Efficient Computational Model for Energy Propagation in Geometrically Represented Large Environments

by

Ajay Rajkumar

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy

Department of Computer Science
Graduate School of Arts and Sciences
New York University
September 1999

Approved: _____
Professor Ken Perlin

To my family

Acknowledgment

I would like to thank all the people who have helped me in one way or another during my stay both at NYU and at Bell Labs. First of all, I would like to thank my advisor Ken Perlin who has been a source of inspiration throughout my stay at NYU. His enthusiasm to pursue ideas is very contagious. His ongoing confidence in me and support have been instrumental in bringing this work to completion. Next, I would like to thank the other members of my reading committee, Bud Mishra and Denis Zorin. Bud, besides his numerous helpful comments on my work, has also been very generous with his advice during my years at the department. Denis gave me some very insightful and important comments during the writing of this dissertation. I would also like to thank Alan Siegel for his encouragement, and advise at some key moments during my stay at the department.

At Bell Labs, I would like to thank Bruce Naylor and Farid Feisullin who introduced me to the transmitter placement problem. Bruce also gave me some of my early lessons on

rigorous problem solving techniques. It was a pleasure working with him. Farid provided some early insight into various issues from the wireless communications perspective. He has throughout been a very big supporter of this work. He was always bringing these teams of technical and business people from all over the world to see a demonstration of the system and one persistent question I was never able to answer was: when do we plan to bring this work to the market?

I would also like to thank all my colleagues over the years for creating very stimulating environments to work in. They have contributed to the many facets of my experiences. I would also like to thank all my friends including Suren Talla, Anthony Triolo, Raoul Daruwala, Kanna Rajan, and numerous past and present officemates for all the fun discussions we have had.

A special thanks go to my brother Sudhir, who has been a *de facto* guide, mentor, and a strong motivator. My utmost gratitude goes to my wife Aradhana for all her love, understanding, and patience during the long hours that I was away. Finally, I would like to thank my parents, my in-laws, my sister Rekha, and the rest of my family, for their unconditional love, support, and belief in me. I just wish I had finished this work early enough for my mom to have seen this day.

Abstract

Current radio propagation algorithms are very narrowly focused to specific types of input models and do not scale well to an increase in the number of receiver locations or the number of polygons in an input model. In this dissertation, we look at the problem of efficiently computing energy propagation at radio frequencies in a range of geometrically defined environments and for various transmitter and receiver characteristics. To achieve this goal, we propose a unified approach to radio propagation for different types of input models and their combinations as well, by representing the geometry as a binary space-partitioning tree and broadcasting energy from the source. The approach is both scalable to large input models as well as dynamically adapts to its scale without incurring unreasonable computational cost. The proposed approach is equally effective for acoustic modeling as well.

We present a new adaptive ray-beam tracing algorithm which initially tessellates the

surface of a transmitter into four-sided polygons. Each polygon is cast as a beam which avoids arbitrarily large gaps or overlaps between adjacent beams. For fast intersection computation each beam carries information of its medial ray as well. As the computation proceeds a ray-beam is adaptively subdivided depending on various parameters such as the projected area of a beam, its current energy level, and the distance between adjacent receiver locations. The proposed algorithm has sublinear-time complexity in terms of the number of receiver locations.

Modeling diffraction off an edge of a wedge is important to compute radio signal that reaches the shadow region of the wedge. Storing these edges explicitly in a data structure can be very expensive for large input models and especially for terrain-based models that have significant elevation variations. We present a new runtime edge-detection algorithm instead of storing the edges statically and its adaptation to environments represented as binary space-partitioning tree.

We have developed a propagation prediction system called *Propagate* using these algorithms with good statistical correlation between predicted and measured results for a number of different input models. The proposed algorithms have been used to model several other important computations related to a cellular network of transmitters such as signal strength and path loss, delay spread, angular spread, carrier-to-interference ratio, and modeling of different antenna diversity schemes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The goal of this dissertation is to develop a unified approach to simulate radio propagation from a given transmitter location in a range of geometrically defined environments. By a unified approach, we mean that rather than having specifically tailored algorithms that are applicable to a narrow class of geometric environments and specific characteristics of transmitters and receivers, the same set of algorithms should be applicable to a large class of environments and other associated parameters. Such a unified approach is not only important in simulating radio propagation for a wider range of receiver and transmitter parameters but also enables propagation simulation from inside buildings to street level propagation and vice versa.

Despite the extent of research on efficient algorithms, current radio propagation algorithms are both very narrowly focused to the specific needs of the type and size of geometry in an environment as well as do not scale well to large environments. For exam-

ple, an algorithm that simulates radio propagation in urban areas with possible receiver locations at a constant height above ground is not equipped to efficiently simulate radio propagation inside buildings where receivers could be located at several heights on different floors. Similarly, algorithms for propagation inside buildings or urban areas cannot simulate propagation in terrain-based environments that have significant local variation in elevation such that receiver locations have to be modeled at continuously changing height rather than at a constant height. The standard algorithms for radio propagation are also not equipped to model diffraction efficiently in terrain-based environments that can potentially have a very large number of wedges.

Efficient algorithms for radio propagation in different types of environments are also key to modeling a number of parameters for a cellular network of transmitters. In our research, we seek to develop algorithms that are computationally efficient as well as accurate within certain acceptable bounds as compared to actual measurements from a field trial. We represent a geometric environment as a binary space partitioning tree which is both efficient as well as facilitates interactive visualization of an environment.

The main contribution of this dissertation is a generalized approach to radio propagation in all types of environments and its associated algorithms. In the process, we analyze and develop algorithms that broadcast energy from the source. The proposed approach, as we will show in this dissertation, is computationally efficient and can simulate different characteristics of transmitters and receivers as well. We also develop other key algorithms to support the above approach such as the hybrid ray-beam tracing al-

gorithm, the adaptive ray-beam subdivision scheme, and a runtime algorithm to detect edges for diffraction. Finally, using the proposed algorithms we discuss the modeling of several important computations related to a cellular network of transmitters such as path loss, delay spread, angular spread, carrier-to-interference, and the modeling of various antenna diversity schemes. The proposed algorithms can easily be used to model other forms of energy propagation such as acoustic wave propagation. We have developed a library of functions based on these algorithms and have implemented an interactive radio propagation prediction system called *Propagate* using this library.

1.1 Motivation

A desire for ubiquitous wireless communication has lead to a tremendous increase in demand for wireless services in all types of environments. These environments range from dense urban and semi-urban environments to large indoor buildings and rural environments. The increase in demand has created large mobile and fixed traffic densities in most areas. A popular method of alleviating user congestion in these areas is to place more transmitters in the same geographical area. As a result, better placement planning tools are desired for positioning these transmitters in close proximity such that there is continuous coverage while limiting inter-transmitter interference. These requirements have made it imperative for wireless service providers to understand the coverage and distribution of electromagnetic energy (in the radio frequency range) from a specific transmitter location (or a set of locations). It is also important to understand energy distribution both from

the perspective of range extension of the coverage area, as well as enhancement of user capacity in a network.

Traditionally, wireless service providers have conducted elaborate field trials to measure coverage and distribution of signal strength at a specific site. An interactive system that efficiently computes the spatial distribution of signal strength in a given environment can achieve this at a much lower cost. Such a system is important for several other reasons. First, for a given environment it is desirable to get maximum coverage with the least number of transmitters. Besides reducing the overall installation cost of a wireless network, this is essential to reduce co-channel as well as adjacent-channel interference from other transmitters. An interactive system can achieve this more quickly than an actual field trial. Second, since transmitters have to be installed at locations that are easily accessible to a service provider, optimal placement from the coverage point of view may not necessarily be an optimal position for accessing a location. However, an interactive system can help a user assess the trade-off between an optimal placement and an accessible location. Third, measuring the propagation effects through field trials for every environment where service has to be provided can be a prohibitively expensive and time-consuming process. An efficient interactive system can again provide an effective solution. Finally, not only can such an interactive system facilitate greater understanding for optimal placement of transmitters, it can also be an indispensable platform to perform a variety of antenna diversity simulations and other radio frequency (RF) engineering simulations.

Successful wireless communication between two locations in space is based on a simple

principle that an acceptable level of signal reaches a receiver location from a transmitter location. An acceptable signal could reach a location simultaneously through a number of different paths and in a number of different ways. These different paths themselves can be of different lengths. Though different path-lengths result in a delay amongst signals that reach the same location, the problem is still of immense interest in wireless communications, since the delay can be used to an advantage by combining them (adding the signal intensity) in several mathematical ways. The above problem of multiple paths reaching a receiver location from the same transmitter has been termed in literature as a *multi-path problem*. Some of the ways in which a signal can reach a location are direct line-of-sight, scattering off a surface (including specular reflections), transmission through a surface, and through diffraction off the edges of an object. It is worth noting that although all frequencies exhibit diffraction phenomenon (that is, a ray bending around corners), at cellular frequencies (900 MHz to a few GHz) diffraction is particularly important for signal propagation. The reason is that in a large number of scenarios, diffraction is the only means by which a signal reaches a shadow region. Cellular frequencies exhibit diffraction due to the comparative scale of objects with respect to the wavelength of rays. In environments that have few obstructions between a transmitter and a receiver, this problem of computing signal distribution from a source is computationally trivial. But for large real-life environments that have millions of surfaces[1] (with different shapes, sizes,

[1]We define a surface as a connected region with the same surface normal. In literature, this has sometimes been referred to as a polygon. But we want to draw a distinction between a naturally occuring surface (polygon) and polygons that are a result of a subdivision process of an algorithm such a finite element mesh for radiosity algorithms.

and orientations) and tens of millions of receiver locations, the problem of computing received signal strength easily turns into a computationally intractable one.

The key to an interactive system is twofold: an efficient and accurate algorithm for propagation of electromagnetic energy in an environment and an efficient data structure to represent the same geometric environment. The energy propagation problem can be stated as a ray tracing problem in a three dimensional optical system. But at the same time a naive adaptation of ray tracing does not scale efficiently for computing radio frequency power distribution in large environments.

The two main problems that manifest themselves are as follows. For a large cluttered urban environment or a large multi-floor building the number of receiver locations at which the received power needs to be sampled can be very large, almost on the order of tens of millions of locations for uniformly spaced locations. A ray tracing based algorithm that traces rays from a receiver, though linear in terms of the number of receiver locations, is still extremely inefficient for a very large number of receivers. The reason is that the density distribution of objects in an environment is not known *a priori*. However, the power from a transmitter attenuates based on the density distribution of objects around a transmitter. Thus, from the outset there is unnecessary computation being performed for a large number of receiver locations. These receiver locations could either be at a large distance from an energy source such that the signal from the source attenuates considerably before reaching it or the receiver may be completely occluded by objects in the environment.

Therefore, the first problem is to devise an approach such that no wasteful computation is performed for any set of receivers, that is, the algorithm performs better than a linear time algorithm in the number of receiver locations. Another problem is to devise an approach to sample an environment in sufficient detail such that it does not miss any objects and yet does it efficiently. Finally, the algorithmic approach to propagation should be general enough that it does not pay a penalty for one type or scale of environment to achieve efficiency for another type of environment. This is important for scenarios where it is desirable to study propagation distribution from one type of an environment to another type of environment.

In this thesis, we will discuss an algorithmic approach which is both efficient and accurate. The approach has been used to build an RF propagation prediction system for wireless communication applications. We will also discuss its algorithmic complexity based on an empirical study for sample datasets, as well as advantages and disadvantages of such a system.The basic contribution of this thesis is a unified framework to simulate and study radio propagation effects in geometrically represented indoor, outdoor, terrain-based, and any combination of such environments. It also contributes the associated key algorithms such as the adaptive ray-beam tracing algorithm; the run-time edge-detection algorithm for diffraction in three-dimensional environments; and against conventional wisdom, the thesis shows that the algorithmic approach of broadcasting energy from the source is efficient as compared to an image-based approach.

1.2 Prior Work

Energy propagation has been a subject of research in many domains. These research domains range from specific areas for different parts of the electromagnetic spectrum to areas such as acoustics where energy propagates in the form of transverse pressure waves. In the early days, to study propagation characteristics, small experiments were setup in controlled conditions. These characteristics were then extrapolated to get results for general conditions. To complement the experimental results, several analytical and numerical approaches were also formulated. With the advent of high speed computers, many of these approaches have been replaced with sophisticated computer simulations. Initially these simulations have to be validated against actual experimental or theoretical results for a certain class of problems. But then many problems for that class of problems can easily be analyzed with a much smaller cycle-time as compared to an actual experiment, which can be both time consuming and costly.

Electromagnetic energy ranges from the shortest gamma rays with a wavelength less than a femtometer (10^{-15} $meters$) to the longest radio waves with a wavelength of a few kilometers (10^{3} $meters$) and greater. Similarly, acoustic waves can range from a few centimeters to tens of meters. As we will see later, a certain subset of this range, $20\,kHz$ to $20\,Hz$ is of immense practical interest. Let us initially map the major research areas that have spawned over the years to deal with specific parts of the electromagnetic spectrum. In the order of increasing wavelength, these research areas are: photon transport for gamma rays; thermal radiation and heat transfer for infrared and ultraviolet rays; optics

in the form of image synthesis for visible light; and radar, wireless communication, and remote sensing applications for radio waves. Similarly, acoustic wave (in the range of human hearing) modeling has been researched for better audio systems, better acoustic modeling for auditoriums, and recently to model sound propagation in virtual reality simulations. At the same time, another perspective comes from the particle transport schemes such as neutron transport.

It is important to note that each of the above domains addresses energy propagation from a different perspective. This perspective (such as a computational model) typically depends on the motivation for seeking such an understanding. One of the important characteristics of propagation is the wavelength[2] with respect to its surrounding environment. This has a direct impact on the solution being sought for a particular domain. The next few sections in this chapter will outline the solutions proposed to understand the modes of propagation in some of these major research areas. Within each section, we will initially give a broad outline of the area itself and then describe the significant strategies pursued in that area.

1.2.1 Propagation schemes for neutron transport

The roots of the transport theory can be traced back to *the Boltzmann equation* proposed almost a century ago. A real interest in neutron transport problems first came about in

[2]Wavelength given by λ is proportional to the inverse of frequency f and is defined as the distance between two successive peaks of a periodic wave. For electromagnetic waves the relationship is given by $\lambda = c/f$, where c is the speed of light in vacuum given by $2.9979 \times 10^8\ m/s$. Similarly for acoustic waves $\lambda = c'/f$, where c' is the speed of sound in vacuum given by $330\ m/s$.

the 1940's with an interest in nuclear chain reactors. Since then several analytical and numerical solutions have been proposed to solve the problem. Most analytical methods consider simple geometries resulting in limited usage for practical applications. With an increase in the availability of computational power, numerical methods have gained popularity [35].

To derive the transport theory of neutrons several assumptions are made. Some of these assumptions are the particle behavior of neutrons and neutrons traveling in straight lines in a single medium. Notice that these are similar to the assumptions made for light transport as well. On the other hand, certain assumptions are different. For example, neutrons travel much further into a solid medium as compared to light waves. As a result, volume scattering has to be considered for neutron transport simulations. Some assumptions are made for neutron particle interaction with its surrounding media. Typically, for both neutrons as well as visible light simulations the medium in which the particles travel can be considered as participating or non-participating depending on the context of the simulations. Particle-to-particle interaction for neutron transport can be neglected only if the density of matter is not high. For optical light simulations, though most of the time it is assumed that the media is non-participating, phenomena such as fog, mist, etc., could constitute as participating media and modeled accordingly.

One of the principal differences between the two types of particles is the way in which principle of conservation of energy is applied. For light photons, this law is strictly followed in the sense that the emitted energy is always considered to be less than or equal

to the incident energy (depending on whether the surface on which energy is incident is lossy or lossless). Whereas, for neutrons the emitted energy may be critical or supercritical depending upon the assumptions of their interaction with the surrounding media.

At the computational level both deterministic and non-deterministic methods of simulating propagation have been suggested for neutron transport. Most of the deterministic solutions are numerical in nature. An important non-deterministic model proposed is in the form of a stochastic method called the *Monte Carlo* method of computation. Unlike the deterministic computational method where the error is systematic, non-deterministic methods use various probability distribution functions to cast a finite number of particles in the environment and trace their paths. The accuracy of the results is controlled by increasing the number of particles, which is achieved at an additional computational cost. Simulations can be built such that this accuracy can be dynamically controlled based on some predefined error functions. Monte Carlo methods have been used in several other propagation applications. We will discuss applications of Monte Carlo methods in greater detail while discussing light transport simulation for global illumination.

1.2.2 Thermal radiation and heat transfer

The primary means of thermal radiation in an environment are convection, conduction, and radiant transfer. For both convection and conduction a physical medium is necessary, whereas, radiant transfer can take place in a vacuum too. Radiant energy propagation can be studied both from the point of view of the classical wave theory as well as quantum

mechanics. In most cases, both these approaches yield similar equations. However, areas where spectral distribution of energy has to be studied, the phenomenon can be better explained using quantum theory by simulating energy as discrete particles [57]. On the other hand, wave theory provides a very strong theoretical framework for the study of electromagnetic radiation. In this framework, electromagnetic radiation follows the laws for transverse waves oscillating perpendicular to the direction of the motion. If we assume that the propagating medium is infinite, homogeneous, and isotropic then the wave theory equations as applied to the electromagnetic waves propagating in presence of a perfect dielectric medium can be stated as

$$\mu\gamma\frac{\partial^2 E_y}{\partial t^2} = \frac{\partial^2 E_y}{\partial x^2} \quad and \quad \mu\gamma\frac{\partial^2 E_z}{\partial t^2} = \frac{\partial^2 E_z}{\partial x^2} \tag{1.1}$$

where, γ is the electrical permittivity, and μ is defined as magnetic permeability of the medium. These wave equations govern the propagation of E_y and E_z (in terms of electrical and magnetic components) in the x direction. For perfect dielectrics there is no attenuation due to the medium. Wave theory is also used to study electromagnetic propagation in other media such as imperfect dielectrics (poor conductors), and metals (good conductors) where waves attenuate within the medium due to absorption of energy. The laws of reflection and refraction have also been derived from the wave equation which are similar to the results obtained through geometrical optics.

1.2.3 Light transport modeling

The theory for light transport was primarily developed during the study of optics to design optical systems also known as geometric optics. However, to model light transport, algorithmic research has actually flourished only in the last two decades, when computer scientists started looking at ways to synthesize visually appealing images. This eventually lead to the emphasis on not only modeling visually appealing images but modeling them correctly as well. Several sophisticated algorithmic approaches have been developed, which we will survey in Section 2.1. Though the theory of geometric optics was developed using a set of canonical geometries, the computer graphics community has used the theory on complex environments having a combination of these geometries.

Geometrical and physical optics

The basic motivation for the development of geometrical and physical optics was the study of propagation of light rays and their behavior upon moving from one medium to another. The main assumption was that rays have infinitesimally small cross section and travel in straight lines known as *path of least distance*. Tracing the path of such rays is known as *ray-tracing*. The first use of ray tracing as a technique to study optical design is mentioned as early as 1929 [12].The method can easily be applied to radar and wireless communication applications for higher frequencies as $\lambda \rightarrow 0$.

The basic idea is that when a ray interacts with a discontinuity in the medium (in the form of a surface), either the ray is reflected back into the hemisphere from where it

originated, with the surface forming the dividing plane, or it gets refracted (also referred to as transmitted) to the other side of the surface. These two ways are known as reflection and refraction respectively. Depending on the two interacting media, both reflection and refraction can take place simultaneously with part of the initial energy reflected and the another part of the initial energy refracted. Besides reflection and refraction, certain material characteristics and surface conditions may also absorb a portion of the incident energy.

Reflection, the main form of scattering in optics, can be *specular* or *diffuse*. If the angle of reflection θ_r that the ray makes with the surface normal is equal to the angle of incidence θ_i, that is,

$$\theta_r = \theta_i \tag{1.2}$$

then we get specular reflection, also known as *Snell's law of reflection.* Diffuse reflection is typically modeled by using a distribution of reflection directions around the point of contact of the incident ray.

Refraction is the main form of *transmission* through a surface. A ray bends from its straight line path as it enters from one medium into another. This bending of a ray is a function of the incident angle and η, the refractive index of one medium with respect to the other. This relationship is given by

$$\sin\theta_t = \frac{\eta_i}{\eta_t}\sin\theta_i \tag{1.3}$$

where, η_t and η_i are respectively the refractive indices of the medium to which the ray is

transmitted and the medium from which the ray is incident. The equation is known as *Snell's law of refraction.*

The amplitude and phase of both reflected and transmitted rays can be computed using the above formulation. The square of the ratio of the reflected and incident energy is referred to as the *Fresnel's reflectance.* If we assume that no energy is being absorbed by the medium then because of the law of conservation of energy the *Fresnel's transmittance* can be computed by subtracting the reflectance from unity.

A limitation of geometric optics is that it assumes that a specular point exists on the surface of the object. If a surface is illuminated by a spherical wave (as opposed to a ray), the radius of curvature of the reflected wave is a function of both the radius of curvature of the incident wavefront and radius of curvature of the body at the specular point. Since the body cross section is dependent on its radius of curvature, the theory can compute spatial variation in the signal for geometries whose radii of curvature is defined. However, it fails if either or both radii of curvature are infinite, that is, if either the wave or the surface is planar or both. Physical optics, on the other hand is able to handle the above special case where geometric optics fails. The solution is approximate if diffuse scattering is considered at the incident point on the surface. For image synthesis applications, the computer graphics literature discusses several heuristic distributions that have been used.

1.2.4 Radar systems

A Radar[3] is a device that transmits electromagnetic energy and detects it by interpreting the received energy patterns. These patterns are generated when incident energy bounces off an object and gets scattered in the direction of the receiver. Radar technology was introduced in the mid 1940's. It was initially developed to substitute visual target detection since radio waves attenuate much less through the atmosphere as compared to light waves, especially in presence of clouds or at night. At lower frequencies radio waves are also known to propagate much beyond the visible horizon. This facilitates target detection much before it is optically visible. Since then radars have gained wide popularity and usage.

There are several types of radar depending on its signal modulation. *Pulsed radar systems* send a burst of energy and then listen to the echo returned by a target. *Continuous wave radar systems* (CW) vary the frequency of the transmitted signal and measure the range of frequency between the transmitted and received signal. One of the simplest forms of CW radar is the moving target indicator (MTI) radar used by the police force to track moving vehicles. The above list of radar systems is by no means exhaustive. There are innumerable variations of the above systems to suit different applications.

Radar systems range from low-frequency radars which are used for long range surveillance to high-frequency radars used for short range surveillance systems. Typically, radars operate over frequency bands from 3 MHz to 300 GHz (IEEE Standard 521—1976). For

[3]An acronym for *radio detection and ranging.*

long range surveillance, mostly VHF (30 MHz – 300 MHz) and UHF (300 MHz – 1000 MHz) frequency bands are used [30]. For other civilian and military applications, radars use higher frequency bands called the *microwave frequency bands*. These higher frequency bands are also referred to by the letters L, S, C, X, and K_u. Some civilian applications of radar technology are: weather detection and tracking, automobile speed detection, collision avoidance, and buried object detection. Military applications range from general surveillance systems to early warning systems.

Although radio waves range over a wide spectrum of frequency bands, wireless communication applications are primarily concerned with the issues of propagation related to wavelengths of a few millimeters to less than a meter. These correspond to personal communications services (PCS) at 2 GHz to cellular range of frequencies at 900 MHz. It is interesting to note that at wavelengths as small as a few millimeters trees, leaves, and vegetation in general start to have a significant impact on the attenuation of radio waves and cannot be ignored while modeling propagation. For now we will discuss approaches taken for radar applications in other segments of the radio spectrum. However, as we will see, some of these approaches are also applicable to wireless communication.

Approaches to the study of propagation for radar systems

A method of radio propagation primarily depends on the wavelength of the waves with respect to an interacting body. At low frequencies, if the wavelength is much greater than the body, there is little phase variation of the incident wave over the spatial extents of

the body, resulting in *Rayleigh scattering*. If the wavelength is comparable in size to the interacting body, propagation is either due to surface scattering or specular scattering. For higher frequencies where the wavelength is much smaller than objects it interacts with, the method of radio propagation is similar to optical frequencies. Most of the energy gets propagated by specular scattering, with angle of incidence equal to angle of reflection. Some other mechanisms for high frequency propagation are diffraction, multiple reflections, and end region scattering. Some approaches used to study the effect of propagation from each of these scattering mechanisms are:

- Analytical approach
- Numerical approach
- Geometrical approach

Analytical and numerical approach

Most of the analytical and numerical methods hinge on the theory of electromagnetics as unified by Maxwell. It states that electromagnetic waves propagate with the speed of light in a vacuum and are a combination of electric and magnetic fields. The two fields are orthogonal to each other and are also orthogonal to the direction of propagation. These equations define the relationship between the propagating (as compared to standing waves) electric and magnetic quantities of a wave source and its surrounding field. Maxwell's equations can be used to compute composite solution that include all modes of scattering.

