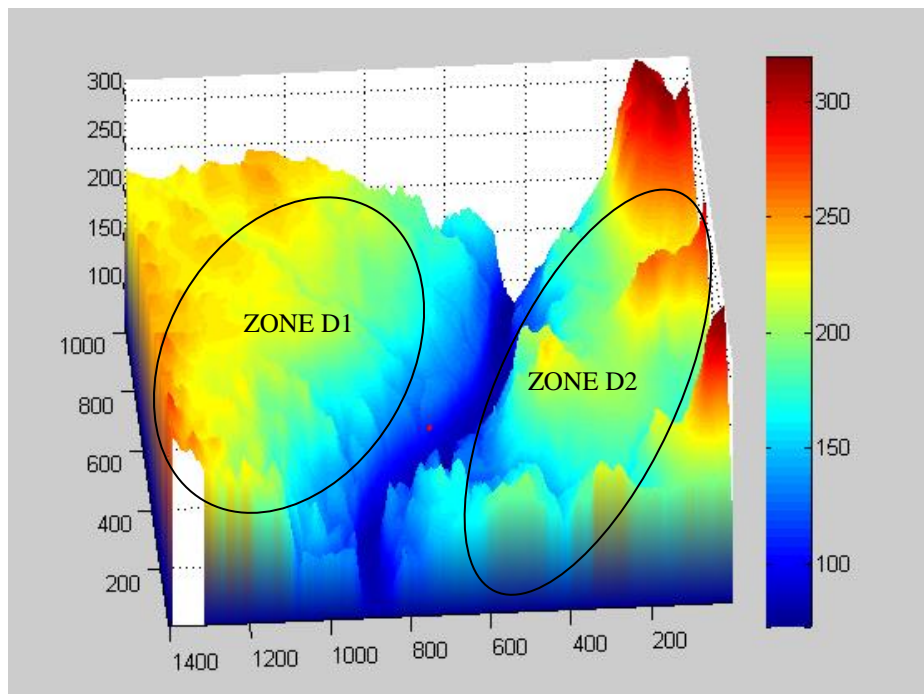
(d) 3D view of Jersey City's path loss simulation

Figure 4.13: Simulation results on Jersey City terrain profile

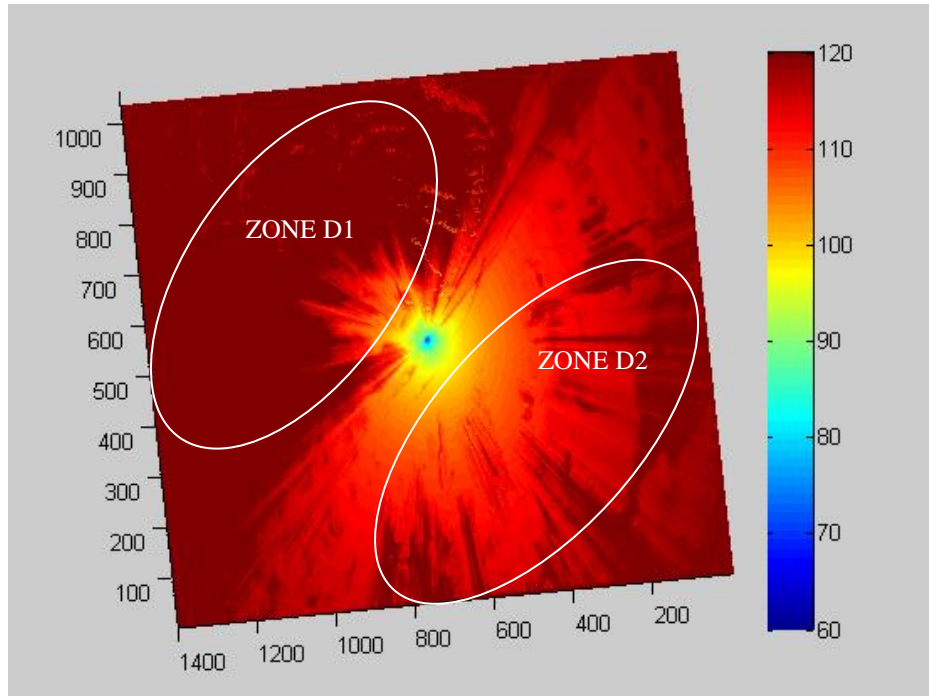
- d. Amsterdam: Amsterdam is the capital of Netherlands where a river separates two high terrains. The transmitter is located in the river with the height of 107.93m as shown in Figure 4.14 (a) and Figure 4.14 (b). As shown in Figure 4.14 (c) and Figure 4.14 (d), the transmitter is misallocated at the base of the left terrain where the signal will be greatly affected when the receiver is located at the left terrain. Meanwhile, the signal with greater strength can be detected at the right terrain as the transmitter is located further away from the base of high terrain.



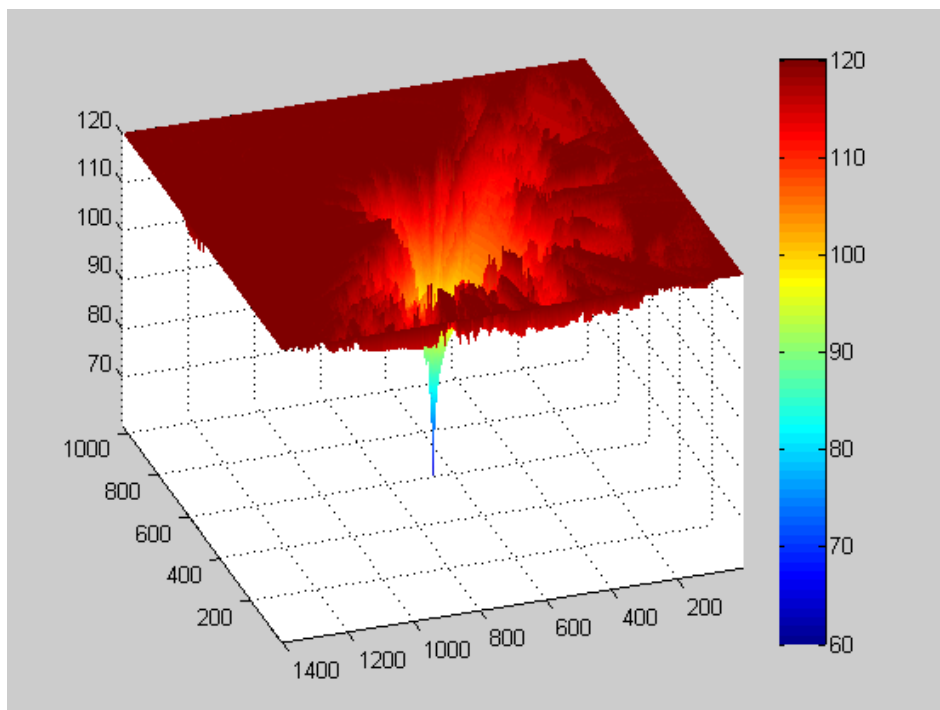
(a) 2D view of Amsterdam terrain profile