The analytical solutions use the classical separation of variables technique to solve the second-order partial differential equations. But these are restricted to simple bodies whose surfaces coincide with the coordinates of the system in which the wave equation is being solved. For example, a sphere or a cylinder whose surfaces can coincide with a Spherical coordinate system. This restricts the usage of these solutions to a set of objects with not many practical geometries. However, analytical methods still provide an important tool for benchmarking results from other approaches [30].

The numerical approach to Maxwell's equation gives us an exact solution. The equations can be solved in an integral or a differential form. One method of solution to the integral equation formulation is known as the *Method of Moments*. Though in principle, the formulation can be applied to any frequency range, it is more suitable for low and resonant frequency ranges. For high frequencies, the scattering effects of propagation are sufficiently localized such that a general solution is unnecessary. At the same time a direct solution to Maxwell's equations can be computationally very expensive especially for complex geometries. One of the methods for solving the differential form is known as the *finite difference time domain* technique and is more useful for structures with varying dielectric and magnetic properties and with interior cavities. It may be noted that several other methods exist to solve these equations.

Geometrical approach

The above two numerical methods are powerful for the computation of final results, but fail to isolate the effects from individual scattering processes like specular and diffuse scattering, and edge diffraction. One, therefore, needs a different approach to study scattering processes individually. The approach of geometric and physical optics as discussed in Section 1.2.3, and wave optics provide us with such alternatives. It is important to note that geometric optics and in some cases wave optics can also be applied both analytically as well as numerically. Extensions to the classical geometric optics have also been proposed in the form of geometric theory of diffraction.

Wave optics: Wave theory can be described both in terms of the scalar wave theory as well as vector wave theory. The main difference between the two is that for vector wave theory an incident wave is defined in terms of both the electric and magnetic components. Waves have further been classified as planar waves, spherical waves, and harmonic waves.

Geometric and uniform theory of diffraction: Neither geometric nor physical optics deals with the sharp edges formed by halving a doubly infinite plane. To account for energy distribution around such edges extensions were proposed to geometric optics. Sommerfeld [60] proposed extensions for two-dimensional geometries to model diffraction at the edges. However, Keller [28] was the first to suggest a solution for such edges by proposing the geometric theory of diffraction (GTD) for a three-dimensional system, where the incident signal propagates all around the wedge formed by the two doubly in-

finite planes. The theory was sufficient for most cases but it still had some shortcomings. Since the diffraction coefficient was a result of wide angle evaluation, there were singularities in the coefficient as a ray approached a shadow boundary or a reflection boundary. To overcome the singularity problem of GTD, Kouyoumjian and Pathak [31] developed the uniform geometric theory of diffraction (UTD). The main idea was to introduce Fresnel's integral term in computation of the diffraction coefficient such that the resultant coefficient remained finite throughout the region around an edge. We will discuss this in greater detail in Section 5.1.1.

1.2.5 Acoustic wave modeling

Sound waves are pressure waves that propagate in a fluid medium[4]. One of the major differences between electromagnetic waves and sound pressure waves is the speed at which they travel through a medium. As mentioned earlier the wavelength of these waves is both a function of the speed of these waves in a medium and their frequency. From the human perspective acoustic waves in the range of human hearing are of most interest. These approximately range from 15 millimeters to 15 meters ($20\,KHz$ to $20\,Hz$) and are again comparable (as in the case of wireless communication for cellular and PCS frequencies) to the geometric objects found in our surroundings. For the same reason they exhibit significant diffraction phenomenon. Some of the classical examples of engineering acoustic applications for modeling of outdoor sound propagation are environmental noise

[4]Air is considered to be fluid as opposed to a solid medium. In a solid medium these waves are called *elastic waves*.

pollution, transportation noise, plant and factory noise, etc. The theory of geometrical acoustics is similar to the geometric optics and most of it can easily be applied directly to model acoustic waves. As one can notice there are many similarities between the type of issues that have to be addressed by radio wave propagation modeling and acoustic wave modeling. We feel that most of the algorithms that we will discuss in the coming chapters for radio propagation can directly be applied for acoustic wave modeling. In Section 2.4, we will discuss some application dependent differences between modeling radio propagation and acoustic propagation and their impact on the algorithmic approaches used.

1.3 Organization of the thesis

The thesis is organized as follows. Chapter 2 contains a review of the related work on propagation algorithms in application areas such as wireless communication systems and computer graphics. The basic theory on free space power attenuation along with different modes of propagation including scattering off surfaces, transmission through surfaces, and diffraction off the corners of objects are discussed. Special emphasis is given to types of approaches that have been proposed and their relevance to different application domains. It contains a discussion on the deterministic vs. non-deterministic modeling for propagation systems. A section is devoted to a comparison between ray-tracing based methods and radiosity based methods and their advantages to specific application domains.

Chapter 3 gives an overview of the types of geometric data structures that can be used to represent different types of environments. It also discusses their relevance to the operations performed on these data structures. For any simulation based on geometric databases, a great deal of accuracy of the simulation depends on obtaining such databases. Methods of obtaining geometric information along with importing such information in the system is also described.

Chapter 4 contains the description of the ray-beam tracing algorithm implemented for prediction of RF propagation in different environments. It describes the dynamic beam splitting algorithm that adapts to the scale of the environment. The algorithm subdivides a beam into child beams based on various criterion such as area projected by a beam and the level of power a beam can contribute. The modeling issues related to the broadcast mode of propagation are described. The issues of sampling error associated with classical ray-tracing algorithms are discussed.

Diffraction is an important means by which power is received in many shadow regions where power does not reach through any other means. Chapter 5 first discusses this importance of diffraction in wireless communication along with its theory and different types of diffraction coefficients that can be computed. Then we describe a new runtime edge detection algorithm for diffraction in three-dimensional partitioning-tree represented environments instead of storing these edges statically in a data structure.

Chapter 6 contains a discussion on several system considerations for wireless communications. It also contains a brief description of our radio propagation prediction system

along with a list of various receiver, transmitter, and environment parameters it considers.

Chapter 7 contains overall results of the comparison of the prediction system with actual measured data from field trials in the same environments. A discussion on the measurement taking process in the field helps in understanding the usefulness of such a prediction system. Several RF engineering applications and parameters can be computed based on such a prediction tool. We discuss computation of some of these parameters.

Chapter 8 contains a conclusion and some directions for future work in the area.

Chapter 2

Related Work

In this thesis we primarily address the issue of efficient energy propagation algorithms in large geometrically defined environments and their applications to RF engineering tools. There have been several areas where such algorithms have been proposed. Some of the foremost areas are computer graphics for image synthesis, local, and global illumination; radio propagation for wireless applications; and acoustic wave propagation for better acoustic designs of auditoriums, theaters, and lately for navigation with auralization in virtual environments. All these areas have one thing in common, which is that the basic energy transport theory stems from geometric optics and its extensions. However, assumptions as well as special requirements of each application introduce enough variations to the basic problem such that there are significant differences in solutions. In this chapter we will survey some of this related work and discuss how it has influenced our work.

2.1 Propagation algorithms for image synthesis

In computer graphics, light propagation algorithms have been actively researched for the past two decades. The questions that computer graphicists have tried to answer are: can we recreate images of the scene which the human visual system so naturally perceives day in and day out? What is necessary to model realistic or visually appealing images? In raising most of these questions, the main motivation for the computer graphics community has been to synthesize photo-realistic images. This emphasis has lead to the study of local and global illumination effects from various types of light sources. This has also lead to the question: how does a part of the electromagnetic spectrum (called visible light) interact with its environment? Though the early ideas in computer graphics came from discussions in the literature of thermal radiation and heat transfer [57], several innovative ideas have since been pursued to model light transport.

From an image generation perspective, the simplest way in which a scenario can be defined is that it consists of multiple light sources, a viewing position (that defines the viewing plane), and a set of objects. Several levels of complexity can be added to this scenario. This complexity may be in terms of different types of light sources, the sheer number of objects or minute details of individual objects, and finally in terms of the objective of view independent images as against view dependent images. Light sources themselves can manifest as different physical shapes or be described as direct or indirect sources. Direct light sources can be further divided into point light sources and area light sources. On the other hand, indirect light source can be described as any light emitting

surface in the environment. Although area light sources can usually be modeled as finite polygons, another level of complexity can be added by modeling a polygonal surface with a suitably defined energy distribution function instead of a uniform energy emittance distribution [68]. In effect, for any photo-realistic image each of the above characteristics has to be modeled carefully.

A typical classification has been: ray tracing based algorithms and radiosity based algorithms. Another corresponding classification suggested is: the Monte Carlo based methods and finite element based methods. Both these classifications are valid in there own right. Several hybrid schemes that combine both these approaches have also been proposed. However, light transport algorithms can be viewed in a unified framework as different numerical methods for solving the *rendering equation* defined by Kajiya in [27]. The equation computes the light transferred from one point to another along with all the light received by the first point and reflected to the second point.

Both ray tracing and radiosity based approaches can perform global illumination computation. However, ray tracing is more suitable to model specular phenomenon easily, whereas radiosity is more suitable for computation of diffuse phenomenon easily. Again, ray tracing is classified as a point-to-point propagation model, whereas radiosity is termed as an area-to-area propagation model. An algorithmic artifact between the two approaches is that ray tracing generates a view dependent image, whereas radiosity computes view independent energy distribution. At the same time, a view dependent image can easily be generated as a post process by radiosity methods, although modeling

specular reflections off a surface can still be a challenge for radiosity algorithms.

2.1.1 Ray tracing based algorithms

The basic algorithmic approach for ray tracing can be stated as follows: given a source position and a direction in a three-dimensional space, a ray[1] is constructed and its interaction with various surfaces in an environment is studied. From a geometrical point of view, if a ray intersects with a surface, it could either reflect specularly, scatter, or transmit through that surface. The new rays that get generated due to the above scattering, reflection, and transmission events depend on the incident angle of the ray and material properties of the surface. This intersection operation of a ray with the surface can be computationally complex for higher-order surfaces. For computational reasons some of these systems approximate such surfaces as piece-wise linear surfaces.

Taxonomy of ray tracing algorithms

Ray tracing as such can be initiated either from a receiver or a transmitter (in the form of a light source). This has resulted in a further classification of the ray tracing algorithms that have been pursued. There has been considerable debate in the literature over the terminology of what constitutes forward or backward energy transport mechanism. Arvo [3] initially coined the term *backward ray tracing* for algorithms that move from the light source since most of the previous algorithms had traced rays from the eye position. Chat-

[1] A ray is defined as an infinite vector in an arbitrary direction with the source position as its origin.

topadhayay [7] and Glassner [24] termed the conventional ray tracing as backward ray tracing and used the term *forward ray tracing* for the algorithms that trace rays from the light source. The latter convention is more suitable from the energy propagation perspective, whereas the former terminology takes the artifacts of an algorithmic implementation into account.

A better naming convention (at the risk of introducing yet another one) is one that is explicitly based on the nature of the algorithm, for example, a *source-based* or a *receiver-based* ray tracing algorithm. Here, the *source-based* algorithm would correspond to tracing rays from the light source and *receiver-based* algorithm would correspond to tracing rays from the eye position.

Derivatives of ray tracing

Over the years several variations have been proposed to this basic ray tracing approach. Amanatides proposed cone tracing [1] where a linear ray is generalized as a cone having a solid angle. The rays are cast into the environment from the eye position. Initially, the solid angle for each cone is large enough to encompass a single pixel. Instead of testing an intersection of a cone with the objects, a projection of the objects on a plane perpendicular to the cone is used for intersection purposes. This simplifies the intersection computation at the cost of accuracy. A new virtual origin is computed along with a new solid angle defined by the intersected area. This new cone is further propagated into the environment to account for scattering off surfaces. The intersection computation is

expensive and yet approximate.

Heckbert and Hanrahan [26] introduced the concept of beam tracing for polygonal environments starting from the eye position. A single beam encompassing the image plane is cast into the environment. The initial beam is fragmented based on the visible polygons which in turn are further spawned into reflective and refractive polyhedral pyramids. For shading and shadowing, beams were cast from a light source. The algorithm took advantage of the coherence between adjacent rays to provide a significant speedup as compared to conventional ray tracing. However, the method failed to model refraction correctly because it did not include the bending of the beam as it moved from one medium to another. An implementation artifact of the algorithm was that it used depth-first sorting to obtain the closest surface intersection instead of an algorithm that preprocesses the scene such that priority ordering can be quickly determined from any viewpoint, for example, the scheme described in [21]. This resulted in the algorithm's time complexity to be $O(n^2)$ for n polygons; in effect, not scaling well for large environments.

Shinya et al., [56] proposed yet another algorithmic variation to ray tracing, which tried to account for the shortcomings of the above two approaches. The new approach was named *Pencil tracing*, where a pencil was defined as a set of rays parallel to and surrounding an axial ray. The surrounding rays were called paraxial rays. The anticipated advantage was that since the paraxial rays share approximately the same direction, the same axial-ray linear transformation could be applied to a set of paraxial rays. This scheme was valid both for reflection and refraction. Error estimation techniques were

also developed to determine an approximate solid angle such that the transformations were valid for the paraxial rays inside the angle.

There have been various attempts at combining different ray tracing schemes such that some computation is done starting from the eye position and the rest starting from the light source. Arvo [3] proposed an algorithm to perform shading calculations starting from the light source and the rest of the computation moving from the eye position. In general, ray tracing is suitable for computing specular reflections easily with point light sources but it is not the best approach for diffuse reflections.

2.1.2 Deterministic vs. non-deterministic propagation algorithms

We will notice that all the algorithms discussed above are considered to be deterministic in approach. Here, determinism refers to the process by which an algorithm decides the specific paths that are to be traced. This process need not necessarily be deterministic as long as enough paths are traced such that all significant objects get sampled. Both deterministic and non-deterministic methods have been proposed for realistic image synthesis. Non-deterministic approaches such as Monte Carlo methods have been widely used to study global illumination effects to generate synthetic images. There are several variations to the basic Monte Carlo method for such simulations. In computer graphics, Appel [2] used a method that traced random particles to generate images. Whitted [72] proposed the idea of randomly perturbing viewing rays. Cook et al., [13] used similar ideas to model several different phenomenon including motion blur, depth of field,

penumbras, translucency, and fuzzy reflections. They termed their work *distributed ray tracing*. Variations of Monte Carlo methods have been used to address the problem of aliasing (by converting it into random noise). Recently, Veach [66] proposed a *Metropolis Monte Carlo* method for light transport where randomly generated paths are mutated to spawn additional paths in the vicinity of important paths. One of the main arguments in favor of Monte Carlo methods is that they are more general and apply to several types of environments with complex surfaces, especially ones which cannot be approximated by polygonal geometries. Veach argues that such methods are more robust than the corresponding deterministic methods [67]. They also allow modeling of diffuse reflections and complex BRDF simulations which are more difficult to model with classical deterministic ray tracing methods.

2.1.3 Radiosity based algorithms

A radiosity method (has also been referred to as a *finite element method* in literature) is the other commonly used illumination and energy propagation technique for image synthesis. This approach is more suitable for global illumination where each polygon or part of the polygon is simultaneously modeled as a receiver as well as an emitter of energy [23, 44]. The advantages of radiosity methods are two-fold. First, they provide view independent illumination results. Second, radiosity algorithms account for all energy in the environment rather than just the line of sight occlusion. On the other hand, if one needs to model only a few viewing positions then computing view independent results

may be wasteful and can be argued to be a disadvantage of the method. The basic computation of a radiosity method is to define a linear equation for each surface in terms of the energy emitted and received by the surface with respect to every other surface in the environment. For example, for an environment containing n patches, the system of linear equation can be defined as a matrix given by

$$\begin{bmatrix} 1 - R_{d1}F_{1,1} & -R_{d1}F_{2,1} & ..... & -R_{d1}F_{n,1} \\ -R_{d2}F_{1,2} & 1 - R_{d2}F_{2,2} & ..... & -R_{d2}F_{n,2} \\ . & . & ..... & . \\ . & . & ..... & . \\ -R_{dn}F_{1,n} & -R_{dn}F_{2,n} & ..... & 1 - R_{dn}F_{n,n} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ . \\ . \\ I_n \end{bmatrix} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ . \\ . \\ \varepsilon_n \end{bmatrix} \tag{2.1}$$

where I_i is the illumination received by i^{th} patch from all other patches, R_{di} is the diffuse hemispherical reflectance emitted by the i^{th} patch, F_{ij} is the factor (also known as a form factor) that relates the energy density incident on patch j from patch i to the intensity reflected from patch i, and ε_i is the emissivity of a patch. The computation proceeds by solving this set of linear equations. Though radiosity methods account for all energy, processing more than a couple of reflection iterations is computationally very expensive. Besides, to account for specular reflection special processing has to be performed. The current state-of-the-art radiosity algorithms take several hours and even days to compute a few iterations for large scenes with several hundred-thousand polygons [61].

2.1.4 Hybrid algorithms

A combination of radiosity and ray tracing algorithms have also been proposed. Wallace [70] combined the two methods to propose a two pass solution to Kajiya's *rendering equation* [27]. Shao [55], and Sillion [58] combined the two approaches to provide specular and diffuse reflections in the same scene. Wallace [71] also extended the *progressive radiosity* [11] scheme using *hemi-cube algorithm* [10], by shooting rays from all surfaces in the environment towards a secondary light source and then determining the surface-to-surface energy contribution. This resulted in addressing the insufficient sampling issue and the aliasing problem inherent to the earlier approaches.

2.1.5 Participating and non-participating media

For the purpose of realistic simulation, one also needs to differentiate between participating and non-participating medium. In most cases, it is computationally simpler to assume that there is no attenuation due to an intermediate medium, that is, there is a vacuum. For the same reason light transport algorithms typically assume non-participating medium. For realistic image synthesis researchers have taken an easier way out by modeling these phenomenon as textures. On the other hand, if a natural phenomenon such as fog, smoke, clouds, etc., which attenuate light to varying degrees has to be modeled, an algorithm can be both computationally expensive and complex. One such example of modeling smoke procedurally is given in [15]. An inexpensive solution commonly used is to attenuate light as a function of distance.

2.2 Ray tracing versus wavefront approach

Ray tracing traditionally starts a ray either from the source to trace its path until its energy attenuates below a certain threshold or has encountered a fixed number of surface interactions; or from the receiver to trace it until the ray reaches a source or travels outside the scene. This approach can also be termed as the *depth first approach*. The wavefront approach on the other hand advances all the rays to the same distance at any given state of the computation. This can also be called as the *breadth first approach*. Moravec [38] very early on had proposed simulation of light wavefronts. The idea was that instead of infinitesimally thin rays, light is represented as wavefronts which were stored as two dimensional arrays of complex numbers. The size of a model and propagation within it were assumed to be a direct function of the wavelength being used for the simulation. That is, the scene was represented as a box made up of cells of size half wavelength wide. Each cell was characterized by three complex coefficients (for refractive index, scattering, and light source). As a wavefront propagated through a scene, the wavefront coefficients were multiplied with the coefficients of the cells describing the scene. The method was considered to be suitable for parallel processing and supposedly superior for extremely complex and dense models[2], where particle form of propagation can be computationally very expensive. Propagation in the direction orthogonal to a wavefront was recognized as a special case, which was defined as a convolution. Fast Fourier Transform (FFT)

[2]The basic assumption was that scenes have matter every one half wavelength, i.e. typically every one half to one meter.

was used to speedup the processing to $O(n \log n)$ rather than $O(n^2)$ for typical particle tracing methods, where n is the number of elements in the two-dimensional array of coefficients. A two dimensional transformation was accomplished by first taking a one dimensional FFT of each row and then taking a one dimensional FFT of the result. One direct advantage of the above method is a view independent computation similar to radiosity solutions and unlike ray tracing. It also alleviated the aliasing problem common for classical ray tracing solutions.

In a more recent work, Elber [17] discusses computation of wavefronts. Light is propagated both from a point source as well as a spherical (with a well defined surface area) source in an environment consisting of freeform surfaces. A major problem for propagation in such environments is in determining a correct reflection wavefront off a freeform surface. For this, a trivariate wavefront equation is solved to compute the trivariate irradiance function. The trivariate wavefront is further approximated as a series of bivariate wavefronts for ease of computation. The principal idea used in the irradiance[3] computation is the *intensity law* of light where irradiance is inversely proportional to the area under consideration.

Combination of depth first and breadth first approach

To schedule the casting of rays, yet another variation of the above depth first and breadth first techniques is a combination of the two approaches. This has been proposed by Pharr et al. With faster computers, rendering algorithms can handle much larger and complex

[3]Defined as radiant flux received per unit area.

scene databases. But this increase in scene complexity reaches a limit if the scene database is so large that it does not fit into the memory all at once. This results in thrashing, thus offsetting the gain achieved through a high speed processor. *Memory-coherent* ray tracing [46] alleviates this by proposing a caching [45] and reordering scheme. The method caches a subset of the geometric and texture databases in the main memory for fast access by the rendering system. First the algorithm statically reorders the scene data depending on the viewing position. Then it dynamically loads these subsets into memory. It also dynamically reorders ray intersection processing to suit the subset in the memory.

2.3 Propagation algorithms for wireless communication systems

Wireless computing and communication is a powerful paradigm that transcends physical connection between a source and a receiver. This has resulted in a host of applications being suggested using wireless transmission. These applications range from voice transmission in the form of cellular phones to data transmission for mobile radio pagers, palmtops, and laptops.

The advances in wireless technology and its applications has precipitated a renewed interest in the computational models that can simulate radio wave propagation. For wireless applications we are mainly concerned with radio frequency propagation of wavelength from a few millimeters to less than a meter. Though both radar and wireless applications operate over the same range of radio frequencies, it is important to understand the

differences in objectives for propagation in each domain. For radar applications, target detection is the main objective. However, for wireless applications one is interested in understanding the exact distribution of energy from a specific source. One of the important applications of such understanding is optimal placement of transmitters in various environments, such that the power does not drop below noise threshold and there is a smooth handoff of a mobile user from one cell to the other. This is necessary for an uninterrupted wireless link for the mobile user. The criterion for an optimal placement is the overall distribution of power in an environment. In particular, power received at a specific receiver position in three-dimensional space assumes considerable importance. One of the major upcoming applications for wireless communication is macrocell, microcell, and picocell placement in suburban, urban, and indoor environments respectively.

2.3.1 Propagation models for simulation of transmitter placement in an environment

A *'good'* placement of transmitters is an iterative procedure which can be very costly if physically performed for every site. Typically, cellular service providers conduct elaborate field trials at selected sites that broadly represent a class of environments such as cluttered urban environments, rural environments, or inside buildings. They physically place a transmitter at a heuristically determined location and then physically measure the exact received signal strength at predetermined locations in three-dimensional space.

A better method would be if an exact geometric model of a site-specific environment

could be created in which one could simulate propagation of energy from various transmitter locations, and predict signal strength in three-dimensional space with reasonable accuracy. Further, there may be more than one transmitter in the environment and it may be worth computing if there would be any co-channel or adjacent-channel interference between various transmitters. Thus, a simulation which quickly computes radio frequency propagation in a geometrically represented model of an environment is highly desirable.

The two important problems that need to be addressed for such a system are an efficient representation of the geometric environment and a computational scheme to propagate energy. We will discuss various schemes of geometric representation and their advantages in Chapter 3. For energy propagation, there have been various attempts at ray tracing based solutions to the above problem for predicting signal strengths in indoor and outdoor environments.

For indoor environments, Valenzuela [64] suggested a ray tracing approach where exact reflection and transmission paths are constructed between a transmitter and the receiver stations. The scheme computes the number of intersections between the source and the receiver and attenuates the signal as a multiple of these intersections but for each reflection it creates a new virtual location. For computational efficiency the number of reflection paths are restricted. The algorithm is strongly biased in favor of transmission paths and this computational artifact could result in skewed distribution of power in an environment where the amount of reflected energy is more than the transmitted energy

(because of the material properties of objects in that environment). Schaubach et al., [52] and Seidel et al., [53] have suggested a pure ray tracing based approach for predicting path loss and delay spread in microcellular environments, where the received signal strength is sampled at large distances to computationally speedup the predictions. Sparse sampling may be sufficient for rural flatlands where most of the power attenuation is due to distance between a receiver and a transmitter also known as *long-term* or *large-scale fading*. However, it is not sufficient for heavily built urban environments, where power distribution could be very different just a few tens of meters apart.

Kreuzgruber et al., [32, 33] suggested a ray splitting method which maintains a constant spatial resolution. A ray is cast as a ray tube. Splitting of a ray tube takes place at predetermined zonal boundaries. The scheme is able to address the problem of missing geometry. However, there is an exponential increase in the number of rays cast into the environment at every zonal boundary. For large environments this approach can be prohibitively expensive. The proposed scheme also does not take into consideration the increase in the projected surface area with an increase in the incident angle. This can introduce arbitrarily large error due to insufficient subdivision of a ray tube for large incident angles.

There has also been an attempt at computing an optimal solution for a placement of a transmitter in an environment. Fortune et al. [20] proposed an adaptation of the Nelder-Mead's [43] direct search method to the transmitter placement problem. The idea was use a combination of some well defined heuristics and the optimization algorithm.

Despite the existence of several local optima the authors observed that the above problem was not suitable for global optimization.

Diffraction is an important component of the total energy in outdoor environments. For prediction of path loss in rural environments Luebbers in [36] and [37] has compared the *knife edge diffraction* model with the *wedge diffraction* model. The main difference between the two is that for knife edge diffraction, the *'knife'* is considered as a perfect absorber and hence the source is treated as being in the plane of the *'knife edge'*. On the other hand, wedge diffraction is similar to GTD where diffraction takes place in any of direct, reflected, or diffracted regions. We describe geometric aspects of GTD, the classification of regions around a wedge, and its coefficients in detail in Section 5.1.1. Thus, GTD turns out to be a more powerful mechanism for diffraction with little additional cost.

To achieve computational optimization for diffraction propagation, a variation has been proposed where edges are treated as secondary energy sources. In the first pass of computation all energy reaching an edge is collected. A second pass is used to propagate that energy further into the environment. This scheme has a drawback. When a ray is incident on an edge it divides the area (in case of two-dimensions), or volume (in case of three-dimensions), around an edge into three distinct regions—the shadow region, the reflected region, and the incident ray region. These regions are a function of the ray's incident angle. The energy attenuates in these regions based on a distribution function over the entire range of 360°. If all energy incident on the edge is collected from all sources

before it is propagated further, we are inherently losing information on the boundaries for these regions which results in significant approximation. For outdoor environment, Vinco et al., [69] have suggested an analytical solution to the diffraction problem around the corners. Their solution applies to building corners in a typical rectilinear street environment besides being valid only for two-dimensional geometries.

As mentioned earlier, a popular technique used by cellular service providers to study propagation effects for wireless applications is to conduct field trials and study these effects in different types of environments. The idea was that if a fairly large number of field trials are conducted in different types of environments and broadly classified into a few types such as cluttered, urban, semi-urban, rural, indoor areas then certain empirical conclusions could be drawn about there attenuation characteristics. These could in turn be used to draw conclusions for similar type of environments. In fact, such field trials have concluded that long-term fading in urban environments follow a *log-normal distribution* whereas short-term fading with reflected energy follows *Rayleigh distribution*. Similarly, direct plus reflected paths follow *Rician distribution* [34]. Often these results have been used in statistical simulations to predict path-loss for broadly classified environments. But such classification yields very approximate results and are good only as a first level of approximation.

Classical image based ray tracing is not efficient for the transmitter placement problem because of the number of receiver locations is very large. At the same time, as is evident from the above survey most of the work on radio propagation algorithms has nar-

rowly focussed on specific types of environment such as inside buildings or in urban areas for street level propagation. This becomes a restriction in scenarios where propagation distribution has to be studied from one type of environment into another. Another problem has been that these algorithms do not scale well to large environments. We propose a unified approach that is equally efficient in all types of large geometric environments and their combinations.

2.4 Propagation algorithms for acoustic wave modeling

Acoustic modeling in geometrically defined environments has to address issues similar to radio propagation modeling such as efficient method of propagation as well as efficient geometric representation. At the same time there are some application dependent differences which impact the actual algorithms applied to solve the individual problems. One major application dependent difference is in the type of antenna being used. If an acoustic modeling system assumes isotropic omni-directional source then the beams can initially be modeled by the largest polygons surrounding the source (for example the six polygonal faces in a room), thus making beam tracing more suitable. However, for wireless applications antennas typically have very complex directional, non-isotropic beam patterns (with rather fine resolution). Modeling the tilt of the beam, both in azimuth and elevation is another important RF engineering consideration. In fact, besides focussed applications such as directional propagation, beam tilting is being seriously considered for network traffic hot-spot mitigation and other network-wide dynamic load-balancing

schemes. Another primary difference between modeling radio frequency propagation and acoustic modeling for auralization in interactive virtual environments is that in former scenario a user is typically at a constant height above ground (or floor in case of inside buildings) whereas a user in the latter scenario can move anywhere in three-dimensional space. For this reason, in the latter case a whole beam-tree data structure has to be stored as opposed to a single height field where we just store possible user positions in the environment. For wireless applications, the user is typically at a constant height from the ground (for example, six feet above ground for hand held cellular phones, which is also a typical height of a car antenna).

In a recent work, Funkhouser et al. [22] used beam tracing for acoustic modeling in simple three-dimensional environments (in terms of environmental detail). The system implemented a full beam tracing algorithm at the cost of not being able to handle large detailed environments. It initially partitions the three-dimensional environment into cells using a binary space partitioning tree scheme and constructs an adjacency graph between adjacent cells. Then it cast beams from an acoustic source and traces them for specular reflection and transmission. These paths are further stored in a beam tree data structure for later interactive visualization and audio playback as virtual characters move within the environment. The work does not scale well for large geometric models with a high degree of local geometric detail because of an exponential growth in beam fragmentation and its associated data structure. For example, if the beam interacts with many cell boundaries then it can get trimmed into narrow beams diminishing the advantages of

beam tracing over simple ray tracing. It is important to note that the performance of any system is directly proportional to the geometric detail available. But since the above system exploits the coherence in an environment by using beam tracing, efficiency of the system is adversely impacted by a detailed model. As we will describe later, a hybrid scheme of representing beams by their medial rays is more desirable to model acoustic waves too.

2.5 Propagation animation

A form of the breadth first approach with which we experimented in the course of the development of the radio propagation algorithms is to advance a ray until it interacts with a surface and then follow the same procedure with the next ray in the environment and so on. This procedure is followed all the way until rays in all directions have been propagated, constituting a single pass. The previous rays are stored in a tree-like data structure for further processing. In this approach, rather than propagating a ray further into reflected and transmitted rays and following that particular ray until it attenuates below a threshold, one computes the path of other rays to similar surface interaction and then returns back to the computing of previous rays. The above form of wavefront propagation has advantages where the system runs in an interactive mode. With each pass, the movement of a wavefront is studied as it propagates in an environment, in a way facilitating *"propagation animation"*. This is in contrast to viewing the energy distribution map in the environment as a final image.

A highly desirable goal for these computational algorithms is to be able to animate propagation of energy in various environments. For this purpose computational speed of an algorithm[4] assumes importance, which leaves a lot to be achieved, especially if accuracy of the images that are generated also has to be maintained.

[4]It is interesting to note that Reif et al., have formally studied the complexity of ray tracing [50] and beam tracing [49] in a three dimensional optical system consisting of finite set of reflective and partially reflective (transmittable) objects. The ray/beam tracing problem was formulated as a decision problem where given an optical system, a light ray's initial position, the direction, and a fixed point *'p'* in three-dimensional space, the question that the ray eventually reaches *'p'* was proven to be undecidable. Thus, though a general solution to the ray tracing problem is computationally unsolvable, its intelligent adaptation to various applications could yield useful results. Also, in the presence of attenuation, and a bound on detectability, one can make this problem decidable.

Chapter 3

Efficient Geometric Representation of Environments

Any simulation of energy propagation using geometric optics and its extension in the form of theory of diffraction, assumes a scheme of geometric representation for a given environment. Representation of a site-specific geometric environment itself is a two step process. First, we need an efficient scheme to represent the database of geometric entities (in the form of a suitable data structure) to quickly answer some key questions. These questions could range from determining the extents of a bounding box for a geometric entity or a group of entities; to the question of whether a ray intersects with any object and if so, the specific surface of the object that is intersected along with its material and other properties. Fast answers to these questions become central to the efficiency of any algorithm that is implemented using such a data structure. The efficiency of a represen-

tation assumes further importance if one of the objectives is also to interactively visualize the environment. Second, we need a mechanism to compile a geometric database for a specific site, which may be available from several different sources. This raw information may also vary in terms of the format, type of information, and finally the scale (or detail) of information. For example, it may include elevation information of a geographic area along with its vegetation cover; a blue print of a city with a height field for each building, or a street layout with an associated approximate height field for each city block; or an architectural drawing of a specific building with detailed floor layout of each floor along with geometric details of each piece of furniture. A manual compilation of such databases can be so tedious and time consuming that any realistically large environment could take unrealistic amounts of time. Thus many ideas have been proposed towards generating such geometric databases in an automated fashion. Some such databases are primarily geared towards interactive visualization and others are specifically geared towards simulation of local or global illumination.

In the following sections we will first discuss some of the popular geometric representation schemes and why they do not serve our purpose of simulating radio propagation. Then we will discuss the rationale for using binary space partitioning trees to build a radio propagation prediction tool. Finally we will discuss the methods of acquiring different types of databases for our prediction system.

3.1 Popular representation schemes

For rendering geometric scenes, the computer graphics community has proposed several different representation schemes. Some of the schemes studied extensively are boundary representation, constructive solid geometry, different spatial subdivision methods, and freeform surfaces represented as high degree polynomials. Besides these basic methods there are several variations to these that have been proposed. The usage of a particular scheme is heavily influenced by whether the representation is used for rendering, computer aided design and analysis (such as solid modeling), computer aided drafting, or for image processing.

3.1.1 Boundary representation

Boundary representation (*Brep*) is typically understood as any representation of objects by surfaces (as opposed to solids). It has been the most versatile and widely used scheme to represent geometry for various application domains. In its simplest form the scheme enumerates all vertices in an environment. Edges form a topological relationship between vertices. Similarly, in three-dimensional space, faces form a topological relationship amongst edges. For *brep*, a combination of these vertices, edges, and polygonal faces (planar) is defined as an object.

A clear advantage of the representation is the simplicity in defining its data structure. However, a major disadvantage of *brep* is that the basic definition does not distinguish between the *inside* and *outside* of an object. Usually additional information has to be

stored to make this distinction explicit. Typically, a surface normal is stored for each polygon, with a convention that the normal is defined as a vector facing away from the outside of a polygon. This enables a distinction between the inside and the outside of a polygonal face with respect to an object. However, one still cannot ensure the consistency of an object in terms of an enclosed space, since there is no mechanism to know if there are cracks between edges or vertices. An accompanying data structure is needed to help find adjacent vertices in an environment. Similarly, determining the closest surface intersection is a costly operation unless a spatial relationship amongst all faces is maintained. This again adds to the complexity of the data structure. Several variations to this basic scheme have been proposed by maintaining relationships amongst different entities in the data structure. Most of the systems for radio propagation have used some variation of boundary representation. Fortune et al. used a brep data structure based on triangulations [20].

3.1.2 Constructive solid geometry

Constructive solid geometry (CSG) is a popular scheme of representation for solids as opposed to a scheme for boundary representation. The scheme was primarily developed to model complex three-dimensional objects. The representation starts with a set of simple primitives such as a cube, sphere, cone, pyramid, and a tetrahedron, etc. Primitives are defined as bounded volumes, typically a result of an intersection of a set of halfspaces. For some primitives such as a sphere or a cone, the bounded volume may be a polygonal

approximation of the true shape with a tight control on its volumetric error bounds. New objects are constructed from these primitives by using one or more of the boolean operations *(union, intersection, and difference).* These operations could be between any two primitives, or between a primitive and a complex object. Affine transformations (such as scaling, translation, and rotation, etc.) can also be performed both on primitives and their combinations. These boolean operations form a hierarchy of intermediate objects, which can be conceptually stored in a hierarchical tree structure. The nodes of this tree represent boolean operations between objects, whereas leaves of the tree represent the primitives. The whole environment consisting of several complex objects may again be represented as a single tree for rendering computations. A primitive can also be defined by its exact polynomial representation. The advantage is their compact representation. But at the same time, after several boolean operations as the complexity of an object increases, it may become difficult to represent the resultant object as a simple polynomial.

There are several other schemes where instancing in some form is used as a means of constructing large databases consisting of replicated object or slightly transformed objects.

3.1.3 Spatial subdivision

Spatial subdivision methods have been used for many modeling and design applications. The basic idea of any spatial subdivision method is to divide space into disjoint subspaces. Space in the current context is being used as a dimensionless quantity, that

is, it can represent two dimensions, three dimensions, or any higher dimension. For practical purposes here we will restrict ourselves to two and three-dimensional spaces. The resolution to which space is subdivided depends on the desired accuracy for a particular application or a specific object. Space can be subdivided by a number of techniques. It can either be subdivided uniformly or nonuniformly. At the same time subdivision can be along the coordinate axes or along any arbitrary axis. An advantage of uniform subdivision is that it is algorithmically simple but it could result in unnecessary division of space (depending on the application). Nonuniform subdivision on the other hand usually subdivides space based on some predefined criterion. The criterion could be based on some logical entity or for that matter could be based on a physical entity. Nonuniform subdivision in general is algorithmically complex but more precise. The higher computational cost may not be a very important consideration since for most applications it is performed as a pre-processing step.

Two of the most commonly used spatial subdivision schemes are Binary Space Partitioning Trees (bspt) and n^d trees. Theoretically speaking the process of subdivision for both these schemes can be applied to any dimensional space. For the same reason they are sometimes referred to as *dimension independent* spatial subdivision schemes. Binary space partitioning trees have also been referred to as *partitioning trees* in literature. The main difference between the two is that n^d trees perform axis aligned subdivision in a d-dimensional space. Here n refers to the number of divisions performed for each dimension. For example, binary partitioning divides the space into two sub-spaces. Each subdivision

operation creates equally sized n^d spatial units. Octree can be considered as a special case of n^d trees for three-dimensional space where every subdivision results in 2^3 (i.e. 8) equal octants along its three axes. In fact, it acquired its name *octree*—a tree of octants, because of this operation. Quadtrees can again be treated as another special case of n^d trees for two-dimensions, which subdivides two-dimensional space into four quadrants. On the other hand, bspt partitions a d-dimensional space with a $(d-1)$ dimensional entity [21, 41]. For example, planes (which are two-dimensional) are used to partition a three-dimensional space, and so on. Here unlike octrees, the partitioning hyperplane[1] does not have a preference towards a specific orientation. Both these schemes define a hierarchy of recursive subdivision.

3.1.4 Other hierarchical schemes

There are some other hierarchical decomposition schemes, which have used levels of enclosing volumes though not necessarily disjoint [9, 14]. The basic idea of both the schemes was to have a hierarchy of bounding volumes, which enabled pruning of large parts of the geometric environment to avoid any unnecessary intersection computation. These bounding volumes were not necessarily of a particular shape or size. Another hierarchical scheme that used only parallelepipeds was proposed by Rubin et al. [51]. The main motivation was that this homogeneity allowed simpler computational processing. These

[1]In literature, hyperplanes have been used to refer to different concepts in different contexts. Here hyperplane is used to denote the dimension independent nature of the dividing plane in contrast to the two-dimensional plane.

parallelepipeds were also oriented such that they minimized the bounding volume for the enclosed objects.

Another hierarchical data structure used for extremely large geometric databases was proposed by Pharr et al. in [46]. The work addressed two problems. First, the problem of memory thrashing because the geometric database is so large that it does not fit into the main memory all at once. Second, to improve the efficiency of the computation for nearest ray intersection. As we will see there is room for improvement on both these fronts. The efficiency comes from the observation that any ray tracing method on such databases can be improved by adhering to the spatial locality in a geometric database. The solution proposed was that the environment be divided into smaller subsets that have roughly the size of the available cache. For efficiency each of these subsets could then be loaded or removed from memory all at once. To avoid loading a subset of geometric entities many times the environment is overlayed with a grid of voxels called the *scheduling grid*. Rays are grouped together for each such voxel and processed only after sufficiently large number of rays have been collected. Additional efficiency is achieved by using a single shape such as triangles for all geometric objects. However, this one-level hierarchy of subsets still does not suffice for fast intersection computation because a large database may end up having a few thousand triangles per grid voxel. A two-level intersection hierarchy consisting of the *geometry grid* and the *acceleration grid* were used in the above work. It is important to note that although they used a two-level hierarchy, the number of levels could easily be extended to three or more depending on the number of geometric

entities in the environment. The basic idea is to bring the number of geometric objects down to a few hundred at the lowest level within any grid voxel. One primary drawback with the above regular grid approach is an implicit assumption that geometric objects are uniformly distributed in an environment. Typically, for non-uniformly distributed environments an octree-like grid structure [59] may be more suitable. Another approach applicable to non-uniformly distributed environments and more general as well (in most cases), may be a binary space partitioning tree, as discussed in Section 3.2.1.

3.1.5 Freeform complex surfaces

As one would notice, all the above representation schemes are either polygonal approximations of freeform objects (e.g., spatial subdivision methods) or in fact, are pure polygonal representations (e.g., brep). CSG in principle can have any primitives with a polynomial definition (e.g., sphere, cone etc.), but for computational reasons polygonal approximations are used frequently. No discussion on geometric representation schemes is complete without mentioning freeform surfaces. This leaves us with a need to represent freeform surfaces (for example, the surface of the earth and other naturally occurring objects). Some can be represented as a function of a polynomial equation. However, there are a host of other representation schemes for this purpose, such as algebraic surfaces, and B-spline surfaces including their variations and parametric forms. Though the representation is exact in these forms it is computationally expensive to perform intersection operations. This computational cost may not justify the advantage of an exact represen-

tation and hence usage of such a representation is purely dependent on the application for which the representation is being used. To achieve the overall goal of a unified approach to model radio propagation as well as for fast intersection computation, we will represent a terrain as a binary space partitioning tree.

3.2 Unified approach to radio propagation in any geometric environment

As was discussed earlier, most of the earlier work to compute radio power distribution has focussed on specialized types of environments. Interestingly, these environments were classified not based on their electromagnetic wave propagation properties but rather on the ability of the algorithms to represent large environments. Needless to say, that specific systems have been developed that can simulate energy in only one type of environment at a time such as urban outdoor environments or inside buildings. At the same time, there are no known systems that simulate all radio propagation effects (for example, line-of-sight, reflections, and diffractions) in environments which have large differences in elevation (valleys and hills). We classify these last type of environments as terrain based environments. The problem is not so much in managing different systems for different geometric environments, but in simulating radio propagation from one type of environment to another type of environment. For example, a transmitter placed at a street corner in a dense urban city such as Manhattan does not restrict its signal propagation merely to the streets, the signal also creeps inside individual buildings. With

more and more buildings having their own cellular and PCS networks, this energy from outside the building can cause large interferences inside. On the other hand, if this signal strength is strong enough and is of the desired radiation (is from the same system, i.e. non-interfering) it can actually influence the final placement of transmitters inside these buildings. For this reason, a unified approach to propagation in a combination of environments goes a long way in achieving this goal.

There are two areas that have to be addressed for a unified approach to radio propagation in any geometric environment. First, is a multi-resolution efficient data structure to represent geometry. Second, is an adaptive scheme to cast rays at a finer resolution in places where there is more geometric detail and at a coarser resolution where either the geometric detail is not high or the objects are sparsely located.

3.2.1 Binary space partitioning trees

Partitioning trees have been a popular method to represent large geometric databases for rendering images. For example, for global illumination computations, for interactive walkthroughs of complex environments [54], and for geometric modeling using set operations [62]. Recently it has been used for acoustic modeling [22]. For radio propagation, we first used *binary partitioning trees* in [18] to achieve computational speed for intersection purposes. The hierarchical multi-resolution nature of partitioning trees was efficient to determine the closest surface intersection in large environments. The scheme also proved to be very effective for interactive visualization and placement of transmitters in large

environments [47].

The representation classifies the geometric objects in an environment by subdividing space. The process starts by choosing a dividing plane (also known in literature as a hyperplane). All the geometric objects are spatially classified depending on whether they are to the left or the right of the this subdividing hyperplane. Entities that fall on the plane are classified as such. There are several criterion that can be used to choose these hyperplanes. Some of these dividing hyperplanes can be coincident to the polygonal faces of the objects.

At the end of the process the geometric environment has a tree data structure where the nodes of a tree are the partitioning hyperplanes with the left subtree containing all the geometric entities to the left of the hyperplane and the right subtree containing the entities that are to the right of the hyperplane. Since bspt classifies the space volumetrically, the leaves of the tree classify the environment as space *inside* an object and space *outside* all the objects (or not inside any of the objects). Partitioning tree representation has a clear advantage in classifying a point in space which at worst is equivalent to traversing the depth of the tree. The depth for a "well constructed" partitioning tree is typically $O(\log N)$, where N is the number of surfaces in the environment [42]. This is a major improvement over Brep which is typically $O(N)$. Some of the other advantages are easy surface visibility determination (also known as *hidden surface problem*), and fast determination of closest surface intersection with a ray in a geometric database.

Here it will be useful to do a one-on-one comparison between the two-level hierar-

chical approach as discussed in Section 3.1.4 and a pure binary space partitioning tree approach. For example, let an environment consist of ten million triangles that are uniformly distributed across the environment. With the above approach of two-level grids, each grid-cell at the lowest level will have a few hundred triangles. A $\log(N)$ search within any grid-cell will take only a few steps, where N is the number of triangular polygons. On the other hand, an approach that constructs a single balanced bspt could take several tens of steps (on an average, $\log(N)$ steps), if each search is performed starting from the root of the tree.

Now, if we assume that many operations have to be performed, in succession, in a local region spanning a single grid-cell then speedup for the two-level approach is substantial, especially once a grid-cell is determined. This is because first the cost of determining a particular grid-cell is amortized over all the operations for the grid-cell. Second, this cost of searching for the grid-cell is not repeated for each operation performed within the grid-cell. In this example, with the assumptions we have made, the former approach has a clear advantage. But at the same time unless the application is tailor-made such that lots of operations have to be performed locally in succession there may be a substantial overhead involved in collecting operations for a local region, in which case balanced trees may still have some advantages as opposed to the grid-based approach. Shade et al. [54] have used bspt representation with balanced trees in terms of the number of geometric primitives contained in each subtree to facilitate hierarchical view-frustum culling. However, neither a uniform-grid approach nor a balanced bspt approach may

work for a realistic environment where entities are typically not uniformly distributed.

The central issue in the usage of a bspt approach for non-uniformly distributed geometry is the construction of good partitioning trees. That is, the depth of a tree should be correlated to the probability of reaching a region. This is analogous to a *Huffman coding* technique where higher probability values are assigned shorter codes. Thus, the concept of variable height trees can result in substantial computational gains depending on the distribution of geometric entities in the environment. In fact, further computational efficiency can be obtained for regions with very high density of geometric entities by using the concept of bounding volumes around these high density regions. This can also be easily incorporated in the construction of a bspt tree by storing additional information regarding a bounding box of the subtree at each node. This leads us to the question of how do we go about constructing such a tree for non-uniformly distributed geometry? Intuitively, we want short paths to large regions and long paths to small regions. We discuss this in the next section.

Construction of good partitioning trees

From the outset it seems that the efficiency of a bspt scheme primarily depends on the partitioning tree that is built for an environment. It is worth noticing that in principle, the same environment can be represented by several different binary trees and proving that a constructed tree is optimal is difficult for a any environment let alone for a class of geometric environments. A worst case scenario would be a partitioning that results in

a linear tree. The average search time for such a tree is $O(N)$, where N is the number of surfaces in the environment. At the other extreme, and the most intuitive, is a balanced tree which minimizes the overall height of the tree to $O(\log N)$. The average search time for such a tree is correspondingly, $O(\log N)$.