(b) 3D rotated view of Amsterdam terrain profile



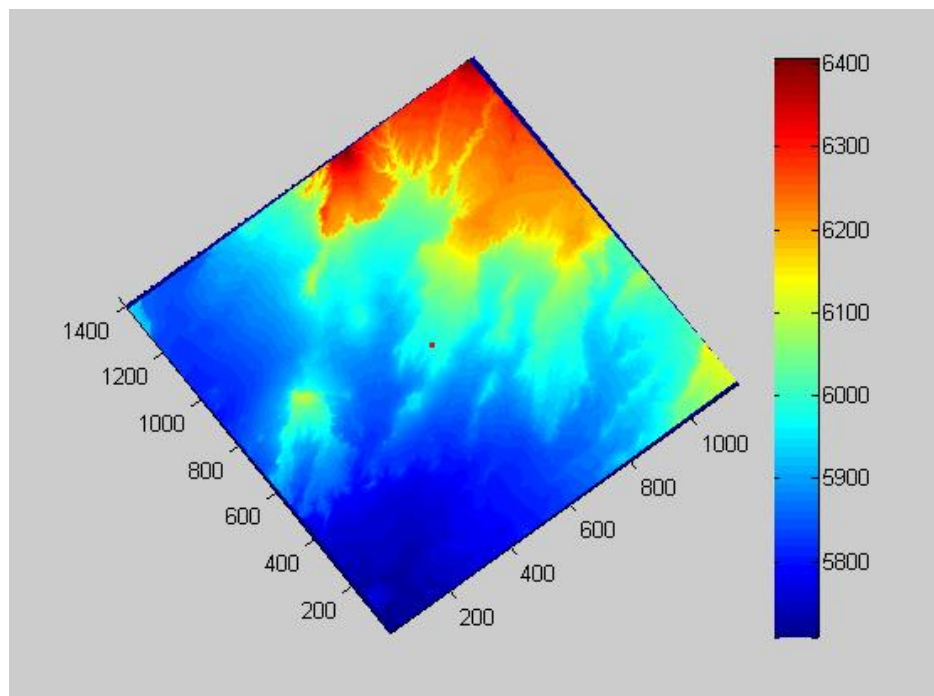
(c) 2D rotated view of Amsterdam's path loss simulation



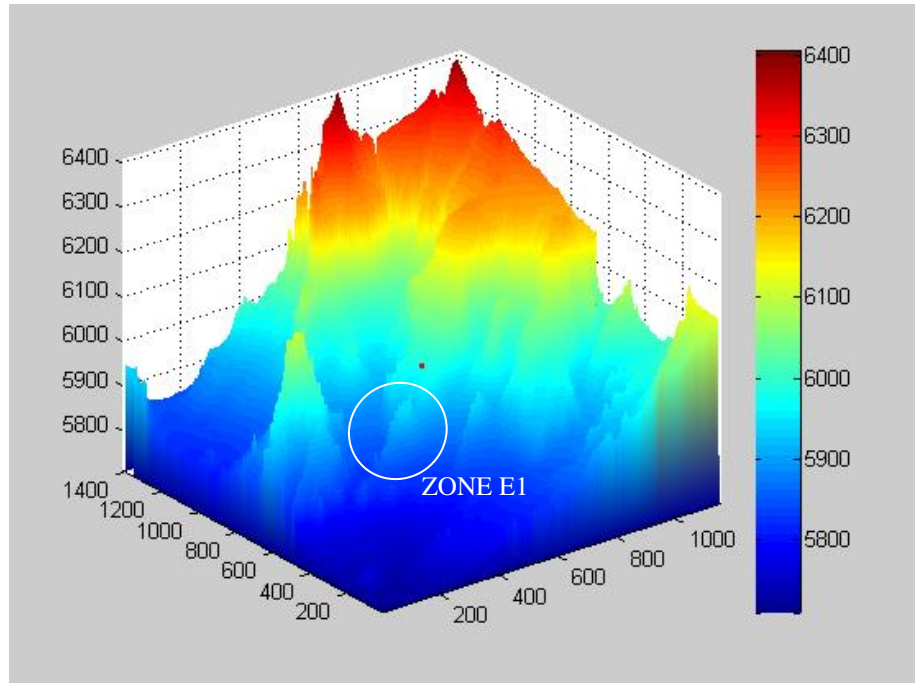
(d) 3D view of Amsterdam's path loss simulation