To construct a partitioning tree the simplest heuristic is to use the set of brep faces as candidate hyperplanes[2]. Note that multiple coincident faces are represented by a single hyperplane. Initially all brep faces are in this set. A hyperplane which covers the maximum surface area and results in least splitting of other faces is considered the best choice for subsequent intersection queries. However, better trees can be built using an expected cost model as discussed by Naylor in [42]. A combination of heuristic methods and an expected cost model for goodness of a tree can be used to generate a final partitioning tree. Since it is difficult to determine an optimal partitioning hyperplane at each step without evaluating all possible hyperplanes, various heuristic methods help in generating a set of candidate hyperplanes. Then, using the expected cost model of goodness, one of the candidate hyperplanes is selected that minimizes expected cost during each subdivision iteration.

The expected cost model for an operation on a partitioning tree is given by the summation of costs at each node of the tree. Cost at each node is given in terms of the cost of its left child tree multiplied by that branch probability plus the cost of right child tree multiplied by its branch probability plus some constant cost for processing that

[2]A hyperplane is defined as an infinite plane having a direction which is normal to the plane it represents and a scalar 'd' which is the distance from the origin of the coordinate system.

node. The left and the right child trees also correspond to the negative (r^-) and positive (r^+) subregions formed by the hyperplane at the node. The following equation states the above relationship for a general operation on the tree.

$$E_{cost}[T] = 1 + p^- \times E_{cost}[T^-] + p^+ \times E_{cost}[T^+] \tag{3.1}$$

For an operation such as finding the surface , the basic idea is that the probability of hitting a subregion is directly proportional to the area of the subregion with respect to the original area, which is given by

$$p^+ = area(r^+)/area(r) \tag{3.2}$$

and

$$p^- = area(r^-)/area(r) \tag{3.3}$$

Since we are building the tree top-down, we must estimate the expected cost of the as yet unconstructed subtree. We use the number of faces in each halfspace as given below:

$$E_{cost}[T^+] \approx |b^+|^{.8} \tag{3.4}$$

Similarly,

$$E_{cost}[T^-] \approx |b^-|^{.8} \tag{3.5}$$

We performed several experiments with the exponent by computing the average height, maximum depth, and standard deviation from the average for a constructed bspt for a given environment. A value of 0.8 was determined empirically to be the best for large

outdoor environments and individual buildings. This exponent could very well be different for different types of environments from flat terrains to very hilly terrains.

The following pseudo code of the partitioning tree algorithm gives an idea of the brep to bspt conversion.

Brep_to_Bspt: Brep b → Bspt T {

If b == **NULL then**

T = a cell

Else

h = Choose_Hyperplane(b)

{ b^+, b^-, b^0 } = Partition_Brep(b, h)

T.faces = b^0

T.pos_subtree = Brep_to_Bspt(b^+)

T.neg_subtree = Brep_to_Bspt(b^-)

}

3.3 Importing geometric databases

An essential step in performing simulation on any site-specific environment is to be able to import these databases into the simulation system in the desired format. This is true, be it a large building with detailed layout, an actual cityscape with exact spatial description of streets and buildings, or a rolling terrain consisting of hills and valleys. This can be a

challenging task for any simulation system.

Terrain geometric databases have been successfully generated by stereo satellite imagery. Geometric databases for urban area with large constructed regions can be created by a combination of city layout blueprints and airplane stereo photographs which provide the elevation information for various buildings and landmarks. Some of the techniques used to generate terrain elevation databases can be used to generate building height information at regular intervals, but certain post-processing has to be performed to reconcile this data with the actual buildings in the environment. Inside specific buildings, architectural blueprints are an important source of such databases. Though all the above methods are useful, they still require some amount of manual work to produce a usable form of a geometric database. We obtained different databases from several different sources for use in simulations outlined in Chapter 7.

3.3.1 Indoor environments

One of the methods we employed to obtain geometric databases of a layout of large buildings was through their architectural drawings. In Appendix A, we discuss at length a process we used to import building layouts and convert them into three-dimensional geometric models for visualization and simulation of radio propagation. An example of this implementation and usage of the process on a large office building (a Lucent Technologies building in Whippany, NJ) is shown in Figure 3.1.

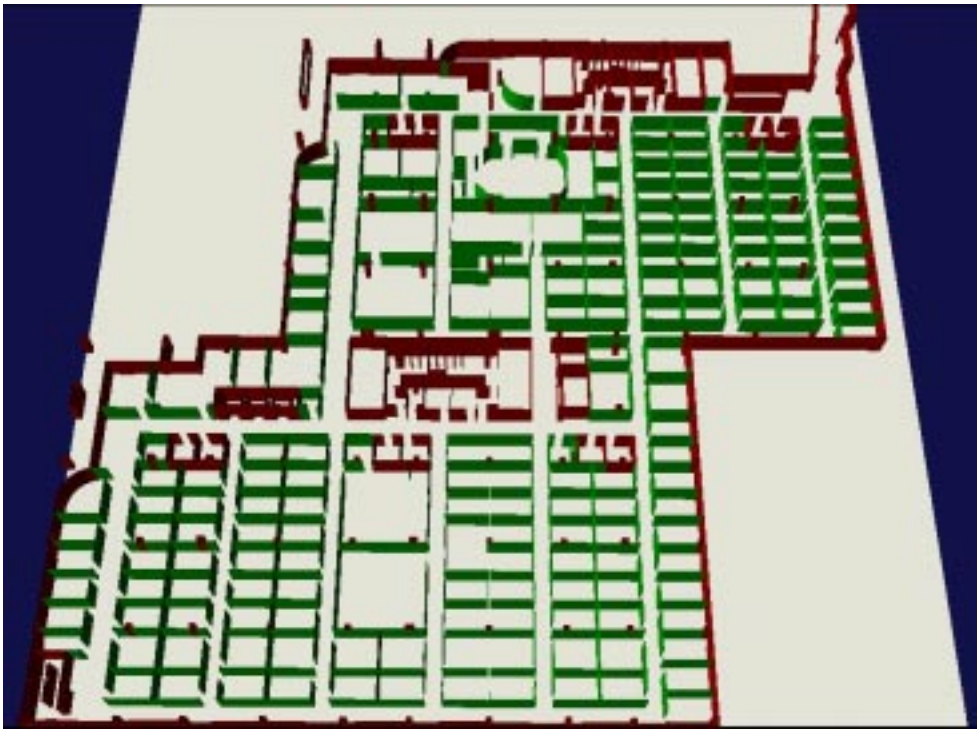

Figure 3.1: An example of a large modern office building extracted from a two-dimensional drawing and represented as a three-dimensional building with most geometric entities reconciled with their actual geometry. Different colors represent different material types. For example, in the image red represents concrete and green represents sheetrock material.

Solid vs. non-solid geometric environment

The most common problem with two-dimensional CAD geometric databases (floor plans) is that walls and other such features are represented as line segments with no associated thickness. However, since walls have finite thickness they are represented in most drawings as two parallel lines with a finite distance between them. To construct a three-dimensional representation of a floor plan an easy technique is to extrude all entities in the height direction. But that results in two parallel planes for each side of a wall with no thickness associated with them, effectively increasing the size of the database at least by a factor of two. This increase may have a direct impact on the processing time of the prediction system. Another problem is that these parallel lines may not be consistent throughout the drawing in the original database itself. For example, at wall intersections sometimes one of the lines is shorter than the other and at other times simply has a gap between the two lines at a corner. There is no easy solution to this problem.

One solution as proposed by Kernighan et al. in [29] is to *coalesce* these two lines. The problem with such an approach is that if either one of the original line segments is used to represent the resultant wall then the physical location of the wall shifts with respect to other entities in the environment. On the other hand, if a new line segment is introduced in-between these two lines instead of the two original line segments then it becomes extremely difficult to reconcile different corners of the walls without substantial manual intervention. If now a thickness is assigned to these planes, then again, reconciling the edges such that there are no gaps at the corners is a problem. This can have a significant

impact on the prediction results in environments with thick walls. This is especially true for the exterior walls of a building. Such positional anomalies at the exterior walls can also have significant impact on the prediction results if energy distribution has to be predicted inside a building from a transmitter located outside the building or vice versa. The ideal case will be to somehow fill the space in-between the two adjacent boundaries of a wall to form a single solid wall of the desired thickness.

To address the above problems and maintain positional consistency of the database we represent such environments as non-solids. Another motivation for developing this representation was in the spirit of developing a unified approach to propagation for all types of environments with inputs from several different sources. The walls are represented by their boundaries as two extruded solid polyhedrals having arbitrarily small thickness. In our context, a solid geometric entity[3] is defined as one that is a closed polyhedral[4] having a finite volume. An environment can then be consistently represented as a binary space partitioning tree. To perform ray-beam tracing as discussed in Chapter 4, we compute the first two intersections (instead of just the first intersection), where the first intersection corresponds to the front side of the wall and the second intersection corresponds to the rear side of the wall with respect to the point of intersection. Conceptually, the processing is the same as shown in Figure 5.5.

The main difference between solid and non-solid geometric representation with respect

[3]To construct a true three-dimensional solid environment from this extruded environment, one could run algorithms similar to the reconstruction algorithm given in [39].

[4]In general, the solid geometric entity need not be polyhedral in shape.

to the simulation system for radio propagation may come from the computational time it takes for each database. We conducted several experiments on the same environment represented as solid and non-solid. A comparison of the normalized run times is shown in Table 3.1.

Type of data base	Normalized time
Thin solid walls	1.0
Thick solid walls	1.0
Non-solid walls	1.23

Table 3.1: Run time comparison between solid and non-solid geometric environments.

Type of data base	Mean (in dB)	Std. Deviation (in dB)
Thin solid vs. non-solid walls	0.67	1.20
Thick solid vs. non-solid walls	1.67	2.73
Thin solid vs. thick solid walls	1.04	2.45

Table 3.2: Comparison between the prediction results for a solid and non-solid geometric environment.

As we notice from Table 3.1 it takes approximately twenty-three percent longer to compute the predictions in a non-solid environment as compared to a solid environment. The reason for the increase in time is that for the non-solid environment two intersections are being computed for the corresponding intersection in case of a solid environment. This is expected based on the bspt represented geometry which finds the closest surface intersection as an $O(\log n)$ operation. The normalized run time reported in the table includes the overhead of reading the geometric database, computing coefficient tables for

each material type and wall characteristic, and writing the final *dB* images. Table 3.2 shows the difference in predicted values taken over all the receiver locations between the two types of representations.

3.3.2 Outdoor environments

There are two methods of importing databases of large cities. One uses building databases that contain height information for individual buildings. These databases are simple to process and can be very accurate. The other method which may not be as accurate as the first approach is to build a three-dimensional database from the two-dimensional street maps. This latter approach has two drawbacks. First, the accuracy of such a database will only be approximating the city to a block level with possibly the mean height of all the buildings in a block assigned to the respective blocks. Second, these databases can only be constructed for dense cluttered city environments such as Manhattan, which has large builtup area and well defined blocks (in a grid format). The obvious advantage of the latter approach is that detailed street maps are quite ubiquitously available whereas for the former approach these databases have to exist with some agency or the other. Notice that we are not giving a lot of consideration to details like a facade of an individual building. These are important geometric details from a visual perspective but are not as important from a perspective of computing long-term fading characteristics in an outdoor environment. Besides, incorporating these details in the database are a matter of debate on the accuracy versus computational efficiency tradeoff. Based on several

heuristic experiments, we concluded that the benefits in terms of accuracy are far less than the increase in the computational cost. The real problem comes is assigning material properties to each individual building. To address this constraint we used a single dielectric constant for the whole environment which resulted in a very good correlation between simulation and the actual measurements. Images with propagation results in large cityscapes such as Manhattan are shown in Chapter 7.

3.3.3 Terrain-based environments

Topographical databases of almost all of the United States and many other regions around the world are available in the form of Digital Elevation Models (DEM) from US Geological Survey (USGS) [16]. These databases provide elevation information at several different resolutions ranging from thirty arc-second data to as high a resolution as one arc-second data. With such a wide range of resolutions a data base can have an elevation data point ranging from every 300 meters to 30 meters. However, any resolution coarser than $50 - 70\,meters$ is not very useful for radio propagation since some significant terrain features may not be represented in a piecewise linear representation of such a terrain. Typically this data is divided into equal sized tiles which we called *terrain patches*. For example, a single three arc-second pre-formatted patch extends to almost $10\,x\,7.5\,miles$.

Integrating terrain patches for run time loading

For radio propagation in large terrain based environments, loading the entire database at one time is not only inefficient but sometimes infeasible as well. A single patch (of

$10\,x\,7.5\,miles$) having seventy meter resolution can easily have a few million triangles depending on local variations in elevation, which increases dramatically for higher resolution patches. Thus, it is important to load parts of the environment at a time. We came up with an approach similar to the one discussed earlier in Section 3.1.4. Instead of a two-level intersection grid we used a single-level grid. We computed volumetric bounding boxes around each terrain patch where the size of the patch can be arbitrarily small. It is important to note that if each patch is large, that is, extending more than ten square miles, then a coordinate system has to be chosen carefully such that the curvature of earth does not affect the seamless integration of these patches. Typically terrain patch information is available in spherical coordinate system which is transformed into a Cartesian coordinate system. One of the primary problems with processing large terrain patches is in selecting a suitable origin for the Cartesian coordinate system. If the origin of the Cartesian coordinate system is defined as the intersection of the zero longitude (GMT) and zero latitude (equator), the transformation distorts the shape of the patches that are further away from this origin. This is due to the curvature of the earth. To alleviate this distortion, during a preprocessing step we select a suitable origin from within a region, for example, for northern New Jersey, a location from one of the patches is selected and stored as a reference coordinate with each terrain patch that is converted into the bspt format. If all selected terrain patches for a simulation have the same reference coordinate with the same origin then the radio propagation simulation is performed else the patches are reconverted into the bspt format with a new common reference coordinate system

before a simulation can proceed. Figure 3.2 shows an example of a single terrain patch and then a concatenation of four adjacent patches.

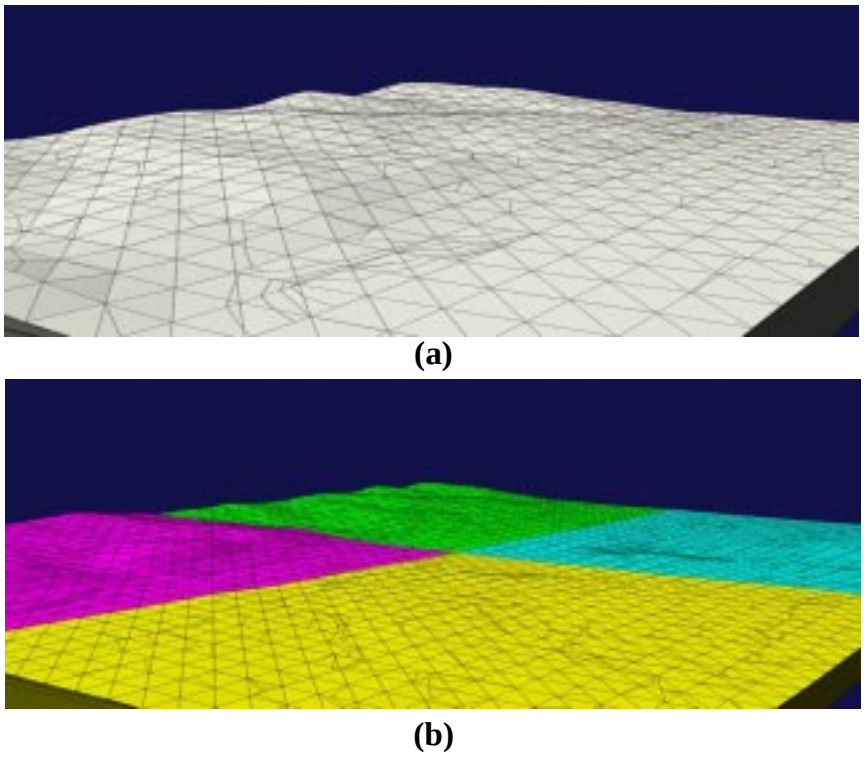

Figure 3.2: The images show a sample terrain of Boulder, Colorado area. For illustrative purposes the images were constructed from a coarse 3 arc-second DEM data. (a) Single terrain patch. (b) Multiple terrain patches (tiles) integrated seamlessly which can be loaded dynamically at run time for simulation.

This simple process of concatenating terrain patches at run time also helps in terms of a suitable user interface. A user can interactively select a set of contiguous (connected) terrain patches, as shown in Figure 3.3, and then run the radio propagation simulation on the environment. The above idea of integrating parts of an environment at run time

can also be extended to other types of environments.

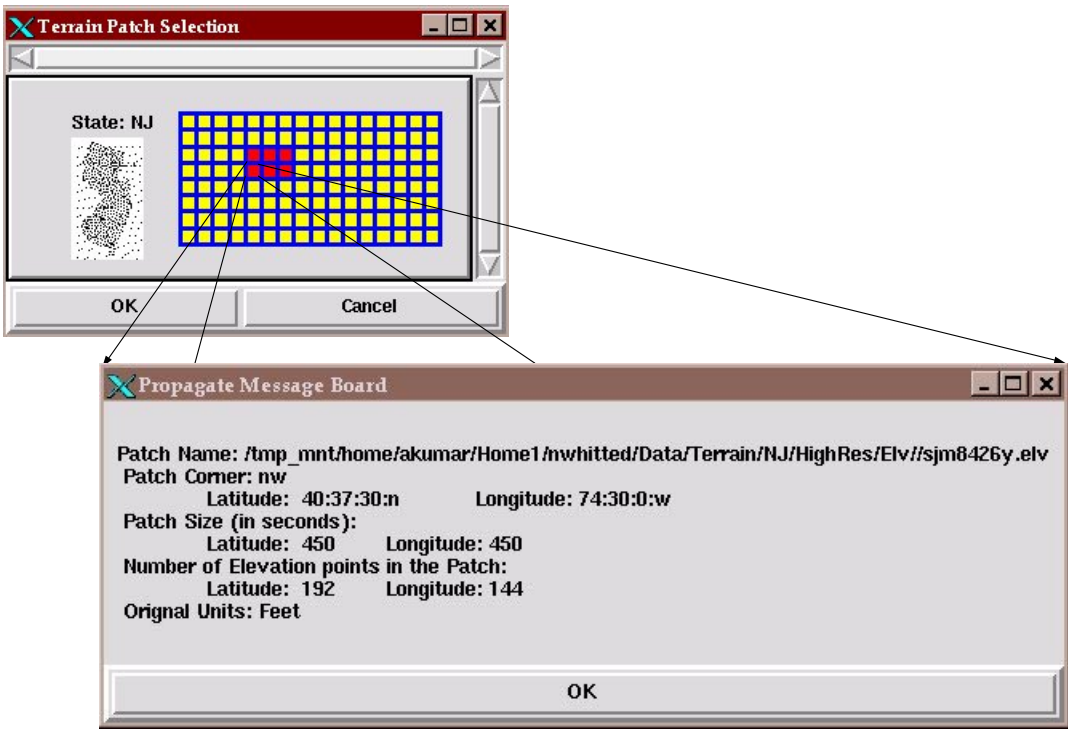

Figure 3.3: The image shows the user interface to select a connected set of terrain patches. A user can select a patch (shown in red), de-select a patch (shown in yellow), or perform an inquiry on a patch.

3.3.4 Integrating terrain with buildings

We propose a simple algorithm to integrate elevation information along with urban city-scapes. We compute the extents of each building in the building database. Then we compute the minimum elevation for the extents of the building under consideration. The

building height should be measured from this minimum point of elevation such that each building individually is above its local ground plane. Since bspt provides us with an easy way of integrating two objects by merging their individual partitioning trees based on boolean operations (such as union, intersection, and difference), terrain, buildings, and other such structures can easily be integrated seamlessly.

Chapter 4

The ray-beam tracing algorithms

In this chapter, we first describe the special requirements that motivated the development of our algorithm. Then we discuss the assumptions that were used to develop the ray-beam tracing algorithm and finally, the actual algorithm. Since the algorithms use a geometric basis to model radio wave propagation they could be used equally effectively to model acoustic waves or wave propagation in some other frequency domains.

4.1 Modeling broadcast mode of propagation and its rationale

A desire to develop a unified approach to modeling energy propagation in different types of environments was an important consideration in the development of these algorithms. As discussed in Chapter 2, most of the light transport algorithms cast rays in the environment starting from the eye position. One of the predominant reasons for this algorithmic

approach is that light transport applications (primarily image synthesis applications) typically have a very small number of viewing positions for which an image has to be generated. In such scenarios, the image-based approach may be well suited. But if the number of viewing positions increases substantially, casting rays from the source may be more efficient.

Modeling propagation from an energy source has several other advantages. We called this approach the *broadcast mode* of propagation. These advantages especially hold for applications such as wireless propagation in radio mobile environments, and in principle, are applicable to modeling acoustic waves as well. Energy from a transmitter, for example, inside large multi-floor buildings, can be traced to several levels above and below the floor where the transmitter is physically located, that is, radio waves are known to travel through walls, floors, and ceilings. This is unlike visible light for which walls, floor, and ceilings act as opaque surfaces. This makes it imperative for radio propagation prediction systems to record signal strength at each of these floors from a given transmitter. An algorithm based on broadcast mode of propagation is efficient because specific paths will only be traced through the environment if they have not attenuated until then. Implementation of this algorithmic approach in such cases gives us a sub-linear time algorithm in the number of receiver locations as we will discuss in Section 7.4. This approach is also suitable for terrain based environments. In such environments receiver locations have to be modeled at a constant height above ground (typical for mobile receiver antennas), which may have continuous changes in elevation. Casting rays from

the source handles such modeling much more naturally.

Another peculiarity of modeling radio propagation that is well addressed by this approach is that often the transmitting source is a directional antenna rather than an omni-directional isotropic source of energy. These directional antennas typically have a complex pattern of energy distribution in three-dimensions. Some examples of these complex patterns are given in Figure 4.1. This directivity of a transmitter is sometimes

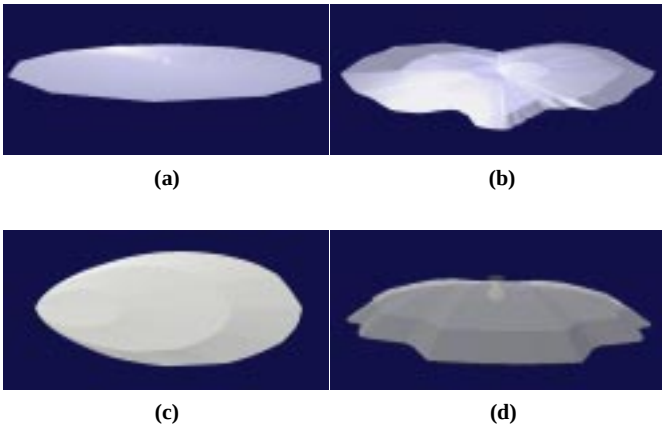

Figure 4.1: These images show three-dimensional renditions of the normalized gain patterns of some of the commonly used directional antennas for radio propagation in urban and indoor environments. (a) An antenna gain pattern with a horizontal beam width of 360° (constant gain) and a vertical beam width of 5°. (b) This has a gain pattern with horizontal beam width of 360° (variable gain) and a vertical beam width of 5°. (c) Horizontal beam width of 120° and vertical beam width of 14°. (d) Horizontal beam width of 65° and vertical beam width of 7°.

achieved by uptilting or downtilting[1] the physical antenna, or rotating it in azimuthal

[1]The same approach of physical tilt is often used to model directional sound sources in large

direction. Casting rays from the source naturally extends itself to the modeling issues of these complex energy sources. The basic idea is to procedurally construct the energy distribution pattern per unit area in three-dimensions at runtime without storing it in the data structure.

4.2 Modeling receiver locations

Another motivation for the development of the broadcast approach was that it facilitates efficient modeling of receiver locations in different environments. For most wireless applications one will notice that mobile user antennas are typically at a constant height, approximately six to eight feet above ground, both while walking as well as while moving in a vehicle. The same is true for users inside buildings, both fixed as well as mobile. For the same reason it is desirable that receiver locations be modeled at a constant height above ground for outdoor and terrain based environments and at a constant height above a particular floor inside a building.

For environments such as inside buildings and large cluttered urban cities that do not extend more than one square mile, it can mostly be assumed that the ground is almost horizontal. In such scenarios, receiver locations can be geometrically modeled as a single horizontal plane. This plane can easily be represented in the partitioning tree data structure without substantial cost in terms of the number of additional polygons. The idea is that for intersection purposes the horizontal plane is tagged with a special

auditoriums.

identifier. If a ray-beam intersects with such a polygon, instead of attenuating the signal, an index of a receiver location is computed based on the point of intersection. Each time such a receiver location is computed, signal strength per unit area of the current beam is recorded in its corresponding data structure (such as a two or higher dimensional array). This data structure also links the polygon to additional information about receivers such as their exact location, their extents, and any other specific receiver characteristics.

The idea of using a single horizontal plane for a set of receiver locations can also be incorporated into the construction of a partitioning tree for the environment. The plane can be used as the first hyperplane which subdivides the environment such that it is placed closer to the root of the partitioning tree. Every time a search is performed on such a partitioning tree, it can be quickly determined if the ray-beam intersects with this plane and received power recorded.

For terrain based environments that have significant elevation variations, a horizontal plane cannot be used to model receiver locations at a constant height above ground as shown in Figure 4.2. The problem manifests itself because of the slope of the ground at the point of intersection. If a vehicle is moving on this slope the receiver antenna mounted on a vehicle is perpendicular to the ground. Depending on the slope of the ground and the spatial distance between two successive receiver locations, power can get recorded in an incorrect receiver bin. The solution comes from the observation that every time an incident ray intersects the ground a receiver receives power once before the incident ray hits the ground and then another receiver location receives power after the ray has

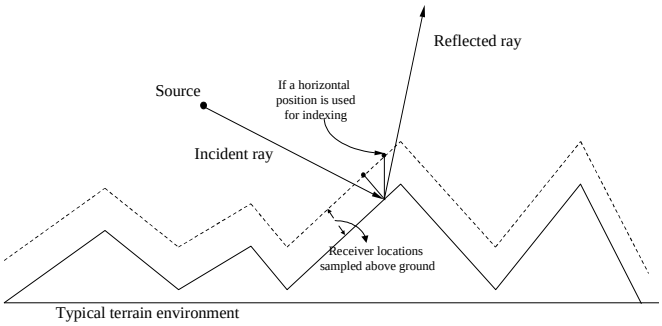

Figure 4.2: Geometric computation to model receiver locations in a terrain based environments that have significant variations in elevation.

reflected off the ground. Diffraction is handled as a special case. For such scenarios, instead of representing the receiver locations as a single horizontal plane and tagging it with a unique identifier, the ground itself is tagged with a unique identifier. Then an index of a particular receiver is computed based on simple geometric calculations.

It is worth noting that the above approach is not restricted to model receiver locations only in a horizontal plane or in two-dimensions. In general, receivers could be modeled as voxels where each receiver has a finite volume in three-dimensions. In this case, the entire environment can be uniformly or non-uniformly subdivided into voxels and received energy recorded in each voxel.

4.3 Sampling error problem

One of the problems of modeling energy propagation from the source is that it is difficult to estimate *a priori* the number of rays that need to be cast from the source. Two rays that start from the same source diverge as the unfolded path length of these rays increases. Though this divergence is a function of the initial angle of separation between rays at the source, the actual distance of separation between the two rays is a function of both the angle and the distance these rays have traveled before they encounter an object. From the perspective of computational efficiency the desire is to cast enough rays such that the environment is sufficiently sampled and yet not over-sampled. At the same time from an accuracy standpoint a prudent approach would be to over-sample rather than under-sample. But then how much does one over-sample? There can always be pathological geometric environments for which a given *rate of sampling*[2] will not be sufficient. The other problem is that typically objects in an environment are not uniformly distributed. To ensure that all regions in the environment are sufficiently sampled, a uniform sampling at the rate of the smallest geometric entity is required, which is surely inefficient. For this reason the conventional wisdom has been that it is inefficient to cast rays from the source.

[2]The *rate of sampling* is defined as the angular separation at which the rays are cast from the source.

4.3.1 An upper bound for the rate of sampling

Can we compute an upper bound for the *rate of sampling*? Here an upper bound is defined as the worst rate at which the rays have to be cast into an environment to ensure that no geometric entities are missed. Clearly, this is dependent on the size of the environment, the size of the smallest object, and the total power of the transmitting source. Let E be the initial energy of the transmitter per unit area. Also let the maximum path loss permitted for the environment be $-X\,dB$ with respect to the initial energy (the negative sign is because it is with respect to the initial energy, otherwise path loss is a positive quantity). If the minimum rate of attenuation given by free space attenuation is assumed to be r^2, where r is the distance in meters from the source to a receiver location then the distance traveled by a ray before it attenuates below the threshold can be defined as

$$r = \sqrt{10^{\left(\frac{X}{10}\right)}} \tag{4.1}$$

or

$$X = 10 \times \log_{10}\left(\frac{1}{r^2}\right) \tag{4.2}$$

If P is the desired separation distance between the two adjacent rays then an angle θ between the two rays is given by

$$\theta = \tan^{-1}\left(\frac{P}{r}\right) \tag{4.3}$$

Notice that for illustration purposes θ is being defined in a single plane and not as a solid angle. For example, with a path loss of $80\,dB$, a signal can travel as far as $10^4\,meters$ of

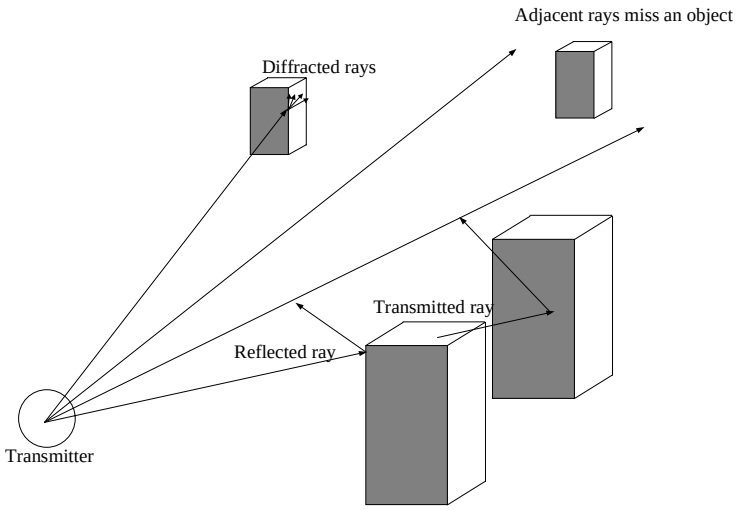

Figure 4.3: Adjacent rays miss arbitrarily large geometric objects.

unfolded path length. To cover a minimum separation of $10\,meters$, rays will have to be cast at every 0.003°. It is evident from the above computation that if all reasonably sized objects in an environment have to be sampled then rays will have to be cast at a very high resolution. Figure 4.3 shows how adjacent rays can miss arbitrarily sized objects if the environment is not sampled adequately.

4.4 A hybrid scheme

From our discussion in the previous sections it is evident that broadcast mode has several advantages for radio propagation. However, it is also evident that a simple ray tracing based adaptation is either extremely inefficient or has to deal with the *sampling error*

problem. To address the issue of sampling error we propose a hybrid scheme of propagation from the source, which is a combination of ray tracing and beam tracing called *ray-beam tracing*. To address the issue of efficiency we propose a scheme that dynamically adapts to the scale of the environment, that is, the algorithm subdivides a ray-beam wherever necessary. We called the combined scheme the *adaptive ray-beam tracing* algorithm.

One can think of a hybrid scheme of propagation as a mechanism of keeping track of the divergence between two successive rays yet at the same time not distorting the physics or for that matter the geometric characteristics of the ray theory. Geometrically speaking, the basic idea of the hybrid scheme is that quadrilateral (four sided) pyramidal beams are cast into the environment, but at the same time a medial ray representing each beam is also maintained in the data structure. The notion of a beam assists the tracer in avoiding geometric entities from falling in between two adjacent rays. The notion of a representative medial ray facilitates faster intersection computation with the planar surfaces in the environment. We now first discuss the main components of the algorithm and then give an overview of the algorithm.

4.4.1 Defining a ray-beam

A transmitter is geometrically represented as a sphere having an explicit surface area rather than just a point. However, the power in any direction is modulated by an antenna gain-pattern and the total power of the transmitter. As a first step, the surface area of

a transmitter is tessellated into four-sided polygons[3]. Here, the size of tessellation can be defined by a user with the default size being every one degree near the equator of the sphere and a coarser size near the poles. This scheme helps us in defining the initial solid angle of a beam in three-dimensions as a function of two angles θ and ϕ, where θ is defined in the horizontal plane and ϕ is defined in the plane perpendicular to the horizontal plane. Finally, for each of the polygons, a beam is defined in the *Model Coordinate System* (MCS) by a virtual up-vector, a virtual origin (which is initially the same as the transmitter location), and a solid angle. Though a beam is cast in MCS, for subdivision purposes it is transformed in a coordinate system called *Beam Coordinate System* (BCS), where BCS is defined such that the virtual up-vector coincides with the z-axis, the medial ray coincides with the x-axis, the apex of the beam coincides with the origin of the coordinate system. The two components θ and ϕ of the solid angle determine the beam extents in y-axis and z-axis respectively.

There were other ray-beam representation schemes which were given some consideration. One of them was to use cones instead of polygonal ray-beams. A cone was defined in terms of a central axis along with a solid angle. A major disadvantage of a cone as opposed to a polygonal beam is that a given surface does not get tessellated completely, resulting in gaps or overlaps in the propagation. These gaps eventually accrue with the total distance traversed. As a result, geometric entities may be missed by the propagation

[3]The tessellation can also be in the form of *n-sided* polygons which may be more suitable with a beam subdivision scheme such as two-dimensional binary space partitioning as discussed in the section on beam subdivision. But at the same time with n more than four, computing a corresponding solid angle will be more complex.

for attenuation purposes. On the other hand, if overlapping cones are used then the same receiver location may receive power from multiple beams (which were overlapping in the first place and should contribute power only once) resulting in incorrect prediction.

4.4.2 Ray-beam subdivision based on the projected area

As a ray-beam is cast in the environment it may intersect a surface at an arbitrary angle. The area the ray-beam projects on the intersected surface is a function of both the incident angle and its solid angle. For a given solid angle, the larger the incident angle with respect to the surface normal, the larger is the projected area. We subdivide a ray-beam into child ray-beams based on this projected area. This is one of the significant differences between the previous approaches and our approach. That is, the subdivision is performed based on the area projected by a ray-beam on the intersected surface rather than subdivision based on the solid angle of a beam. The difference in the two subdivision schemes is shown in Figure 4.4. One will notice that as the incident angle with respect to the surface normal increases to $\pi/2$, a subdivision at the solid angle can still leave arbitrarily large projections on the intersected surface. On the other hand, our scheme can guarantee the required rate of subdivision. The effect of maintaining a subdivision rate based on the projected area is that the number of child beams spawned from a parent beam is much larger if the incident angle is large, which limits the arbitrarily large errors that can otherwise get introduced due to the subtended area approaching infinity.

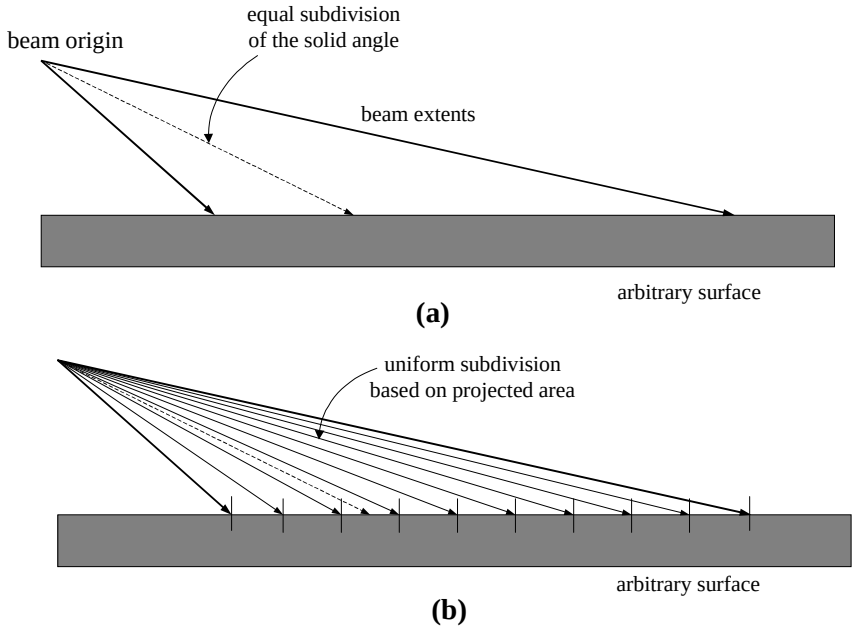

Figure 4.4: A two-dimensional view of a beam subdivision process based on surface sampling. (a) Equal subdivision of the solid angle causes the inequality between the two projected areas on the surface. This disparity increases with an increase in the incident angle. (b) Uniform subdivision of the projected area of the beam alleviates this problem.

4.4.3 Adaptive subdivision

This is a key component of the algorithm to control the sampling error problem. As the path-length of a ray-beam increases so does the area it projects on any surface at a given incident angle. To control this projected area the algorithm maintains a *rate of subdivision* to adaptively subdivide a ray-beam. Initially, this rate is set by the surface tessellation of a transmitter which is a function of the wavelength for a particular simulation run and the approximate size of geometric objects in an environment. However, as the simulation progresses, criterion such as the total power currently contributed by a beam, the distance between the receiver locations, and the size of the geometric entities are also taken into account. The idea of using power is that a beam should be subdivided at a finer resolution if there is a significant component of the initial power remaining in the beam, and at a coarser resolution if the power has attenuated sufficiently. On the other hand, the idea of using this distance between receiver locations and the size of the geometric entities is not to subdivide a ray-beam more than is necessary. Once an intersection point has been computed, both the incident angle at the point of intersection and the distance (the ray-beam has traveled since the last intersection) can be computed. This gives us the projected area and a measure to determine if subdivision is necessary. If a subdivision is performed, new child ray-beams are computed. Each of these child ray-beams are independently cast into the environment and the process continues until the power in a ray-beam attenuates below a user defined threshold or the number of surface intersections exceed a predefined limit.

4.4.4 Beam partitioning

Several beam partitioning schemes can be used to maintain a certain rate of subdivision on an intersected surface. In general, all schemes subdivide the projected area. Each subdivision in turn is defined as a new ray-beam with its solid angle corresponding to the extents of the subdivision. The three main partitioning schemes that we have considered are shown in Figure 4.5. For ease of illustration a beam projection is shown as a rectangular region with the intersected surface completely enclosed by the projection. In a general case if the intersected surface is not completely enclosed by the current beam projection, the adjacent beam projection accounts for the remaining part of this surface. We now discuss the three main partitioning schemes along with their advantages and

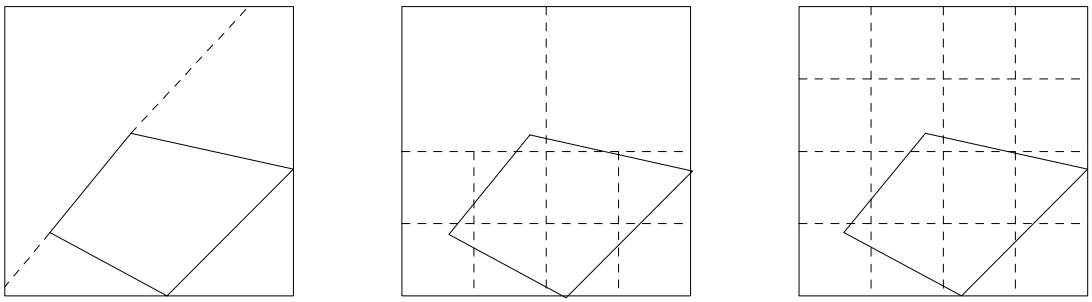

Figure 4.5: Three basic types of beam partitioning. (a) 2D binary space partition of the beam extents. (b) Quadtree partitioning. (c) Uniform spatial partitioning. In (a) each enclosed convex region is a child beam of the parent beam, and in (b) & (c) dotted lines are the new beam partitions.

disadvantages.

- **Two-dimensional binary space partitioning:** Here a beam is partitioned into two-dimensional convex regions to form sub-beams. The scheme primarily uses the edges of a polygonal surface to partition a beam projection. An advantage of the scheme is that an exact subdivision of a beam projection based on the intersected surface can be obtained. A disadvantage of the scheme is that it results in an arbitrary sided sub-beam, which cannot be approximated by a solid angle. Another disadvantage of this scheme is that if the size of an individual beam is not restricted, a situation could arise where complete geometric entities get engulfed by a particular beam size. This may force a particular spatial resolution to be maintained, which is greater than or equal to a combination of the density of the receiver locations for the simulation and the detail of geometric entities. Also notice that beam partitioning in case of Figure 4.5 (a) is not unique as we can have more than one convex partitioning for the same beam projection.

- **Quadtree partitioning:** This scheme recursively subdivides the bounds of a beam projection into four quadrants. The partitioning is determined based on the desired spatial resolution of a beam. The subdivision continues if there are any edges that still need to be resolved. The size of a quadrant in which the unresolved edge falls should be larger than the current beam partitioning resolution. The main disadvantage of the scheme (as in the case of binary space partitioning) is that a particular partitioning may result in large beams such that small geometric entities may be missed. An advantage of the scheme is that is simple to implement and

may not partition a beam unnecessarily. This may be a better partitioning scheme than uniform partitioning in certain cases.

- **Uniform spatial partitioning:** This scheme uniformly subdivides the bounds of a beam projection, depending on the desired spatial resolution. The main advantage of the scheme is that it is able to maintain the level of desired subdivision very easily. The scheme is also very easy to implement. It is important to note that if a uniform quadrilateral beam intersects with a surface at an arbitrary angle it projects an arbitrary quadrilateral. Defining this arbitrary quadrilateral beam with the solid angles θ and ϕ is not possible. For this reason we subdivide an arbitrary quadrilateral with uniform rectangles in two-dimensions thus again being able to define the solid angle for a child beam. A combination of the above partitioning schemes would be the most effective in containing the number of beams to its minimum and still cover the environment sufficiently.

4.4.5 Overview of the algorithm

A high level overview of the ray-beam tracing algorithm is as follows: The algorithm takes two parameters, one a description of the geometric environment and the other is a simulation state describing the characteristics of transmitters, receivers, and the environment. A list of some of these parameters is given in Figure 6.1 and Figure 6.2. The algorithm starts by tessellating the surface area of a source into four-sided polygons. For each polygon a ray-beam is constructed with the origin of the initial ray-beam coinciding

with the origin of the source and its solid angle defined by the extents of the polygon. Each polygonal ray-beam is cast in the environment. If the ray-beam intersects with the plane representing the receiver locations, current power level is recorded for each receiver that the ray-beam intersects. If the ray-beam intersects with any other surface and if it is determined that the ray-beam has to be subdivided based on various criterion (such as remaining power in the ray-beam, distance traveled since the last intersection, and the distance between successive receiver locations) then child ray-beams are constructed. Each child ray-beam is independently cast in the environment. If an edge is detected then diffracted ray-beams are constructed and cast. Else reflected and transmitted ray-beams are constructed and cast. This procedure continues until the power level of a ray-beam falls below a certain threshold. The following is the pseudo-code of the algorithm.

```
Ray-BeamTracing (Bspt env, SimulationState state) {
  For each Tx
    For each tessellated polygon on the surface of Tx
      Construct ray-beam
      Do until ray-beam power < threshold
        If ray-beam intersects the plane representing receiver locations then
          Record current power level in each intersected receiver
        If ray-beam subdivision required then
          For each subdivided ray-beam
            If edge detected then
              Construct and cast diffracted ray-beams
            Else
              Construct and cast reflected ray-beam
              Construct and cast transmitted ray-beam
}
```

4.5 Some practical computational optimizations

There are several optimizations that are easy to implement yet can provide significant computational efficiency when applied to an actual system. Some of these are general optimizations whereas others are very specific to a type of environment or a particular

scenario being modeled.

In urban environments the individual buildings are large enough that the signal gets attenuated long before it exits the building from the other side. In such scenarios if propagation is not being modeled from outside buildings to inside buildings then the transmission paths can completely be ignored. Here buildings act as large opaque surfaces. Thus coverage in most outdoor areas can predominantly be restricted to a combination of reflected and diffracted paths. Another easy to implement optimization is for environments that only have vertical walls and no over-hanging facades or concave geometric objects. In such environments, a ray vector can be tested for their z-direction. If the direction of the ray is in the positive z-direction and the ray origin is above the height at which receiver locations are being modeled, then processing of this ray vector can be terminated. The reason is that this ray-beam will never contribute to any of the receiver locations. Empirical results indicate that this simple optimization can result in a more than fifty percent run time improvement in some scenarios.

Another optimization is the parameterization of the ray-beam subdivision process as a function of remaining energy in the beam which results in significant computational efficiency without much loss of accuracy.

Chapter 5

Diffraction in wireless communication

As we have mentioned in the earlier chapters, for radio propagation diffraction is an important form of scattering energy around wedges. In this chapter, we will first discuss this importance of diffraction in wireless communication, including the theory of diffraction, and its usage in three-dimensional environments. Then we discuss the algorithms we developed to compute diffraction using the ray-beam tracing algorithm described in the previous chapter in an environment that is represented as a binary space-partitioning tree. One of the major challenges of computing diffraction in large environments is to control the computational cost involved in finding an edge in an environment, besides the enormous storage demand it places on any data structure. We will discuss a new runtime algorithm to detect edges dynamically in partitioning tree represented geometries rather

than storing edges statically in the data structure. The algorithm incurs an additional cost of finding an adjacent surface in the bspt.

At an intuitive level, diffraction can be defined as bending of an incident ray at a surface discontinuity to contribute energy throughout the shadow region of the surface. These discontinuities can be in the form of an edge of a wedge, a corner of an object, or any other surface discontinuity. Here, a wedge is defined as a meeting of two planar surfaces and a corner is defined as a meeting of three planar surfaces. It is important to differentiate between refraction where the bending of the incident ray is due to a difference in the densities of the two mediums, and diffraction where the bending of rays is due to a physical discontinuity within the same medium. The phenomenon of diffraction occurs at all electromagnetic frequencies as well as for acoustic waves. However, for electromagnetic waves it is more relevant at some frequencies than others. At a theoretical level this relevance is directly a function of the wavelength with respect to the size of the geometric objects these waves interact with. At a more practical level this relevance is a function of the objectives for which such a simulation is being performed. For example, at visible light frequencies (due to its small wavelength), when light rays interact with objects we use in our everyday life, diffraction as such is minimally noticed by a human eye. As a result, in absence of any ambient light, we do not see objects in the shadow region. Interestingly, for the same reason, we do not see many examples of diffraction being simulated for image synthesis. But at the same time, if light rays are simulated off rough surfaces at a *microgeometry* level then diffraction can be of importance [25]. The relevance

of diffraction is evident for wireless communication, especially in the 1 GHz to 2 GHz range, because many of the objects in our surrounding environment are comparable to the wavelength (0.5 to 1 foot) in this range.

Classical theory of geometric optics (for reflection and transmission) is based on the basic principle that rays travel in straight lines. This implies that it does not provide any theoretical basis for propagation in the shadow region of an obstacle. On the other hand, wave optics provides a sound basis of dealing with the wave nature of the electromagnetic energy. At the same time, there have been attempts at extending the theory of geometric optics to account for diffraction at surface edges and as a result, diffraction coefficients have been formulated for attenuation of the incident signal.