Figure 4.14: Simulation results on Amsterdam terrain profile

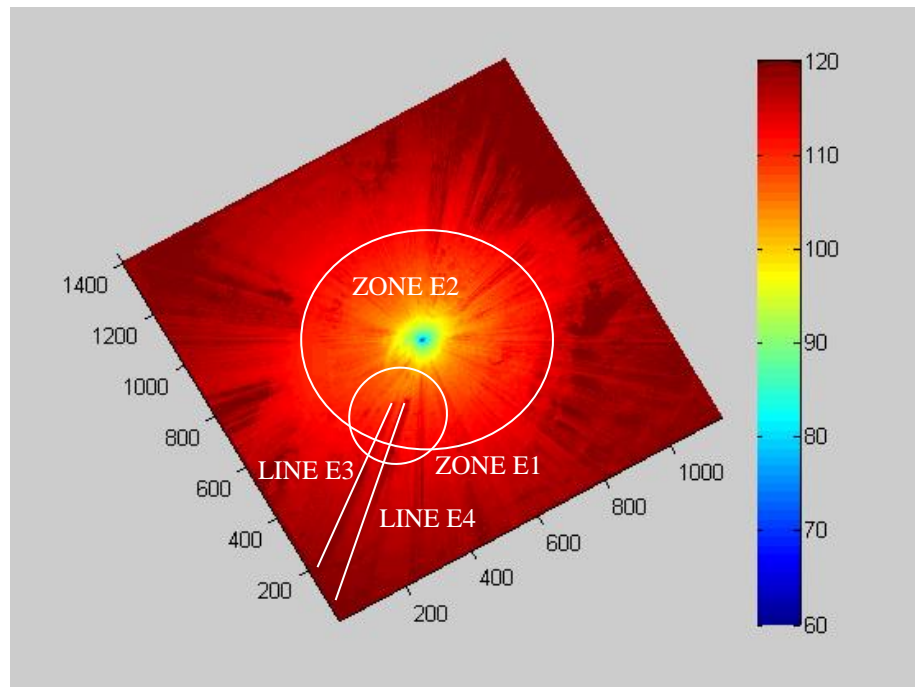
- e. Arrowhead Butte: Arrowhead Butte is one of the terrain profiles with high terrain elevation difference. The transmitter is located at the middle of a hill with the height of 5980.5m as shown in Figure 4.15 (a) and Figure 4.15 (b). As shown in Figure 4.15 (c) and Figure 4.15 (d), ZONE E2 shows a fairly even distribution of path loss due to no obvious obstacles blocking between the T-R pair. However, outside of ZONE E2 shows uneven distribution of path loss due to the existing of obstacles blocking between the T-R pair. For an illustration, LINE E3 and LINE E4 show higher path loss due to the blockage of multiple hills from ZONE E1.



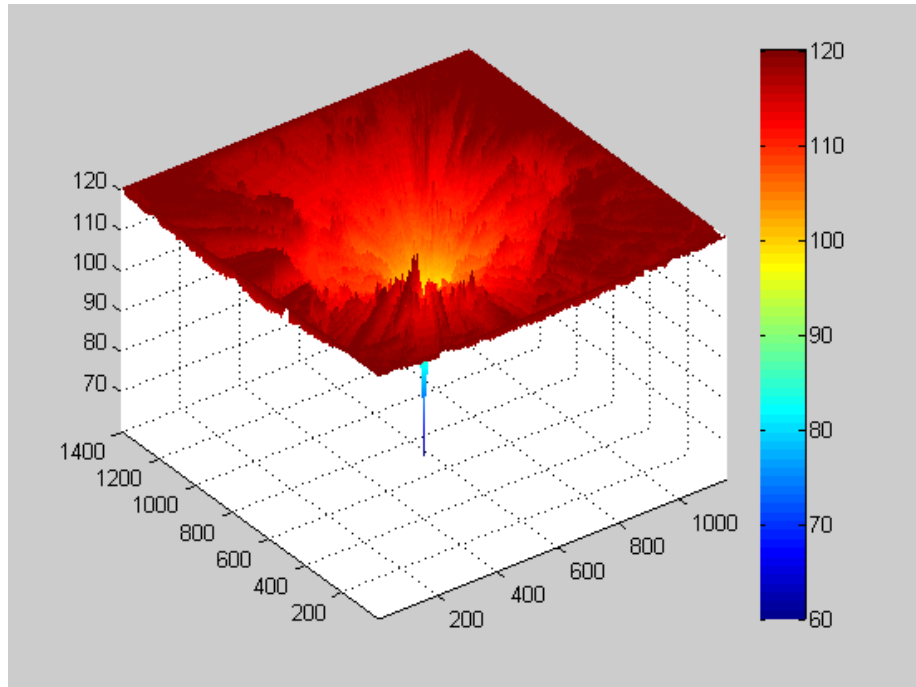
(a) 2D view of Arrowhead Butte terrain profile