5.1 Theory of diffraction

If a ray is incident on an edge of a wedge and follows only the laws of geometric optics then the area around the wedge can be divided into three distinct regions. The first region is where both direct (line-of-sight) rays as well as reflected rays from the source are visible. The boundary of this region is defined by the ray that reflects off the edge when a ray from the source is incident on the edge. This boundary is aptly known as the *reflection shadow boundary*. The second region is where only direct rays reach. This region is bounded by the reflected ray on one side and by the boundary formed by the incident ray that brushes the edge on the other side. This latter boundary is also known as the *incident shadow boundary*. Finally, the third region is formed between the incident

shadow boundary and the face of the wedge in the shadow region where neither direct nor reflected rays reach. Although this third region is called the shadow region, energy is not totally absent. Experiments reveal that energy does reach this shadow region. These boundaries along with the three distinct regions are shown in Figure 5.1. To

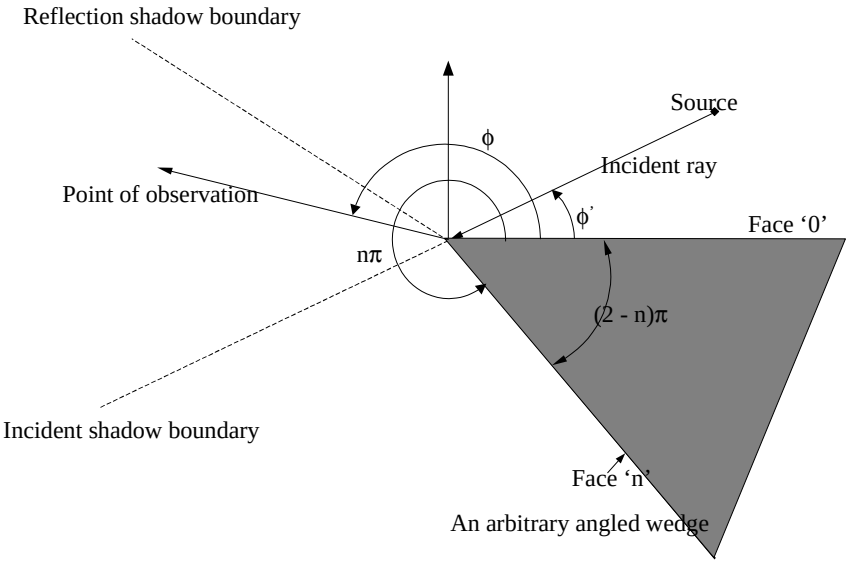

Figure 5.1: An incident ray on the edge of a wedge divides the region around the wedge into three distinct regions. These regions are defined by the reflection shadow boundary, the incident shadow boundary, and the wedge.

account for the energy distribution in the shadow region, extension to geometric optics (also known as *ray optics*) was proposed, which led to what we know as the *geometric theory of diffraction*. To formulate the theory that is applicable to different geometries,

a number of canonical[1] geometries can be considered. Since we geometrically represent a curved surface as its piece-wise linear approximation, we will not consider a diffraction coefficient for curved surfaces separately. Let us assume a wedge with a straight edge as shown in Figure 5.1.

If attenuation around a wedge is to be modeled numerically in an anisotropic media then the total diffracted field at any point $E^d(s)$ around the wedge can be written in a compact form by the following formula [4]:

$$E^d(s) = E^i(Q_d) \cdot \overline{D} \cdot A(\rho, s) \cdot e^{-jks} \tag{5.1}$$

where $E^i(Q_d)$ is the incident field at the point of intersection, $\overline{D}$ is the dyadic[2] diffraction coefficient, $A(\rho, s)$ is the spatial attenuation spreading factor, which is different for a plane, cylinder, and a spherical wave (ρ is the distance between the reference point Q_d at the edge and the second caustic of the diffracted ray; and s is the distance along the diffracted ray from the reference point), and e^{-jks} is the phase variation term. The spreading factor for a plane wave is given by $\frac{1}{\sqrt{s}}$. This is similar to the compact version used to compute the reflected or transmitted field. Notice that the dyadic form is the most general form of a diffraction coefficient which is more suitable for anisotropic media where different polarizations have to considered simultaneously. Typically a scalar form

[1]Diffraction coefficients derived from asymptotic solutions to the simplest boundary-value problems, which have the same local geometries as the point of interest are referred to as canonical problems.

[2]A dyadic is defined by the sum of N dyads where a dyad is defined as a juxtaposition of two vectors A and B as $(A\,B)$ without a dot product or a vector product between them. In general a dyadic has nine terms in a matrix form but typically no more than three dyads are required to represent a dyadic. In our case, only two dyads suffice to define the diffraction coefficient dyadic.

of the diffraction coefficient can be used depending on the application, material type of the wedge, and resolution at which the simulation is being conducted.

There are two dominant approaches to compute the above diffraction dyad $\overline{D}$. They are the geometric theory of diffraction (GTD) and its extension in the form of the uniform geometric theory of diffraction (UTD). Since the diffraction coefficient varies depending on the material type of the wedge, they can be classified into three categories namely: a perfect conductor, a perfect absorber, and neither of these two being classified as a finite conductor. In some cases, the formulation can be simplified based on these assumptions. Let us now discuss the computation of each of these diffraction coefficients, their advantages as well as disadvantages.

A word on terminology

The scalar diffraction coefficients, also known as dyads, are usually defined in terms of their incident polarization. If the incident field is electric then it satisfies the homogeneous *Dirichlet boundary* conditions on both faces of the wedge. Similarly, if the incident field is magnetic then it satisfies the homogeneous *Neumann boundary* conditions on both faces of the wedge. Some references in literature have instead used the terms soft and hard polarization. These terms have been taken from acoustics, where they refer to Dirichlet and Neumann boundary conditions, respectively. Thus in electromagnetics, soft represents an electrically polarized incident field, whereas hard represents a magnetically polarized incident field. For wireless communication applications such as mobile radio propagation,

the terminology of soft represents horizontal polarization while hard represents vertical polarization.

Wedge diffraction in three dimensions

As defined earlier, a wedge is considered as two planar surfaces meeting at an angle less than π. The intersection of these two surfaces is defined as an edge of the wedge. Let the exterior angle of the wedge be defined as $n\pi$, as shown in Figure 5.1, where n is a real number. Then the wedge angle w can be given by $w = 2\pi - n\pi \equiv (2 - n)\pi$. For convenience, the wedge angle is usually referred to by the parameter n.

Keller introduced geometrical theory of diffraction [28] for three-dimensional geometries. The hallmark of the Keller theory for diffraction was the introduction of the *Keller cone* as shown in Figure 5.2. It was defined as a cone formed around an edge away from the incident ray with the axis of the cone making a half angle with the edge. Keller conjectured that unlike the usual reflected ray off a surface, which proceeds in a unique direction after reflection, an *edge-reflected* ray could proceed in any direction inside this cone. This resulted in a mathematical formulation for a GTD diffraction coefficient.

5.1.1 GTD wedge diffraction coefficient

Since GTD is a direct extension of geometric optics, the two shadow boundaries show up as a discontinuity in the diffraction coefficient, where the solution of the coefficient tends towards infinity. These discontinuity result in singularities in the solution of the diffracted field along the boundaries, which can be significant for numerical simulations. At the same

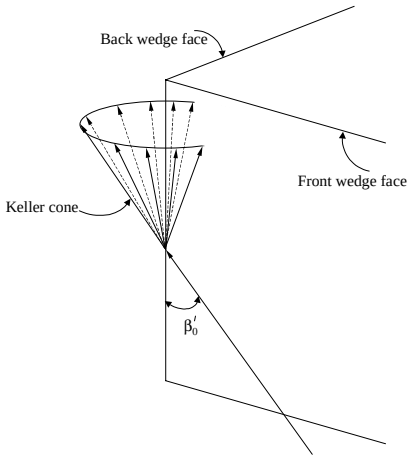

Figure 5.2: A ray incident on an edge of a wedge diffracts along a cone called the Keller cone with its half angle equal to angle formed by the incident ray with the surface of the wedge.

time the formulation is useful in certain scenarios for other types of simulations.

We now define the actual diffraction coefficient. Let ϕ' be the angle formed by the incident ray with the front face of the wedge and ϕ be the angle at which the diffracted field is being observed. If the point of observation is at an angle π from the front face of the wedge then angle $\phi - \phi'$ gives us the reflection shadow boundary and angle $\phi + \phi'$ gives us the incident shadow boundary. For notational convenience let $\omega_1 \equiv \phi - \phi'$ and $\omega_2 \equiv \phi + \phi'$. Then, for a perfectly conducting wedge the diffraction coefficient for soft and hard polarization (referred to by subscripts s and h respectively) can be given by

$$D_s(\omega_1, \omega_2, n) = \frac{e^{-j\pi/4} \sin\frac{\pi}{n}}{n\sqrt{2\pi k}} \left[\frac{1}{\cos(\frac{\pi}{n}) - \cos(\frac{\omega_1}{n})} - \frac{1}{\cos(\frac{\pi}{n}) - \cos(\frac{\omega_2}{n})} \right] \tag{5.2}$$

$$D_h(\omega_1, \omega_2, n) = \frac{e^{-j\pi/4} \sin\frac{\pi}{n}}{n\sqrt{2\pi k}} \left[\frac{1}{\cos(\frac{\pi}{n}) - \cos(\frac{\omega_1}{n})} + \frac{1}{\cos(\frac{\pi}{n}) - \cos(\frac{\omega_2}{n})} \right] \tag{5.3}$$

where $k = \frac{2\pi}{\lambda}$ is a constant dependent on the wavelength of the incident ray called the wave-number. Notice that in the above formulation the singularities at the two boundaries manifest themselves in the form of the angles ω_1 at the reflection boundary and ω_2 at the incident (line-of-sight) boundary. All angles for these computations are given in radians. Figure 5.3 and Figure 5.4 graphically show the computation of the diffraction coefficient

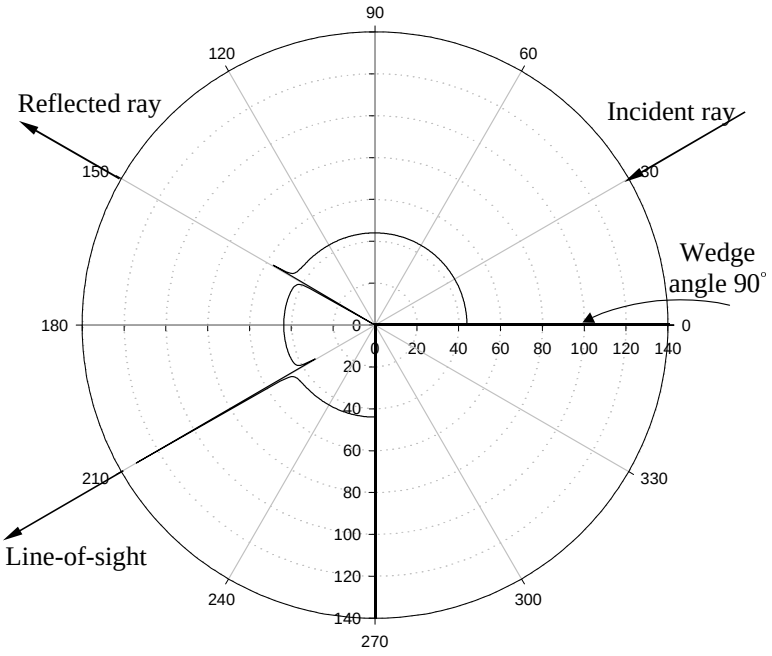

Figure 5.3: Polar plot of GTD diffraction coefficient for a perfectly conducting wedge for soft polarization. For illustration purposes we have assumed the wedge angle to be equal to 90° and the angle the incident ray makes with a surface of the wedge equal to 30°.

for soft and hard polarization respectively. Here, for illustration purposes we assume

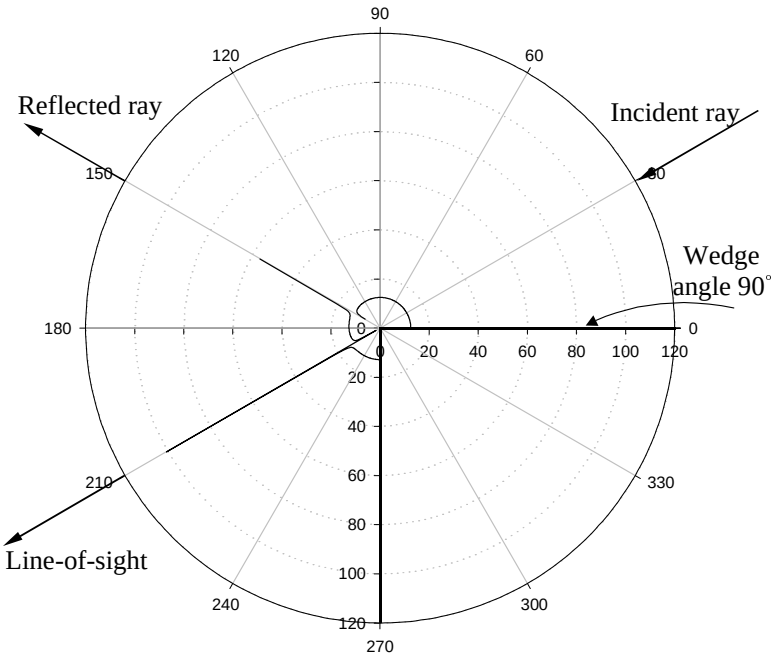

Figure 5.4: Polar plot of GTD diffraction coefficient for a perfectly conducting wedge for hard polarization. For illustration purposes we have assumed the wedge angle to be equal to 90° and the angle the incident ray makes with a surface of the wedge equal to 30°.

that the wedge angle which is a function of n, is 90°, and the incident ray makes an angle of 30° with the top surface of the wedge, (i.e., $\phi' = 30^\circ$). The coefficient has been normalized. However, one will notice that in a real environment the wedge angle can vary between 0° and 180°, and a ray can be incident anywhere around the outside of a wedge. Accordingly, the diffraction coefficient values will change. Notice the singularities at the reflection boundary and the line-of-sight boundary which correspond to the angles $\omega_1 = \pi$ and $\omega_2 = \pi$ respectively. Also, notice that the coefficient does not have any singularities in the shadow region beyond line-of-sight towards the other face of the wedge (away from the incident ray). Therefore, the above formulation suffices if propagation is to be modeled only in the shadow region.

A perfectly absorbing wedge is defined as a wedge, which completely absorbs all incident energy. For such a wedge, the boundary conditions for both soft and hard polarization remain the same. Also, the energy field is primarily given by the rays that directly reach the point of observation rather than the reflection off the wedge. Hence, the diffraction coefficient can be defined in much simpler terms and is given by [19],

$$D(\omega_1) = \frac{1}{\pi - |\omega_1|} + \frac{1}{\pi + |\omega_1|} \tag{5.4}$$

where $\omega_1$is a function of ϕ and ϕ' as described earlier in the section. It may be observed that because no reflections are being considered, the coefficient is not dependent on the wedge angle.

In a typical environment most of the objects are neither perfectly conducting nor perfectly absorbing. This makes it imperative to consider diffraction around finitely con-

ducting wedges. The intuitive idea for such wedges is that because of finite conductivity of a wedge, the reflection and incident shadow boundaries around it may not be well defined. As a result, the diffraction coefficient should account for imperfect reflections as well as some level of signal transmission through the edge of the wedge. The same reasoning is used to account for losses at rough surfaces. These reflection and transmission anomalies can be accounted for by introducing additional terms I_0, I_n, R_0, and R_n, where I_0 and I_n are the correction terms to account for differences between finitely and perfectly conducting wedges for faces 0 and n respectively at the incident shadow boundary. Similarly, R_0 and R_n are the correction terms to account for the differences at the reflection shadow boundary for wedges for faces 0 and n respectively. The diffraction coefficient can finally be stated as

$$D^{\alpha}(\omega_1, \omega_2) = I_0^{\alpha} D_0^{\alpha} \omega_1 + I_n^{\alpha} D_n^{\alpha} \omega_1 + R_0^{\alpha} D_0^{\alpha} \omega_2 + R_n^{\alpha} D_n^{\alpha} \omega_2 \tag{5.5}$$

where D_0^{α} and D_n^{α} are the diffraction coefficient terms for a perfectly conducting wedge for edge faces 0 and n. Here, $\alpha \in \{s, h\}$ refer to the scalar coefficients for soft and hard polarization respectively. The coefficient with the argument ω_1 is discontinuous near the incident shadow boundary, whereas the coefficient with the argument ω_2 is discontinuous near the reflection shadow boundary [8]. One may notice that the correction terms given in 5.5 can vary depending on the geometry or material properties of the wedge. For example, for terrain based environments, typically there is negligible transmission through the terrain. In such cases, I_0 and I_n can be set to unity. Similarly, to compensate for any reflection losses at the reflection shadow boundary the terms R_0 and R_n can be replaced

by the reflection coefficients for the edge faces 0 and n respectively. The absolute values of these constants are unity for perfect conductors. All the above discussion still does not resolve the issue of singularities at the two boundaries.

5.1.2 UTD wedge diffraction coefficient

As discussed in Section 5.1.1, the reflection and incident shadow boundaries around the wedge cause singularities in the diffraction coefficient derived by GTD. To account for these singularities, UTD was formulated. A diffraction coefficient for a perfectly conducting wedge was initially proposed in [31] and is given by the following equations

$$D_s(\omega_1,\omega_2,n) = \frac{e^{-j\pi/4}}{2n\sqrt{2\pi k}\sin\beta_0'} \times \left[\begin{array}{c} \left\{\cot\left(\frac{\pi+\omega_1}{2n}\right) F\left[La^+(\omega_1)\right] + \cot\left(\frac{\pi-\omega_1}{2n}\right) F\left[La^-(\omega_1)\right]\right\} - \\ \left\{R_0\cot\left(\frac{\pi-\omega_2}{2n}\right) F\left[La^-(\omega_2)\right] + R_n\cot(\frac{\pi+\omega_2}{2n}) F\left[La^+(\omega_2)\right]\right\} \end{array} \right] \tag{5.6}$$

$$D_h(\omega_1,\omega_2,n) = \frac{e^{-j\pi/4}}{2n\sqrt{2\pi k}\sin\beta_0'} \times \left[\begin{array}{c} \left\{\cot\left(\frac{\pi+\omega_1}{2n}\right) F\left[La^+(\omega_1)\right] + \cot\left(\frac{\pi-\omega_1}{2n}\right) F\left[La^-(\omega_1)\right]\right\} + \\ \left\{R_0\cot\left(\frac{\pi-\omega_2}{2n}\right) F\left[La^-(\omega_2)\right] + R_n\cot(\frac{\pi+\omega_2}{2n}) F\left[La^+(\omega_2)\right]\right\} \end{array} \right] \tag{5.7}$$

where D_s and D_h are the coefficients for soft and hard polarization respectively. Here, k has the usual meaning, β_0' is the angle between the incident ray and the tangent to the diffracting edge, and $F()$ is the Fresnel transition function. The various components in the argument of this transition function are given by

$$L = k \cdot \left(\frac{s \cdot s'}{s + s'}\right) \tag{5.8}$$

where s is the distance from the source to the wedge and s' is the distance from the wedge to the point of observation and

$$a^{\pm}(\omega) = 2\cos^2\left(\frac{2n\pi N^{\pm} - \omega}{2}\right), \text{ where } \omega = \{\omega_1, \omega_2\} \tag{5.9}$$

Here, N^+ and N^- are the integers that most closely satisfy the following equations:

$$2n\pi N^+ - \omega = \pi \text{ and } 2n\pi N^- - \omega = -\pi \tag{5.10}$$

Notice that for a perfectly conducting wedge R_0 and R_n are also unity as described in the previous section. For a finite conducting wedge R_0 and R_n are suitably modified.

It is important to note that as discussed in Section 5.2 a wedge is detected at runtime rather than storing each of them explicitly in the data structure. Once the edge of a wedge is detected the shadow boundaries can be determined with the help of the incident ray.

5.1.3 Practical considerations to model wedge diffraction

Having discussed the formulation of the two diffraction coefficients and their advantages and disadvantages, we now divert our attention to the practical issues of using these diffraction coefficients. It is important to notice that although the proposed ray-beam tracing algorithms are independent of the type of diffraction coefficient that is being used, there can be significant computational advantages of using one over the other in an actual prediction system. For our prediction system, we used GTD diffraction coefficient instead of UTD. This is because of the way we model diffracted energy. If diffraction is

modeled analytically or numerically then each edge has to be treated as a new source and energy propagated in all three regions around the edge. Singularities in GTD around the reflection boundary as well as diffraction boundary can have a significant impact on the results. We model diffraction only in the shadow region instead of all around the wedge. The idea is that because of the law of conservation of energy, the incident energy gets distributed as diffracted energy all along the angle $n\pi$ around the wedge. However, energy emanating in any specific direction along a diffraction path is little as compared to the contribution from the direct or reflected paths in the other two regions around a wedge. Hence, the overall impact on accuracy of not accounting for diffraction paths in other regions is minimal. On the other hand, since the shadow region is the only region that does not receive energy from any other path, the small level of energy from a diffraction path can have a significant effect in this region. In effect, computational efficiency achieved using this scheme is more significant as compared to the lost accuracy.

The other reason is that for most of radio propagation simulations in large environments, typically the interest is in the energy distribution further away from the wedge rather than very close to the wedge. If the distance between the point of interest and the wedge at which the signal is being diffracted is greater than $100\,\lambda \approx 15 - 30\ meters$, then the results given by the GTD and UTD diffraction coefficients are very close to each other. That is, the singularities in the GTD formulation are more pronounced in the near field around a wedge (less than $5\,\lambda$) than further away from the wedge as shown in [4]. The propagation prediction results as shown in Chapter 7 show that the above

assumptions are within acceptable bounds.

5.2 Modeling diffraction in geometrically defined environments

To model diffraction in a geometrically defined environment, typically edges as well as corners have to be explicitly stored in the data structure. In certain scenarios, these edges could also be approximated by a cylinder where the axis of the cylinder coincides with the edge under consideration. Similarly, a corner could be approximated by a sphere with the center of the sphere coinciding with the corner. This approach may be appropriate for small outdoor environments where each building edge could be represented as a cylinder and a corner as a sphere. However, for large cityscapes or terrain based environments the approach can be computationally very expensive. In such environments differences in elevation can potentially show up as a large number of wedges. Besides, for intersection purposes an independent data structure will have to be maintained to efficiently search for these edges and corners. It is important to note that for any piecewise linear representation such as a brep or bspt, storing wedge information for an environment which has a few million distinct faces could easily increase the size of the data structure by multiple times. In the following discussion, it may appear that we have used edge and wedge interchangeably. However, wedge is referred to as the three-dimensional object with two distinct intersecting faces, whereas an edge is defined as the line of intersection where the two faces are intersecting.

5.2.1 Dynamic edge-detection in partitioning-tree represented environments

We describe an efficient method to dynamically determine wedges in partitioning-tree represented environments. To determine if a wedge has been detected in such environments, we simply use ray intersection. That is, instead of computing the first point of intersection with a ray, we compute the first two successive points of intersection with the ray. This process is described pictorially in Figure 5.5 for a bspt representation.

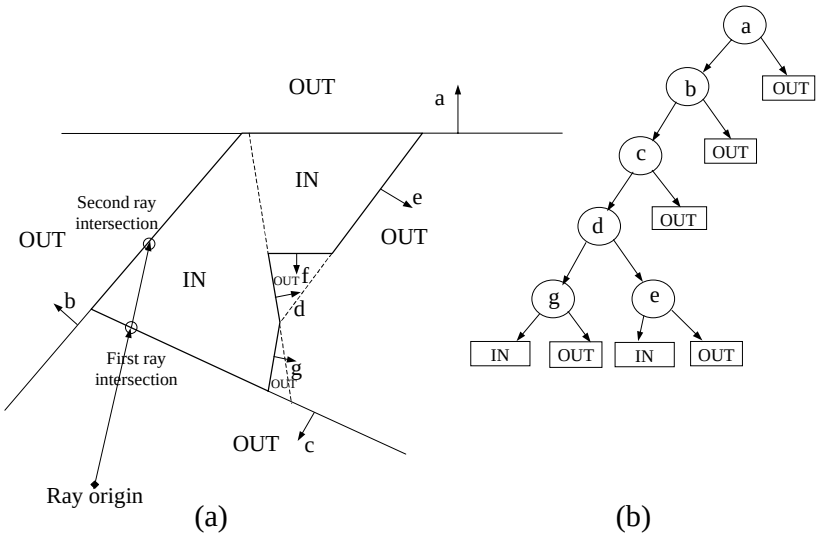

Figure 5.5: An object represented as a binary space partitioning tree. It also illustrates a ray intersection with the tree to compute two successive points of intersection.

The first two intersections determine the planes that intersect with the ray. The

basic idea is that if a wedge is encountered then the first intersection (as the ray travels from an 'outside' region to 'inside' region) will represent one side of the wedge and the second intersection (as the same ray further travels from 'inside' region to 'outside' region) will represent the second side of the same wedge in a partitioning-tree data structure. However, the above scenario may not necessarily represent a wedge. The other scenario may be that although the detected geometry is a wedge it still may not qualify for diffraction because the point of intersection is too far away from the edge. To eliminate such spurious cases and determine if the intersected point is close to an edge we perform additional computation.

Trigonometric computation is used to calculate distance between a point of intersection and the corresponding opposite face of the possible wedge. Let pd_1 be the perpendicular distance from the first intersection to the second surface of the wedge shown as AB in Figure 5.6. Similarly, let pd_2 be the perpendicular distance from the second intersection to the first surface of the wedge shown as CD in Figure 5.6. Also let d_1 and d_2 be the actual distances of first and second point of intersection from the edge shown as AE and CE respectively in the figure. To compute the distance from a surface we first compute the equation of the planar surface and then proceed to compute the perpendicular distance as well as the actual distance.

Let $\overrightarrow{N_1} = (l_1, m_1, n_1)$ be the normal to the first surface and $\overrightarrow{N_2} = (l_2, m_2, n_2)$ be the normal to the second surface, then perpendicular distance from the first intersection point

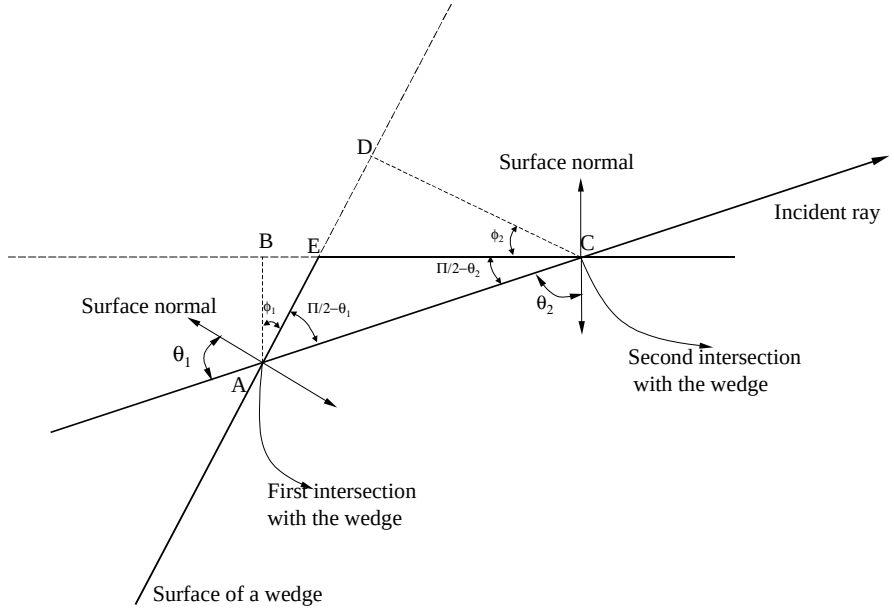

Figure 5.6: Dynamic determination of an edge as depicted in a two-dimensional case. Notice the first two points of intersection are computed for each ray.

to the second surface is given by

$$pd_1 = l_1 \cdot x_2 + m_1 \cdot y_2 + n_1 \cdot z_2 + D_1 \tag{5.11}$$

where D_1 is from the equation of the first plane[3]. Similarly, perpendicular distance from the second intersection point to the first surface is given by

$$pd_2 = l_2 \cdot x_1 + m_2 \cdot y_1 + n_2 \cdot z_1 + D_2 \tag{5.12}$$

Now, θ_1 is the angle between the incident ray the and the normal to the first surface, given by (5.13) and θ_2 is the angle between the incident ray and the surface normal to the second surface given by (5.14)

$$\theta_1 = \arccos(-\overrightarrow{(R)} \cdot \overrightarrow{N_1}) \tag{5.13}$$

$$\theta_2 = \arccos(\overrightarrow{R} \cdot \overrightarrow{N_2}) \tag{5.14}$$

where $\overrightarrow{R}$ is the incident ray. We will notice that $\triangle ABC$ gives us the relation given by equation (5.15)

$$\phi_1 + (\pi/2 - \theta_1) + (\pi/2 - \theta_2) + \pi/2 = \pi \tag{5.15}$$

Solving the previous equation for ϕ_1, we get

$$\phi_1 = \theta_1 + \theta_2 - \pi/2 \tag{5.16}$$

Finally, d_1 can be computed by

$$d_1 = \frac{pd_1}{\cos(\phi_1)} \tag{5.17}$$

[3]Equation of the plane is given by $A \cdot x + B \cdot y + C \cdot z + D$ where (A, B, C) is the surface normal.

Similarly, d_2 can easily be computed using pd_2 and ϕ_2, where $\phi_2 = \phi_1$. Once these distances have been computed a wedge is detected if either of these distances is less than a user setable ϵ called the *edge-detection factor.* We used the above method to detect edges in outdoor urban environments for roof-top and street-level deployment of transmitters as well for diffraction in terrain based environments with good correlation with the measured data. Some of the results for comparison between the measured and predicted values are shown in Table 7.1.

One of the cases that is currently not accounted for by the above edge detection algorithm is inside buildings where the walls are extremely thin as shown in Figure 5.7. The problem in this special case is that even though the point of intersection may be

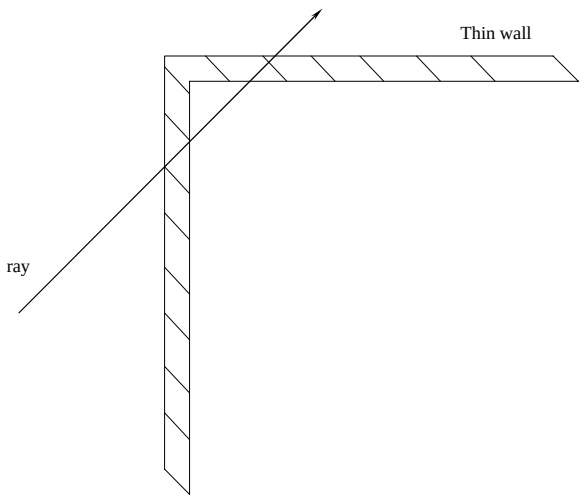

Figure 5.7: Edge-detection of thin walls inside buildings.

within the limits of an edge-detection factor (close to an edge), the edge itself may not get detected by the algorithm. This is because to detect an edge we only compute the first

two ray-surface intersections. In this case, the first two intersections will compute the two intersection points on each side of the same wall. Since the two sides are parallel to each other, the above computation for edge detection will not form the $\triangle AEC$ as shown in Figure 5.6. As a result, the diffraction event will not be initiated by the ray-beam tracing algorithm. Instead, a reflection and a transmission event will be initiated. However, the case does not pose a significant problem for correct computation of power distribution in the shadow region for environments such as inside buildings which may have thin walls. This is because for propagation inside buildings the shadow regions receive most of the coverage through transmission and reflection paths instead of diffraction paths. As we can notice, the signal in this case will get contributed in the shadow region through two successive transmission paths.

5.2.2 Frequency based parameterization

The *edge-detection factor* itself can be defined in terms of the frequency (wavelength), thus providing an important parameterization. Besides being a function of wavelength, ϵ can be a function of the sampling rate at which received signal is being sampled or the geometric detail (size of geometric entities) in the environment. In a wireless propagation and analysis simulation system, such parameterization gives a useful control to the user with which they can relate to rather than the factor being an abstract constant.

5.2.3 Searching a partitioning tree for successive intersections

One method of searching for successive points of intersection in binary space partitioning tree is to every time start a search from the root of the tree. This can be expensive if the tree is large. Another easy method is to remember the last node at which the intersection was successful. Any subsequent search could then start from this node. This can be useful in cases where the *locality of region* has been maintained while constructing the partitioning tree. In any case, the worst case scenario is the same as searching from the root, but on an average the search can be twice as fast.

Chapter 6

System considerations for wireless communications

A radio propagation prediction system based on the proposed algorithms can be used to efficiently compute several different parameters. In this chapter, we will initially discuss some issues specific to simulation of radio propagation and then discuss computation of some of these parameters such as path loss and actual signal strength distribution from a given source. We will also briefly describe the propagation system *Propagate* which simulates these computations. Some other parameters such as delay spread, angular spread, and spatial diversity will be discussed in Chapter 7.

Computation of some of these parameters can also facilitate an efficient installation of a cellular network of transmitters. Typically, a transmitter is a generic term used to indicate that the signal is being transmitted from that location. Similarly, a receiver

is a generic term used to indicate that the signal is being received at that location. However, in wireless communications there are several variations of this terminology that can be used depending on the context. If only the electronics is being considered then both a transmitter or a receiver can be referred to as an antenna. Another terminology commonly used in this context is a *base station* and a *mobile,* where a base station includes the electronics of an antenna as well as the computing resources necessary to be able to communicate with the other parts of a network. A mobile on the other hand is the device carried around by a user and includes an antenna and other computing resources. It is important to note that both a base station and a mobile can operate as a transmitter (Tx) as well as a receiver (Rx). In our discussion, unless specifically mentioned, we have implicitly referred to the base station as a transmitter and the mobile as a receiver.

6.1 Modeling issues specific to wireless communication

Signal attenuation also known as fading in wireless communication environments can primarily be divided into two types: large-scale fading and small-scale fading. Large-scale fading is typically the result of attenuation due to obstructions including distance separation between a transmitter and a receiver. On the other hand, small-scale fading is a result of the multi-path phenomenon observed in wireless communication. There has been considerable debate on whether it is important to consider phase and polarization of an incident ray to accurately predict either large-scale or small-scale fading.

6.1.1 Importance of modeling phase

The phase of a ray (which is dependent on the wavelength) varies with the distance traveled in free space as well as the number of obstructions the ray has encountered. Every time the ray interacts with an obstruction it changes its phase by a right angle (90°). The first and the more pronounced form of phase variation amongst rays reaching the same location from the same transmitter manifests itself in the form of time-delayed multi-path rays. On the other hand, as we move away from the transmitter, a ray is reflected, transmitted, and diffracted by many objects such that its phase is sufficiently randomized not to have a significant net impact on the total phase variation of the signal. Thus, the net phase variation from the latter can be assumed to be zero. It is worth noting that although for our purposes modeling phase variation from the time-delayed signal suffices, computing phase due to obstructions is equally simple.

6.1.2 Importance of modeling polarization

The polarization of a radiated wave is defined as that property of a wave whose direction and relative magnitude of the electric field vector varies with time. The polarization of the electric field of a wave can be decomposed into two orthogonal components, namely horizontal (also known as perpendicular) polarization and vertical (also known as parallel) polarization. If the electric field vector is perpendicular to the plane of incidence[1] it is called perpendicular polarization. Similarly, if the electric field vector is parallel to the

[1]The plane of incidence is defined as the plane containing the incident ray vector and the normal to the incident surface.

plane of incidence then it is called parallel polarization. Another terminology commonly used is with respect to the interface plane. If the electric field is parallel to the interface plane then it is called horizontal (or E) polarization and if it is perpendicular to the interface plane then it is called vertical (or H) polarization. The total field is the vector sum of these two fields.

If a vertically polarized wave is radiated in the environment, part of the energy after interacting with the environment gets converted into horizontally polarized energy and the rest of it remains as vertically polarized. Similarly, a horizontally polarized signal gets converted into the two separate components. If both the radiated wave as well as the received wave are of the same polarization then any energy converted into the other type of polarization is considered as attenuation loss resulting from an interaction with the environment. Empirical studies [65] as well as our own field trials [63] have shown that for mobile communication if the radiated wave is vertically polarized then the energy converted into horizontal polarization is almost negligible, that is, more than $10\,dB$ lower than the received vertically polarized signal. For this reason, it is not necessary to explicitly model polarization for the computation of signal strength attenuation. But at the same time, if polarization diversity is being considered either at the base station or at the mobile then explicit modeling of polarization becomes important.

6.1.3 Modeling wideband vs. narrowband propagation

For wireless communications different access schemes can be used to modulate signal propagation in an environment. These schemes use varying modulation bandwidths. For example, wideband propagation (in case of code division multiple access) uses up to a $5\,MHz$ wide channel whereas narrowband propagation (in case of time division multiple access) uses only $30\,KHz$ wide channel. A very natural question that can be posed is: can we use the same ray-beam tracing algorithm to model both types of signals? Theoretically, *coherence bandwidth* is defined as the maximum separation for which the signals are still considered to be correlated. Narrowband is defined as where the bandwidth is substantially narrower than the coherence bandwidth and wideband is defined as where it is much wider than the coherence bandwidth. A wideband signal typically experiences selective fading due to the different frequency components having different fading characteristics. As of now we have not performed detailed simulations on this aspect but the prediction system can run at several different frequencies to compute the individual fading characteristics. The reasoning being that the only difference between different runs is the variation in the wavelength of the signal.

6.1.4 Reflection and transmission

As stated earlier, reflection can be modeled both as specular as well as diffuse. The difference being that specular reflection follows Snell's law of reflection to compute the direction of the reflected ray whereas diffuse reflection casts rays around the point of

incidence as a function of a predefined distribution.

For attenuation of signal due to reflection (or transmission) typically an obstruction can be treated as a single homogeneous material. For example, this is a good approximation for ground in a terrain based environment. However, walls inside buildings are usually made of multiple layers of sheet rock (of different thickness) interspersed with layers of air for better insulation. If this is modeled by casting a reflection and a transmission ray-beam at the boundary of each layer, it can computationally be extremely expensive. An alternative is to model them as a single obstruction with the computed attenuation corresponding to the number of layers in a wall. The coefficients for multi-layered objects have been described in [6].

Reflection and transmission coefficient

The reflection coefficient for an arbitrary number of layers is given by

$$R = \frac{Z_{in}^{(n)} - Z_{n+1}}{Z_{in}^{(n)} + Z_{n+1}} \tag{6.1}$$

where n is the number of layers. Computation of these coefficients can be expensive. To alleviate this expense at run time we pre-compute these coefficients for every material type in the environment as well as for incident angles at close intervals and store them in a table. Similarly, the transmission coefficient is given by

$$T = \frac{2Z_{in}^{(n)}}{Z_{in}^{(n)} + Z_{n+1}} \tag{6.2}$$

6.2 Discussion on material properties

To design a system which can be used to predict signal propagation in a site-specific environment in lieu of an actual field trial, it is not only necessary to model the shape and size of objects as exactly as possible but also be able to define their material properties as close to reality as possible. This is because signal attenuation by a surface is a direct function of its material properties. But at the same time, depending on the geometric detail being represented, determining a homogeneous material property for an object and assigning individual surfaces a material type can be rather challenging.

The range of objects (that may affect wireless propagation in an environment) can broadly be classified as materials that are dielectrics (also known as insulators), magnetics, or conductors. There can be various shades or combinations of these but for simulation purposes the above classification suffices. Permittivity is a measure of polarizability for dielectrics and is given by ε, with units as Fm^{-1} (farads / meter). Similarly, permeability is a measure of magnetism of the material and is given by μ, with units as Hm^{-1} (henries / meter). Finally, conductivity is a measure of the ease with which current flows of the material and is given by σ, with units as Ωm^{-1} (ohm / meter). A term relative permittivity[2] can be defined as

$$\varepsilon_r = \frac{\varepsilon}{\varepsilon_0} \tag{6.3}$$

where ε_0 is the permittivity of vacuum. Though permittivity and relative permittivity

[2]In some literature, inverse of the square root of relative permittivity is referred to as *index of refraction,* a commonly used constant in most of the rendering and global illumination systems.

have the same numerical value – a constant, relative permittivity is unitless since permittivity of vacuum is 1 by definition . This is also known as the *dielectric constant*. Similarly, we can define a relative permeability term which is a dimensionless quantity, given by $\mu_r = \frac{\mu}{\mu_0}$. But since most of the materials except ferromagnetics have permittivity close to 1, for practical purposes μ_r can be treated as unity. The intrinsic impedance η is given by $\eta = \sqrt{\frac{\mu}{\varepsilon}}$, and signifies the ease at which the wave can travel in the medium.

The radian frequency is given by $\omega = 2\pi f$, where f is the frequency in Hz. If the displacement current is much greater than the conduction current, that is, $(\omega\varepsilon \ggg \sigma)$ then the medium behaves like a dielectric. If $(\sigma = 0)$ the medium is a perfect, or lossless dielectric, whereas if σ is not equal to zero then the medium is lossy or imperfect dielectric. Finally, if $(\omega\epsilon \lll \sigma)$, that is, the conduction-current is much greater than the displacement-current then the medium is classified as a conductor [4].

It is important to note that frequency is an essential component in determining whether a medium is a dielectric or a conductor. For example, the earth behaves like a conductor at around 1 kHz whereas at microwave frequency of 30 GHz it behaves like a dielectric. The better the dielectric, the lower the number of free electrons which can transfer charge from one place to the other; hence its conductivity is low and it is a better insulator.

For indoor environment we identify broad categories of types of material and assign a type to each surface. For outdoor environments we can use the same procedure for individual buildings or use a broad material type for the whole environment. For terrain-

based environments we assign a single material type to the whole terrain.

6.3 Path loss and signal strength computation

Algorithmically the computation of both path loss and signal strength are very similar. The basic difference between the two is that path loss is the relative attenuation of power between the source and the receiver without regard to the specific antenna characteristics such as signal gain pattern, number of branches, and the beam width of each branch, whereas, signal strength is the absolute attenuation between a transmitter and a receiver. For path loss computation the source transmitter is typically considered to be an omni-directional isotropic antenna. In the most general form, we can define path loss as a product of the attenuation a signal encounters at each obstruction starting from the source until it reaches the receiver location. The path loss for each propagation path between the source and the receiver location can be given by

$$pl_p = \frac{\prod_i \eta_i \prod_j \zeta_j \prod_k \xi_k}{|dist_p|^2} \tag{6.4}$$

where pl is the path loss for a particular path p, i is the number of reflections the path encounters and η_i is the reflection coefficient of the i^{th} surface. Similarly, j is the number of transmissions with ζ_j being the transmission coefficient of the j^{th} surface and k is the number of diffractions with ξ_k being the diffraction coefficient of the k^{th} wedge. Besides the attenuation due to surface obstructions a propagating wave also attenuates in free-space, which is given by the inverse of the square of total path length. One will notice that in a vacuum free-space attenuation is given by an exponent of *two*. Sometimes a

higher exponent is used to account for other hard to model atmospheric attenuation such as moisture content in air or attenuation due to vegetation. Since wireless propagation accounts for all paths reaching a sample point from the same source, the total path loss Pl is defined as

$$Pl = \sum_{p} pl_p \tag{6.5}$$

In general, computing path loss with an isotropic source can be more expensive than computing it for an actual transmitter with a complex gain pattern. This is because for path loss computation, paths in all directions have to be traced until a specified attenuation is reached irrespective of the initial power an individual path may have. On the other hand, one of the advantages of using path loss computation, despite additional computational cost is in a scenario where repeated simulations have to be performed on the same environment with the same transmitter location except a change in the initial power or gain pattern of the transmitter. In such scenarios the actual attenuation can be computed by simple post-processing on the path loss results.

6.4 A propagation prediction system

We developed an interactive system called *Propagate* that models radio propagation in several different types of environments. The graphical user interface of the system was written in Tcl/Tk, where Tcl is a high level scripting language and Tk is a library of graphics primitives. Another library called *Polymnia*[3] was used to represent our geometric

[3]The library was primarily written by Bruce Naylor at Bell Labs.

environment as a binary space partitioning tree. A user can interactively select and visualize an environment, set transmitter and receiver characteristics, and place them in the environment for a simulation. Some of the transmitter and receiver characteristics that can be modeled by the system are shown in Figure 6.2. Similarly, some of the

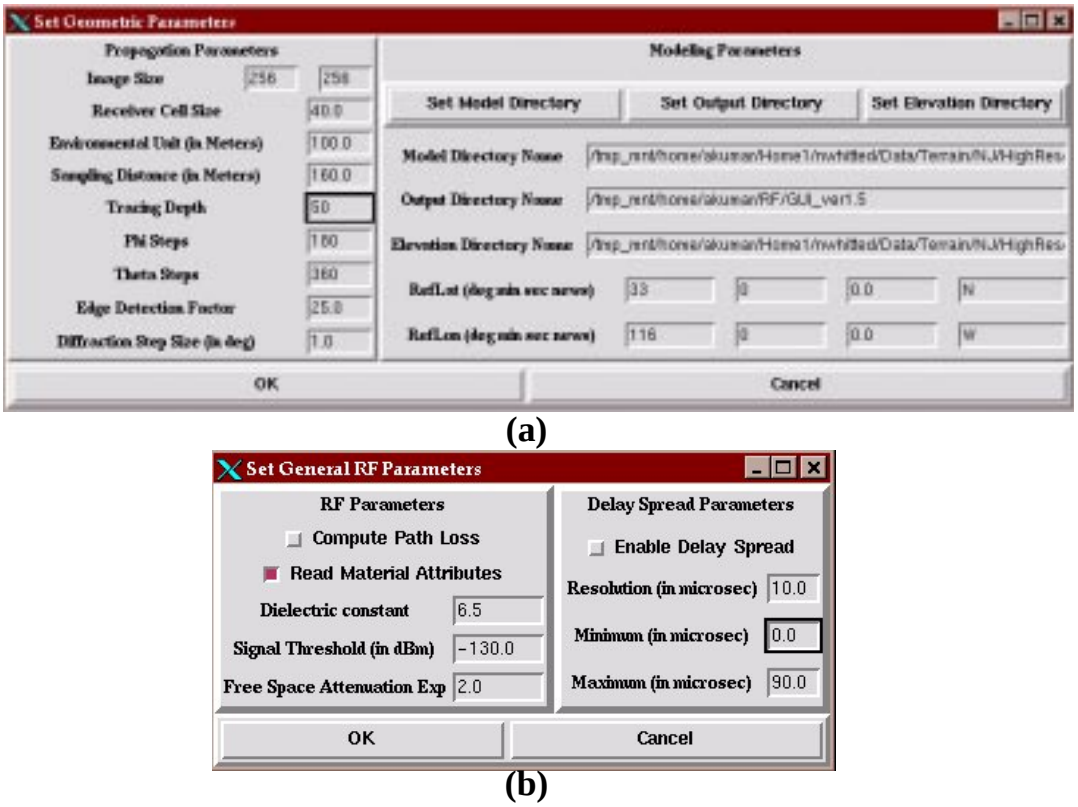

Figure 6.1: (a) A list of parameters that control the geometric properties of the ray-beam tracing algorithm. (b) A list of general radio propagation parameters.

parameters related to a geometric environment including parameters that control the size of a ray-beam is shown in Figure 6.1. Some options related to the type of environment or the type of propagation paths that can be independently simulated by the system are

shown in Figure 6.3.

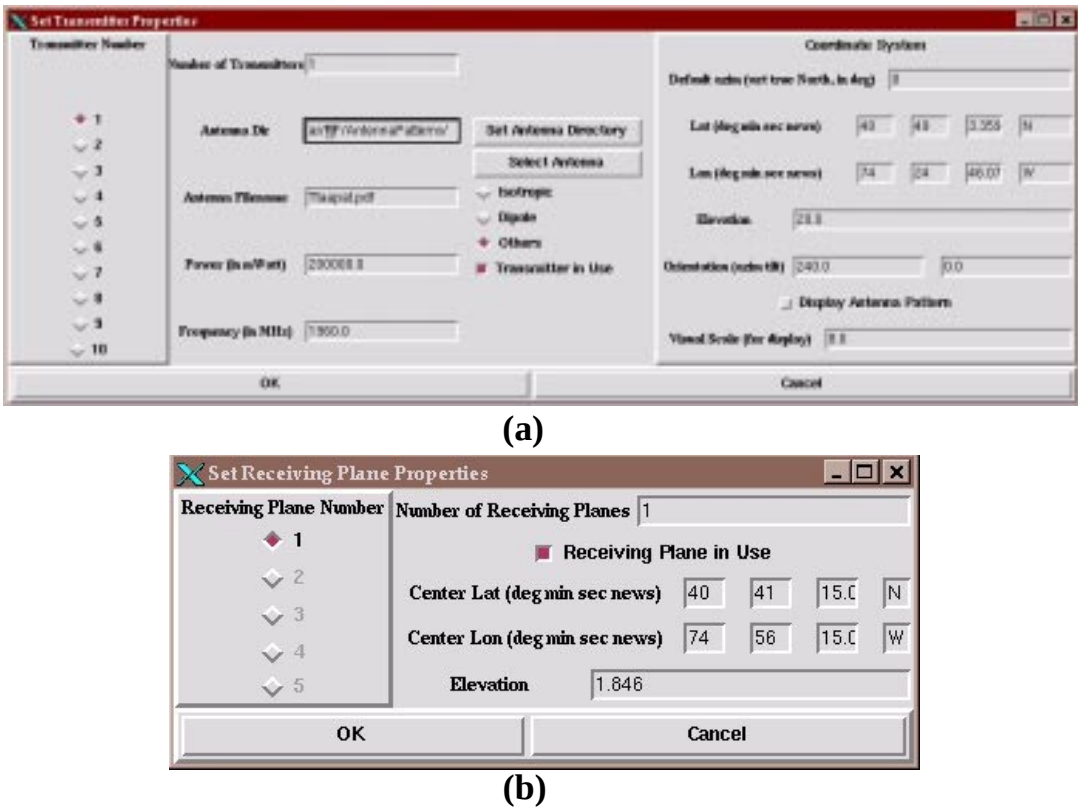

Figure 6.2: (a) A list of characteristics that define a transmitter. (b) A list of characteristics that define a set of receiver at a specific height.