(b) 3D rotated view of Arrowhead Butte terrain profile



(c) 2D rotated view of Arrowhead Butte's path loss simulation

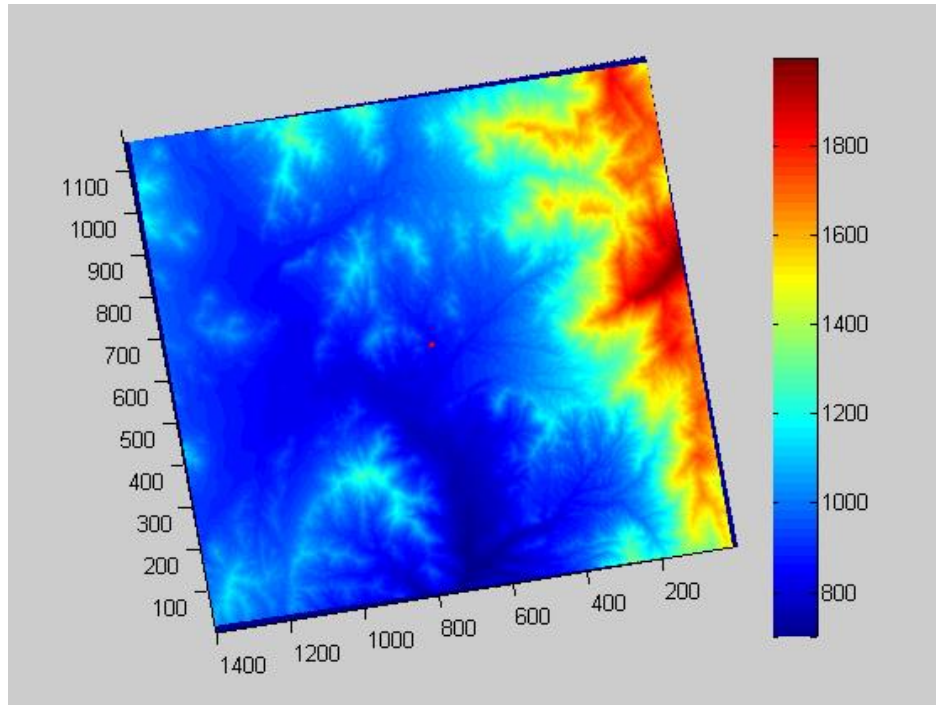


(d) 3D view of Arrowhead Butte's path loss simulation

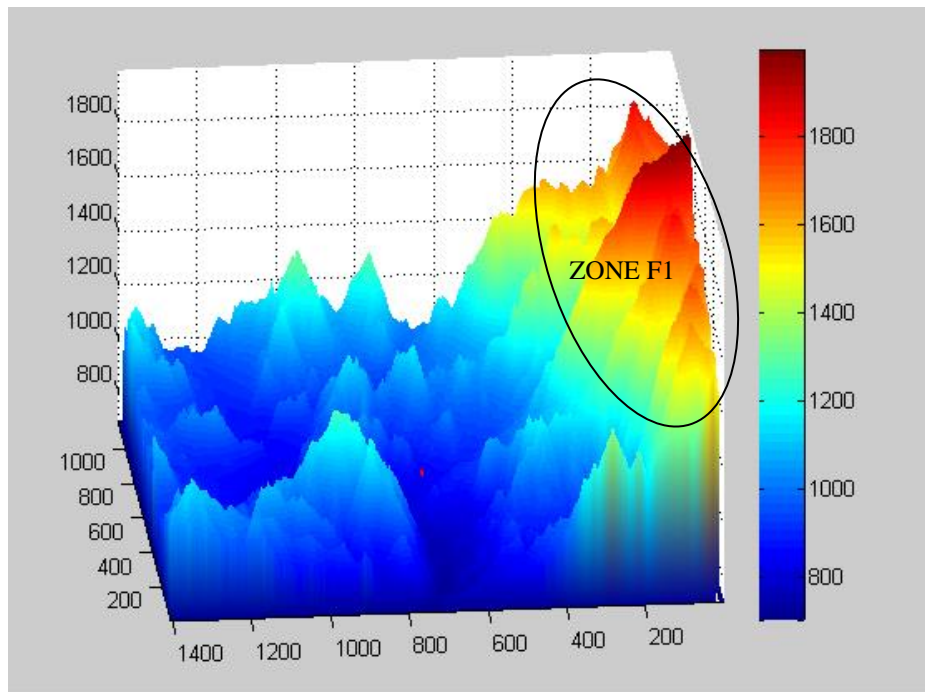
Figure 4.15: Simulation results on Arrowhead Butte terrain profile

- f. Acton: Acton is one of the terrain profiles with high terrain elevation difference. The transmitter is located closer to the lowest position with the height of 841.09m as shown in Figure 4.16 (a) and Figure 4.16 (b). As shown in Figure 4.15 (c) and Figure 4.15 (d), signal is not generally affected by the multiple hills as the transmitter is located closer to the lowest position while the receiver is located at the higher position. However, Zone F1 shows the higher path loss between the T-R pair as the signal is affected by the obstructions lying between the transmitter and the receiver located at the higher positions.

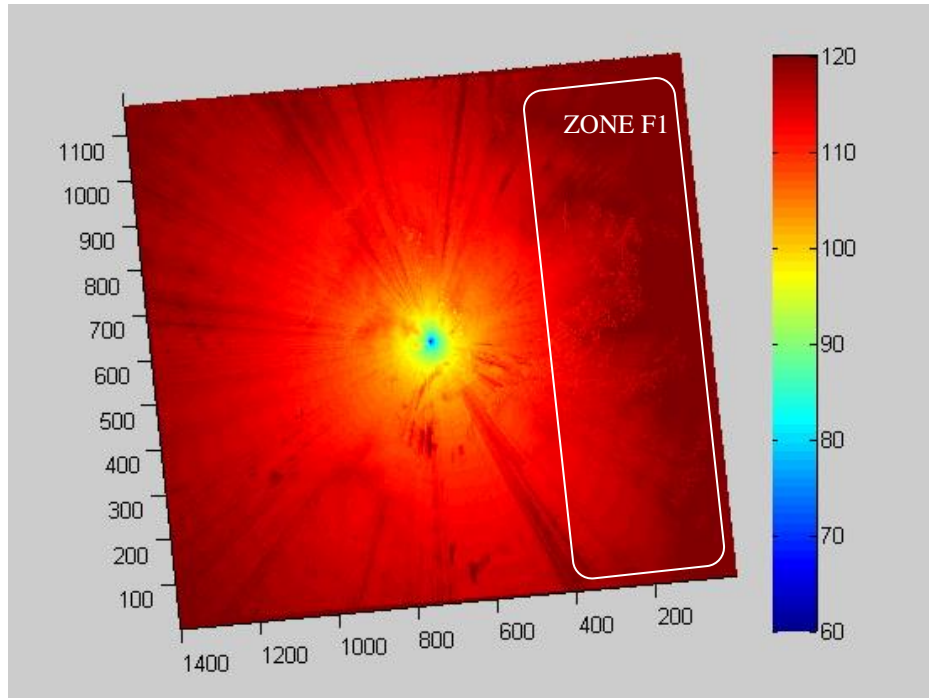




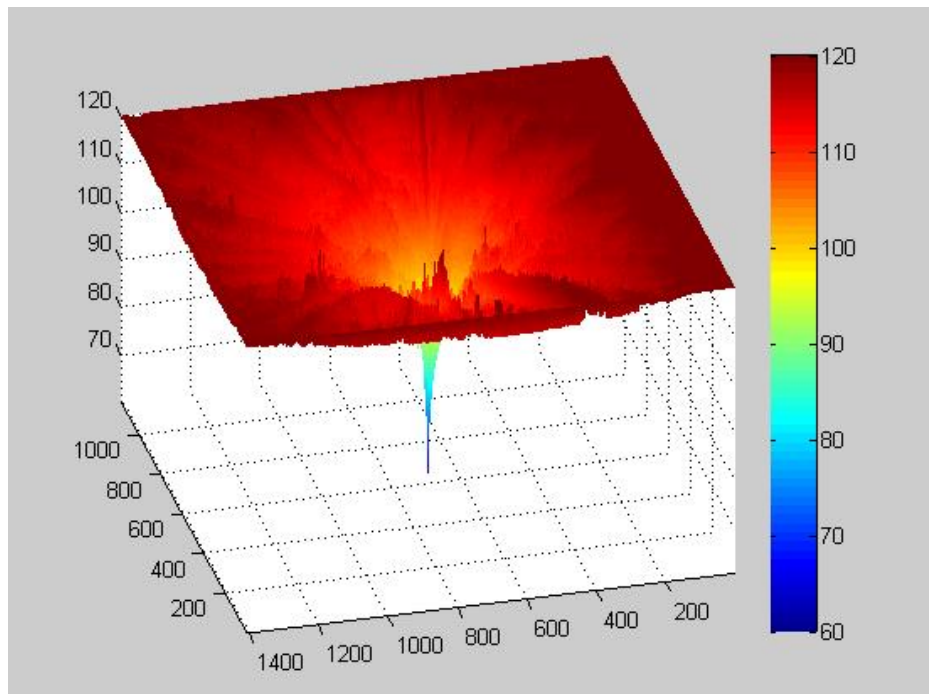
(a) 2D view of Acton terrain profile



(b) 3D rotated view of Acton terrain profile



(c) 2D rotated view of Acton's path loss simulation

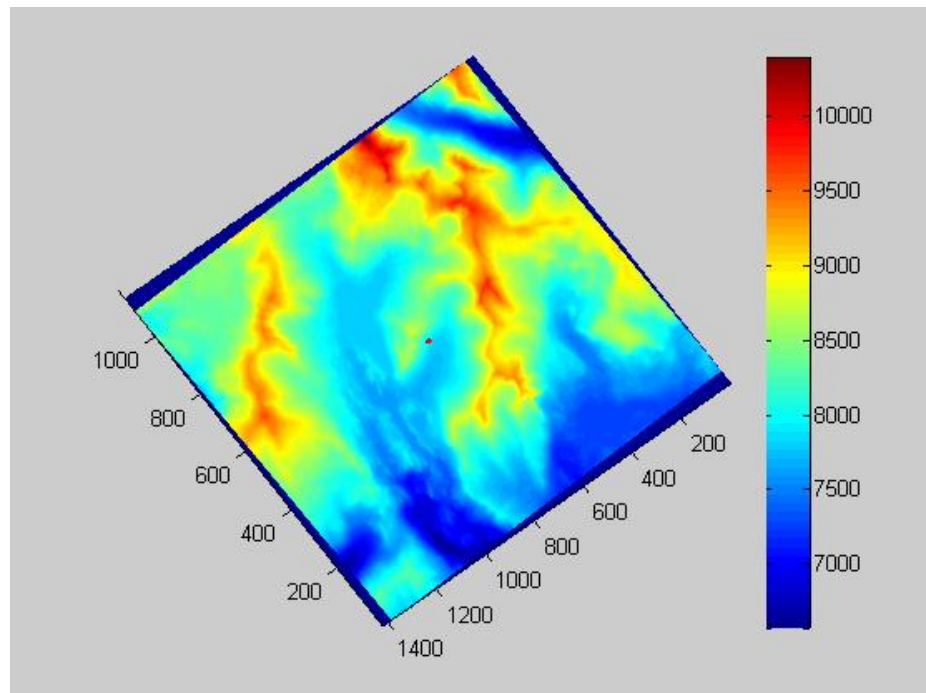


(d) 3D view of Acton's path loss simulation

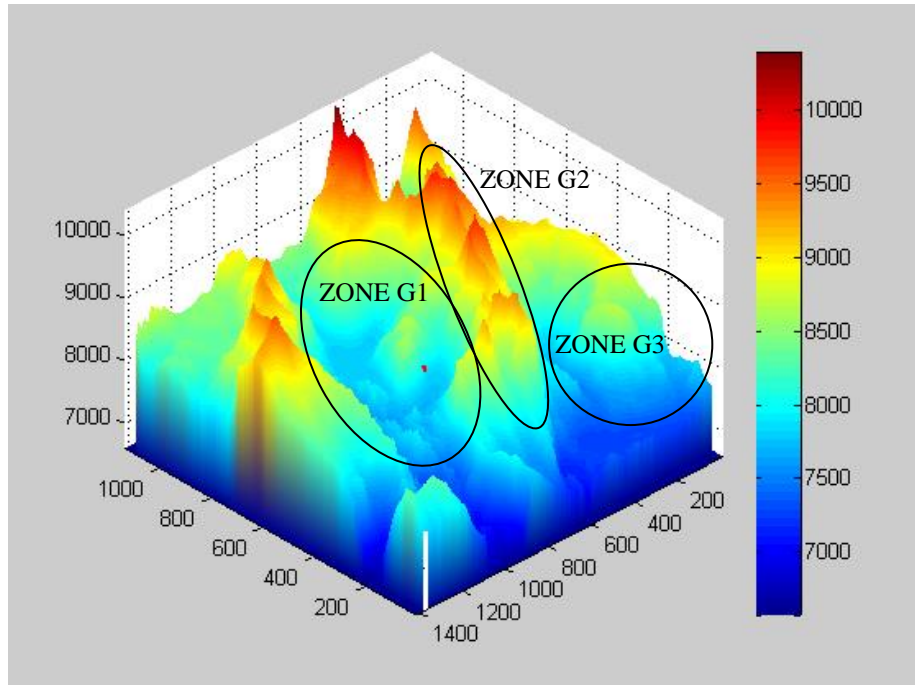
Figure 4.16: Simulation results on Acton terrain profile



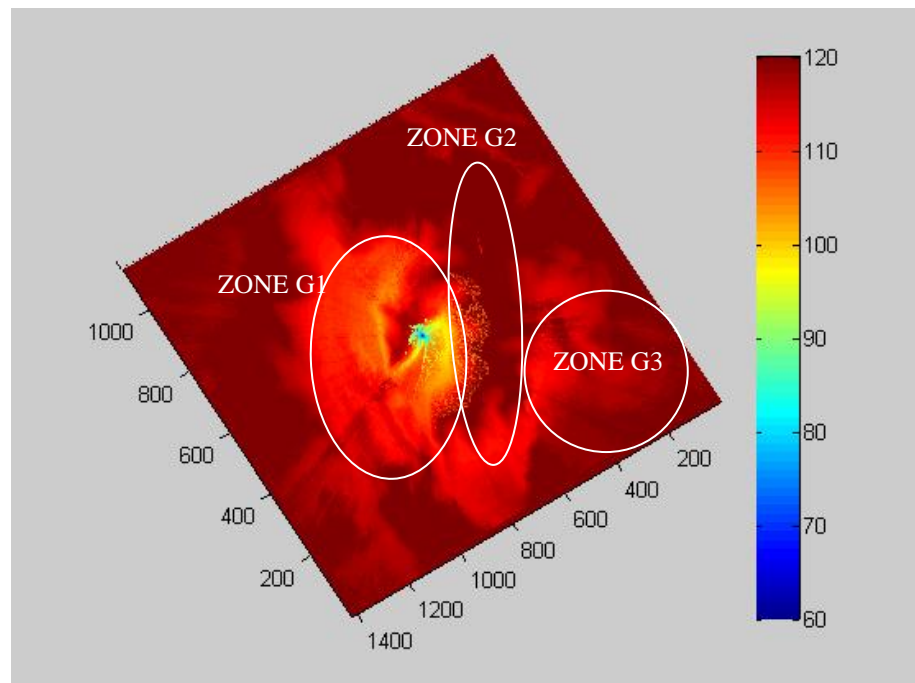
g. Caples Lake: Caples Lake is one of the terrain profiles with high terrain elevation difference. The transmitter is located at a mountain cove with the height of 7920.4m surrounded by multiple mountains as shown in Figure 4.17 (a) and Figure 4.17 (b). As shown in Figure 4.17 (c) and Figure 4.17 (d), the receiver is easily blocked in ZONE G2 and ZONE G3 as compared to ZONE G1 due to the transmitter is located nearby a few of mountains.



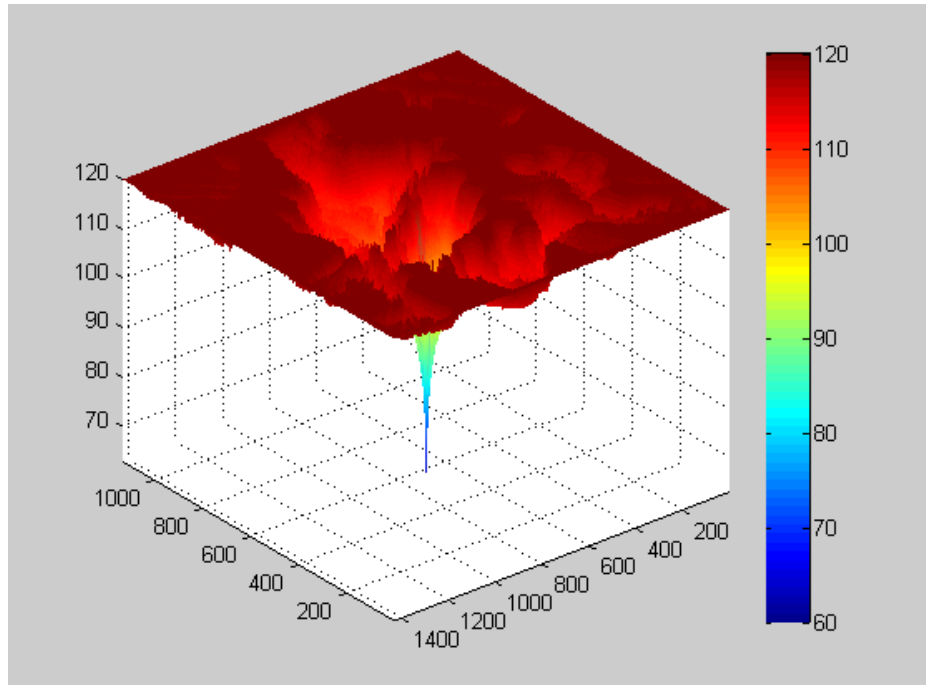
(a) 2D view of Caples Lake terrain profile



(b) 3D rotated view of Caples Lake terrain profile



(c) 2D rotated view of Caples Lake's path loss simulation



(d) 3D view of Caples Lake's path loss simulation

Figure 4.17: Simulation results on Caples Lake terrain profile

The results obtained from the analysis of path loss between the T-R pair on all seven maps further strengthen *Observation 1* and *Observation 2*. Besides, the results also suggest the following observation:

***Observation 3:*** The typical strategy that places a transmitter in the highest position may not always result a lower path loss between the transmitter and other receivers.

## CHAPTER 5

### CONCLUSION

#### 5.1 Conclusion

We have proposed an improved path loss simulation that incorporates 3D terrain model. The results obtained suggest that a more accurate path loss between a transmitter and a receiver can be predicted after considering the obstacles that block the Fresnel zone between the two nodes. These obstacles are extracted from DEM data with high resolution provided by USGS. Besides, the higher time complexity in computing the path loss after considering the obstructions is partially offset by off-loading the computationally intensive components to GPU. More precisely, a speedup of 20x to 42x is achieved by offloading the computationally intensive components to GPU where the CUDA texture memory and faster interpolation computation are exploited. These results also answer one of the open problems raised by Mittag et al. (2011) and Djinevski et al. (2015) to propose a parallelised path loss simulation framework that considers the terrain profile of the deployed communication network.

Lastly, the proposed simulation framework has been integrated as a module into NS-3 for realistic path loss simulation which considers the terrain profiles between the transmitter-receiver pair. The proposed framework can

also be exploited by radio frequency engineers for speeding up the planning and deployment of transmitters in radio network. In addition, our proposed framework allows researchers to easily implement different path loss simulation models on NS-3 depending on the terrains they want to simulate by varying the parameters since most of these models rely on similar components.

## **5.2 Future Work**

In view of the research conducted in this project, some possible interesting directions are suggested as follows:

- It would be interesting to improve the proposed terrain aware model with higher accuracy by considering more factors that may affect the path loss computation between two antennas, such as reflection.
- It would be interesting to improve the proposed terrain aware model with lower time complexity by further effectively parallelising the searching of obstacles and the smoothing of obstacles.

## REFERENCE

Abdelrazek, A.F., Kaschub, M., Blankenhorn, C. and Necker, M.C., 2009. A novel architecture using NVIDIA CUDA to speed up simulation of multi-path fast fading channels. *IEEE 69<sup>th</sup> Vehicular Technology Conference*, 26-29 April 2009. Barcelona pp. 1-5.

Amoretti, M., Picone, M., Zanichelli, F. and Ferari, G., 2013. Simulating mobile and distributed systems with DEUS and ns-3. *International Conference on High Performance Computing and Simulation (HPCS)*, 1-5 July 2013. Helsinki, pp. 107-114.

Andelfinger, P., Mittag, J. and Hartenstein, H., 2011. GPU-based architectures and their benefit for accurate and efficient wireless network simulations. *IEEE 19<sup>th</sup> International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 25-27 July 2011. Singapore, pp. 421-424.

Bello, P., 1963. Characterization of randomly time-variant linear channels. *IEEE Transactions on Communications Systems*, 11 (4). pp. 360-393.

Bilel, B.R., Navid, N. and Bouksiaa, M.S.M., 2012. Hybrid CPU-GPU distributed framework for large scale mobile networks simulation. *IEEE/ACM 16<sup>th</sup> International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, 25-27 Oct 2012. Dublin, pp. 44-53.

Bullington, K., 1947. Radio propagation at frequencies above 30 megacycles. *Proceedings of the IRE*, 35 (10), pp. 1122-1136.

Camp, T., Boleng, J. and Davies, V., 2002. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, 2(5), pp. 483-502.

Chengetanai, G. and O'Reilly, G.B., 2015. Survey on simulation tools for wireless mobile ad hoc networks. *IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 5-7 March 2015. Coimbatore, pp. 1-7.

Chong, P.K. and Kim, D.Y., 2013. Surface-level path loss modelling for sensor networks in flat and irregular terrain. *ACM T Sensor Network*, 9 (2), pp. 1-32.

Chu, X., Liu, C., Ouyang, K., Yung, L.S. and Leung, Y.W., 2015. PErasure: a parallel Cauchy Reed-Solomon coding library for GPUs. *IEEE International Conference on Communications (ICC)*, 8-12 June 2015. London , pp. 436-441.

Cook, S., 2013. *CUDA programming: a developer's guide to parallel computing with GPUs*. Waltham, USA: Elsevier.

Davies, V., 2000. Evaluating mobility models within an ad hoc network. Master's Thesis. Colorado School of Mines.

Desa, H., Sofian, M. and Zairi, S., 2009. Study of integration 2.4GHz and 5.8GHz in RFID tag. *Proceedings of the International Conference on Man-Machine Systems (ICoMMS)*, 11-13 October 2009. Batu Ferringhi, Penang, pp. 5A61-4.

Deygout, J., 1966. Multiple knife-edge diffraction of microwaves. *IEEE Transaction on Antennas and Propagation*, 14 (4), pp. 480-489.

Djinevski, L., Filiposka, S., Mishkovski, I. and Trajanov, D., 2015. Accelerating wireless network simulation in 3D terrain using GPUs. *Adhoc & Sensor Wireless Networks*, 29(1-4), pp. 253-264.

Durgin, G.D., 2009. The practical behaviour of various edge-diffraction formulas. *IEEE Antennas and Propagation Magazine*, 51 (3), pp. 24-35.

Epstein, J. and Peterson, D.W., 1953. An experimental study of wave propagation at 850 MC. *Proceedings of the IRE*, 41(5), pp. 595-611.

Fujimoto, R., 2015. Parallel and distributed simulation. *IEEE Winter Simulation Conference (WSC)*, 6-9 Dec 2015. Huntington Beach, CA, pp. 45-59



Giovaneli, C.L., 1984. Analysis of simplified solutions for multiple knife-edge diffraction. *IEEE Trans on Antennas and Propagation*, 32, pp. 297-301.

Graham, A.W., Kirkman, N.C. and Paul, P.M., 2007. *Mobile radio network design in the VHF and UHF bands: a practical approach*. UK: WILEY.

Gupta, N., 2013. Texture Memory in CUDA. [online] Available at: <cuda-programming.blogspot.my/2012/02/texture-memory-in-cude-what-is-texture.html> [Accessed 10 October 2015].

Han, S.Y., Lee, D. and Abu-Ghazalah, N.B., 2013. Double regression: efficient spatially correlated path loss model for wireless network simulation. *Proceedings IEEE INFOCOM*, 12-19 April 2013. Turin, pp. 1824-1832.

Hata, M., 1980. Empirical formulae for propagation loss in land mobile radio services. *IEEE Transactions on Vehicular Technology*, 29(3), pp. 317-25.

Hoehner, P., 1992. A statistical discrete-time model for the WSSUS multipath channel. *IEEE Transactions on Vehicular Technology*, 41(4), pp.461-468.

Hwang, T., Yang, C., Wu, G., Li, S. and Li, G.Y., 2009. OFDM and its wireless applications: a survey. *IEEE Transactions on Vehicular Technology*, 58 (4), pp. 1673-1694.

Iskander, M.F. and Yun, Z., 2002. Propagation prediction models for wireless communications systems. *IEEE Transaction on Microwave Theory and Techniques*, 50 (3), pp. 662-673.

Ivey, J., Riley, G., Swenson, B., and Loper, M., 2016. Designing and enabling simulation of real-world GPU network applications with ns-3 and DCE. *IEEE 24<sup>th</sup> International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS)*, 19-21 Sept 2016. London, pp. 445-450.

Kasampalis, S., Lazaridis, P.I., Zaharis, Z.D., Bizopoulos, A., Paunovska, L., Zettas, S., Glover, I.A., Drogoudis, D. and Cosmas, J., 2015. Longley-Rice model prediction inaccuracies in the UHF and VHF TV bands in mountainous terrain. *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 17-19 June 2015. Ghent, pp. 1-5.

Kim, H.W., Song, E.H., Park, J.H. and Jeong, Y.S., 2015. Parallel processing for separate sensor of WSN simulator with GPU. *IEEE 29<sup>th</sup> International Conference on Advances Information Networking and Applications (AINA)*, 24-27 March 2015. Gwangju, pp. 255-262.

Lee, W.C.Y., 1997. *Mobile Communications Engineering: Theory and Application*. NY, USA: McGraw Hill.

Li, J., Sun, J., Song, Y., Xu, Y. and Zhao, J., 2014. Accelerating the reconstruction of magnetic resonance imaging by three-dimensional dual-dictionary learning using CUDA. *36<sup>th</sup> Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 26-30 August 2014. Chicago, pp. 2412-2415.

Li, Z., Zhu, C. and Gold, C., 2004. *Digital terrain modelling: principles and methodology*. NY, CRC press.

Liu, L. and Ma, H., 2012. On coverage of wireless sensor networks for rolling terrains. *IEEE Trans Parallel Distribution System*, 23 (1), pp. 118-125.

Mansfield, S., Veenstra, K. and Obraczka, K., 2016. TerrainLOS: An Outdoor Propagation Model for Realistic Sensor Network Simulation. *IEEE 24<sup>th</sup> International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 19-21 Sept 2016. London, pp. 463-468.

Maor, E., 2007. *The Pythagorean theorem: a 4,000-year history*. New Jersey, USA: Princeton University Press.

Mark, J. and Zhuang, W., 2003. *Wireless Communications and Networking*. New Jersey, USA: Prentice Hall.

Miletic, V., Subic, T. and Mikac, B., 2014. Optimizing maximum shared risk link group disjoint path algorithm using NVIDIA CUDA heterogeneous parallel programming platform. *International Symposium on Telecommunications (BIHTEL)*, 16 December 2014. Sarajevo, pp. 1-6.

Mittag, J., Papanastasiou, S., Hartenstein, H. and Strom, E.G., 2011. Enabling accurate cross-layer PHY/MAC/NET simulation studies of vehicular communication networks. *Proceedings of the IEEE*, 99 (7), pp. 1311-1326.

Okumura, Y., Ohmori, E. and Fukuda, K., 1968. Field strength and its variability in VHF and UHF land mobile radio service. *Review of the Electrical Communication Laboratory*, 16(9-10), pp 825-73.

Ordiales, J.L., Fontan, F.P. and Hernado, J.M., 1994. Validation results of a GTD based propagation prediction model and comparison with conventional models of propagation. *7<sup>th</sup> Mediterranean Proceedings Electrotechnical Conference*, 3, pp. 1170-1173.

Philips, C., Sicker, D. and Grunwald, D., 2013. A survey of wireless path loss prediction and coverage mapping methods. *IEEE Communications Surveys and Tutorials*, 15(1), pp. 255-270.

Proakis, J.G. and Salehi, M., 2008. *Digital Communication*. 5th ed. New York: McGraw-Hill.

Rappaport, T.S., 2002. *Wireless Communications: Principles and Practice, Upper Saddle River*. NJ: Prentice Hall.

Rice, P.L., Longley, K.A., Norton, K.A. and Barsis, A.P., 1967. Transmission loss predictions for tropospheric communications circuits. *Technical Note 101 U.S. Dept. of Commerce NTIA-ITS*.

Ryoo, S., Rodrigues, C.I., Bagsorkhi, S.S., Stone, S.S., Kirk, D.B. and Hwu, W.M.W., 2008. Optimization principles and application performance evaluation of a multithreaded GPU using CUDA. *Proceedings of the 13<sup>th</sup> ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPOPP)*, 20 February 2008. NY, USA, pp. 73-82.

Salz, J., 1985. Digital transmission over cross-coupled linear channels. *AT&T Technical Journal*, 64 (6), pp. 1147-1159.

Shannon, C.E., 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27 (3), pp. 379-423.

Stallings, W., 2005. *Wireless Communications and Networks, Upper Saddle River*. NJ: Prentice Hall.

Stojanova, S., Djinevski, L., Mishkovski, I., Filiposka, S. and Trajanov, D., 2013. Micro-benchmarking NS-2 and NS-3 network simulators using the terrain aware radio propagation extension. *International Conference on Internet Society Technology and Management (ICIST2013)*, March 2013. Kopaonik, Serbia, pp. 81-84.

Topcu, S., Goktas, P., Karasan, E. and Altintas, A., 2015. A new approach to diffraction modelling for line-of-sight (LOS) paths. *IEEE Topical Conference on Antennas and Propagation in Wireless Communication (APWC)*, 7-11 September. Turin, pp. 696-699.

Truong, T.P., Pottier, B., and Huynh, H.X., 2016. Monitoring of environment: A high performance method for radio coverage exploration. *IEEE Radio and Antenna Days of the Indian Ocean (RADIO)*, 10-13 Oct 2016. St. Gilles-less-Bains, pp. 1-2.

Valcarce, A., De La Roche, G. and Zhang, J., 2008. A GPU approach to FDTD for radio coverage prediction. *IEEE International Conference on Communication Systems*, 19-21 November 2008. Guangzhou, pp. 1585-1590.

Vuckovik, M., Trajanov, D. and Filiposka, S., 2011. Durkins propagation model based on triangular irregular network terrain. *ICT Innovations 2010*, pp. 333-341.

Yip, C.M. and Asaduzzaman, A., 2014. A promising CUDA- accelerated vehicular area network simulator using NS-3. *IEEE International Performance Computing and Communications Conference (IPCCC)*, 5-7 December 2014. Austin, TX, pp. 1-2.

Zhou, G., He, T., Krishnamurthy, S. and Stankovic, J.A., 2004. Impact of radio irregularity on wireless sensor networks. *Proceedings of the 2<sup>nd</sup> International Conference on Mobile Systems*, 6-9 June 2004. Boston, USA, pp. 125-128.

## LIST OF PUBLICATION

Loo, Z.B., Chong, P.K., Lee, K.Y. and Yap, W.S., 2017. Improved path loss simulation incorporating three-dimensional terrain model using parallel co-processors. *Wireless Communications and Mobile Computing*. vol. 2017, Article ID 5492691, 11 pages, 2017, doi:10.1155/2017/5492691.