6.4.1 Texture maps for interactive querying of signal strength

In an interactive system, it enhances a users experience much more if one can inquire the level of signal strength at a location as one is navigating through that environment. For such querying of signal strength, we construct a sample plane which represents a set of receivers at a constant height and integrate it into the environment by constructing a

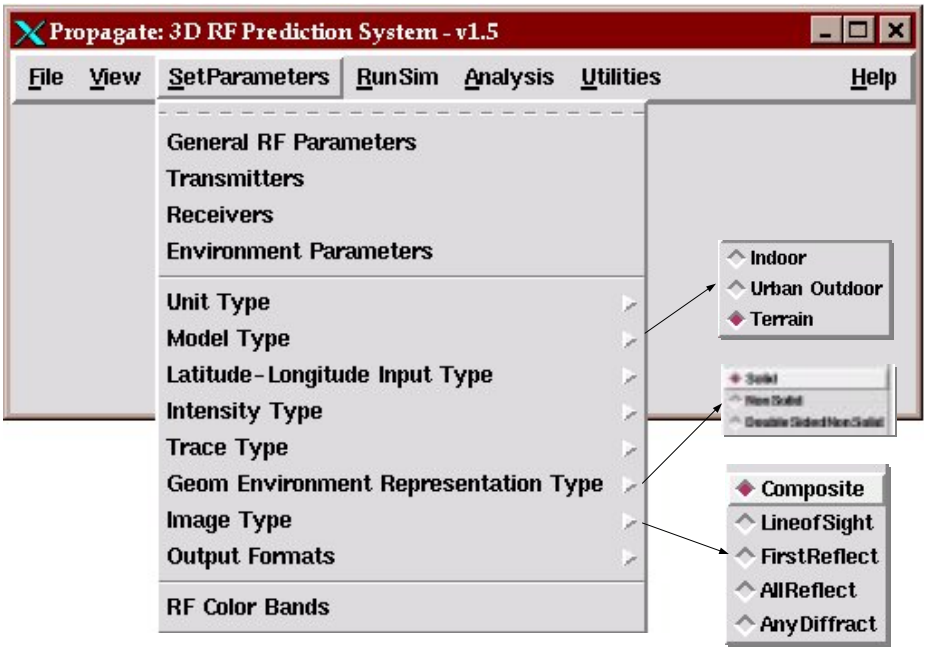

Figure 6.3: A list of some of the options for the simulation system.

new partitioning tree. We used *Sculpt,* a drawing program developed by Bruce Naylor to display our three-dimensional environments. We initially convert a predicted coverage map into a texture map which is mapped onto the respective sample plane for visualization. Each of these sample planes are tagged with a unique identifier and the predicted values are stored in a data structure. During interactive navigation a user can point at any receiver location on any of the sample planes. The system computes an index corresponding to the point of intersection which in turn reports the signal strength for the corresponding receiver location. Some examples of coverage maps used as textures is given in Section 7.2.

Chapter 7

Results

We now present the overall results based on the algorithms that have been developed. Some of the results have already been presented along with individual sections where the respective algorithmic issues were discussed. The algorithms were compared based on how well they predict power distribution in a given environment and also based on their run time efficiency. We compared the results of our algorithms with the actual measurements in several different types of environments. For our tests we will simulate a typical mobile communication application. The scenario is that a transmitter (at the base station) transmits radio signals into an environment. It is desirable to determine whether a mobile moving in the environment is able to receive an acceptable signal from this transmitter. There can be several variations to this basic scenario which can be transformed into this basic scenario.

7.1 Validation of the prediction model vs. field measurement data

For the current system to have any usage as a practical RF engineering tool, the prediction model needs to be compared with the actual measurements and results validated for similar conditions in the field. We will devote the initial part of this section to discuss the process of taking measurements. Then we will describe the process of comparing these measurements with the predictions from the simulations and the justification of why this process is acceptable. The measurement-taking process is indeed very tedious to say the least. In fact, the real value (in terms of the saved man-hours) of such a tool is only realized if one has been involved in taking such measurements!

The algorithms and the simulation system built around it was validated by comparing the predicted values in an environment and the values of the corresponding measurements. This also brings up the issue of an averaging scheme that can be used for such a comparison.

7.1.1 Discussion on the measurement-taking process

There are several ways in which measurement and the predicted results could be compared. One possible method is to do a point-to-point comparison between the predicted and measured power at a given location. In some senses this point-to-point comparison is actually an area-to-area comparison where the area may be a function of the wavelength. This brings in the question of what is the optimal area for comparison. Obviously, one

problem is present in the measurement-taking process. Typically, in urban outdoor as well as terrain based environments, measurements are taken by connecting a mobile receiver along with a global position system (GPS) receiver to a computer. The mobile receiver records the amplitude of the received power and the GPS receiver records the geographical coordinates at that location. This setup is placed in a van which is driven around in an environment. For practical reasons, it is difficult for a van to move slower than $20\,miles/hour$, especially if these measurements are taken during the day. This speed translates to approximately $9\,meters/sec$. On the other hand, the sampling rate of the currently available GPS receivers is approximately once a second. This implies that received signals can be sampled approximately every $10\,meters$. However, location information at higher resolutions can be obtained by interpolating the intermediate positions. Another problem with the currently available GPS receivers is that the true receiver locations are dithered (for military reasons) such that it is off by a few tens of meters. This may not be a significant distance from a visual perspective in a large city of several square miles. But for area-to-area comparison between measured and predicted values this can be quite significant, especially, if the area being compared is less than every ten meters. One method of accounting for this error can be by using another reference GPS receiver and then subtracting the common dithering factor. However, for our purposes of validating the simulation model with the measured results we overlay the measured GPS locations on an actual street map to account for any spatial discrepancies.

A similar setup can be used for measurements inside a building. Instead of a van, a

cart carrying all the equipment can easily be used. The main difference is in the process of correlating a location with the received signal strength at that location. The problem is that for a GPS receiver to function properly it should initially have line-of-sight with at least four to five satellites. Once the GPS receiver has locked its position, the receiver should then be locked to at least three satellites at all times, which is obviously not possible inside a building. The line-of-sight with any three satellites is necessary for the triangulation algorithm to compute a unique position on earth. An alternative method to obtain location information is to manually map the path of the cart through the corridors to individual data points. The positioning may not be the most exact but is within acceptable range of error. Here the locations can be at a much finer resolution.

The other issue for averaging is from the point of view of propagation, that is, small-scale fading causes deep nulls or other sharp variation in signal amplitude within a couple of meters. Hence, some spatial averaging is necessary to average out these small-scale fading effects. Typically, the mean signal strength received at a particular location is computed by averaging total power received over an area of $5\,\lambda$ to $40\,\lambda$. This translates to a range of 1 *meter* to 10 *meter* in the 1 GHz to 2 GHz frequencies for cellular and PCS bands respectively [48]. A good measure of comparing these results is based on mean difference between the predicted and measured values and standard deviation from the mean.

We divided these point-to-point test cases into two categories. The first case is defined as line-of-sight (LOS) where there is at least one direct path between the transmitter and

the receiver location. This does not preclude any other types of paths such as reflected or diffracted paths from the transmitter to the same receiver location. The other case is defined as non-line-of-sight (NLOS). This is identified such that there is no direct path between the transmitter and the receiver but there may be many other types of paths.

7.1.2 Line-of-sight comparison

The test case is inside a medium sized building, a Lucent Technologies facility at Crawford Hill, NJ. The transmitter is located approximately eight feet above the floor on one side of a long corridor. For comparison purposes the receiver locations are assumed to be all along the corridor at approximately six feet above the floor. Notice that there can be several paths that reach a given receiver location based on transmissions through walls and reflections off walls, the roof, and the floor. Figure 7.1 shows a point-to-point comparison between the measured and predicted values at hundreds of locations along the corridor.

7.1.3 Non-line-of-sight comparison

We used a large cluttered urban environment like lower Manhattan as a test case for the non-light-of-sight comparison. The transmitter is located at approximately thirty feet above ground. The receivers are located at a height of six feet above ground, a typical height of a car antenna or a person walking on a street. The graph compares the measurements with the prediction based on the ray-beam tracing algorithms on a side street two blocks away from the street where the transmitter is located. This ensures

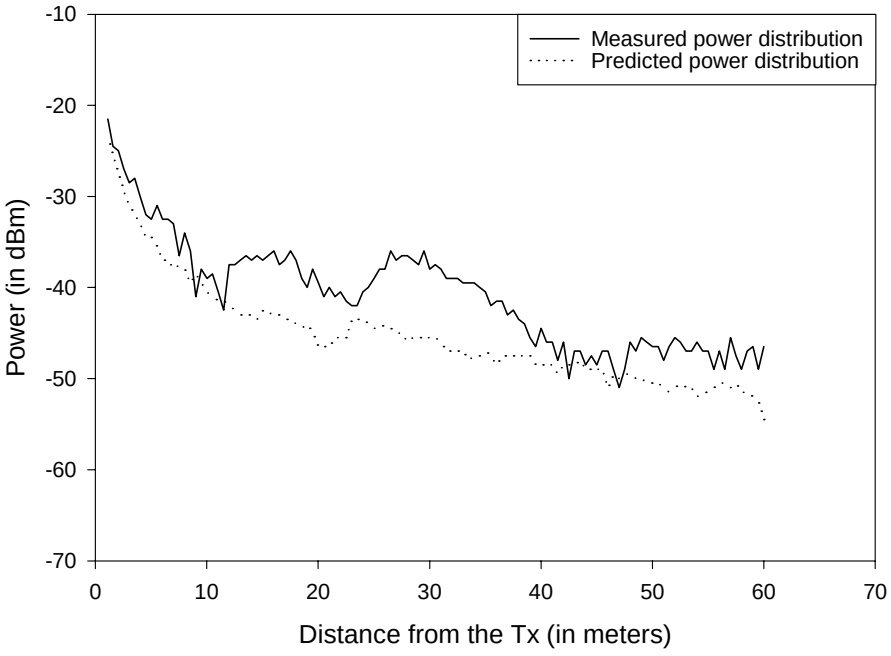

Figure 7.1: Comparison between measured and predicted power in line-of-sight from the transmitter inside a large building.

that there is no line-of-sight path between the transmitter and any of the receivers. As in the case of line-of-sight simulations, the comparison is point-to-point where a bin size is approximately five square meters. Figure 7.2 shows a point-to-point comparison between the measured and predicted values at hundreds of locations along the back street.

7.1.4 Environment-wide statistical comparison

To evaluate our algorithms for an entire environment we used two statistical measures. The first is a mean difference between the measured and predicted power amplitude for all receiver locations in the environment and the other is the standard deviation from the mean. The mean difference is given by

$$MD = \frac{\sum_i^N (X_{meas_i} - X_{predicted_i})}{N} \tag{7.1}$$

where X_{meas} and $X_{predicted}$ are the measured and predicted signal strength values respectively at a sampled location, and N is number of sample points. The standard deviation is given by

$$SD = \sqrt{\overline{X^2} - \overline{X}^2} \tag{7.2}$$

where X is the difference between the measured and predicted values, $\overline{X^2}$ denotes the mean of the squares of the values, and $\overline{X}^2$ denotes the square of the mean of the various values. Both these measures are given in dB. Table 7.1 shows a comparison for a number of environments based on the above statistical criterion.

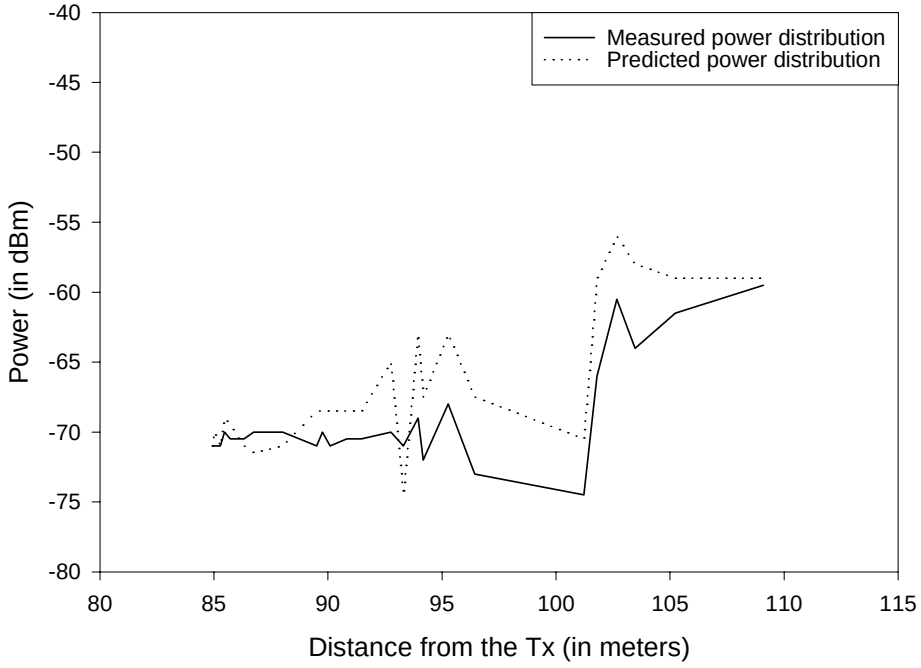

Figure 7.2: The graph shows a comparison between measured and predicted power on a side street (non-line-of-sight) from a transmitter (Tx) placed in lower Manhattan. The non-line-of-sight street is running parallel to the street where the transmitter is placed and shows a case of propagation around the block. The sudden increase in power beyond the 100 meter mark is because that area is at an intersection of a cross road and as a result received power from additional paths. One will also notice an anomaly between the predicted and measured signal strength at around 93 meters in terms of a sudden drop in the predicted value as compared to the measured value. Although, we do not have an explanation for this drop in power in this particular case, we have conjectured based on our experience with other input models and their analyses. Predictions by such a system as ours are only as accurate as the input model. If there are significant differences between the geometric input model used for predictions and the actual environment in which the measurements were taken then we may notice such anomalies. In this case it is possible that the geometric model of lower Manhattan we used may have some geometric entities which were not present when the measurements were taken.

Type of geometric environment	Absolute mean (in dB)	Std. deviation (in dB)
Outdoor – lower Manhattan	1.90	8.25
Inside bldg. – Crawford Hill	1.50	6.20
Inside bldg. – Middletown	0.40	5.10

Table 7.1: The table shows the statistical correlation between the measured and predicted values for different types of environments. The results are computed based on a point-to-point comparison over the entire environment.

7.2 Coverage maps from the *Propagate* system

The ray-beam tracing algorithm records the total received power for a each unique receiver location from a given transmitter location. This is generated as a coverage map. If there are more than one receiver locations for a given latitude-longitude coordinate, that is, power has to be recorded at more than one height (for example, inside buildings) then multiple coverage maps corresponding to each unique height are generated for the same transmitter location. We used *Propagate* to compute power distribution in several different environments. Figure 7.3 shows an aerial view of the propagation distribution inside a large office building. The transmitter is placed approximately eight feet above the floor in one of the corridors and power is sampled approximately six feet above the floor all across the building. Similarly, Figure 7.4 shows the coverage of power from a transmitter in lower Manhattan in the wall street area, which is an example of an urban environment. An example of coverage in a terrain-based environment is shown in Figure 7.7 and Figure 7.8. As mentioned in Section 6.4.1, these coverage maps can be

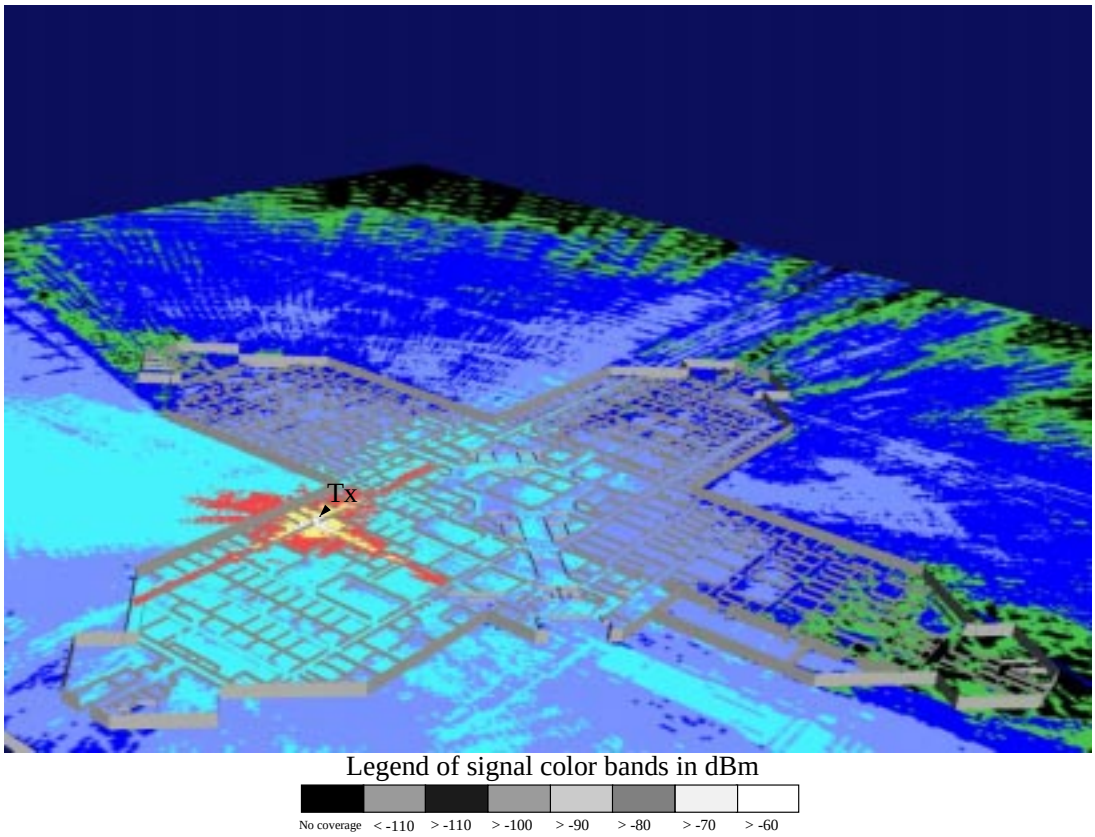

Figure 7.3: The image shows an aerial view of the complete coverage map from a transmitter location marked as 'Tx' with received power color-banded for easy visual identification. The geometric environment is a large modern office building.

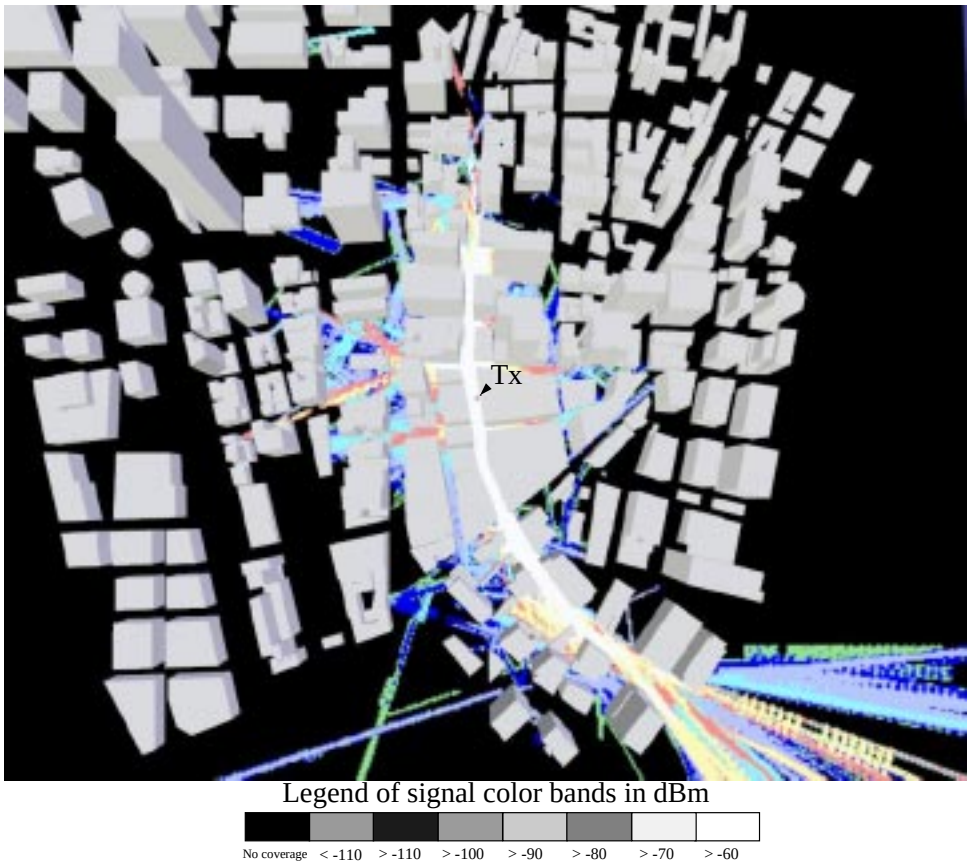

Figure 7.4: The image shows an aerial view of lower Manhattan with predicted coverage from a transmitter labeled as 'Tx'. The signal strength has been color-banded for easier visualization

queried by a user during interactive navigation of an environment.

7.3 Comparison of run times for different types of antennas

In Section 4.1, we had mentioned that one of the advantages of a broadcast mode of propagation is that it can handle complex antenna gain patterns and is able to derive runtime efficiency for antennas that have narrow horizontal or vertical beam widths. Table 7.2 shows a comparison of run times for different types of antenna gain patterns based on the proposed algorithms. One will notice from the table that depending on the horizontal and vertical beam width of an antenna, the run times could vary dramatically.

Tx: antenna characteristics		Number of receivers	Run time (in sec)
Horizontal beam width	Vertical beam width		
360° with constant (10dB) gain	5°	182x130x4	7767.8
360° with variable gain	5°	182x130x4	4618.3
120°	14°	182x130x4	4411.3
65°	7°	182x130x4	2418.5

Table 7.2: The table shows run times for different types of antennas. The antennas differ in their gain patterns as well as horizontal and vertical beam widths. Here, a beam width of an antenna is defined as the width of the beam beyond which its gain falls below half its peak value, that is, by more than $3\,dB$.

7.4 Sub-linear time complexity

As discussed in Section 4.2, one of the motivations of developing an algorithm based on the broadcast mode of propagation was its runtime efficiency with respect to the number of receiver locations. In this section we empirically show that our ray-beam tracing algorithm has sub-linear time complexity in terms of the number of receiver locations. One of the scenarios in which this efficiency gets effectively demonstrated is inside large buildings with multiple floors. This is so because radio waves (unlike rays in the visual spectrum) do travel through ceilings, floors, and other opaque surfaces such as walls to either contribute energy at another floor or cause interference at another floor. For this reason power distribution from a given source has to be sampled at several floors which increases the number of receivers multiple fold. It is important to observe that all receivers on a particular floor have the same height. This can easily be modeled by our algorithm by representing all receivers at the same height as a single horizontal plane.

To evaluate the performance of our algorithm in such a scenario we used a large five-floor building[1]. This multi-floor environment is shown in Figure 7.5. A comparison of the run times between the ray-beam tracing algorithm and an image based algorithm that is linear in the number of receiver locations for this large multi-floor building is shown in Table 7.3. As one will notice from the normalized run times given in this table, the performance of ray-beam tracing algorithm has sub-linear time complexity in terms

[1]This geometric database of the Computer Science building, University of California, Berkeley was provided by Thomas Funkhouser working at the time at Bell Labs, Lucent Technologies Inc.

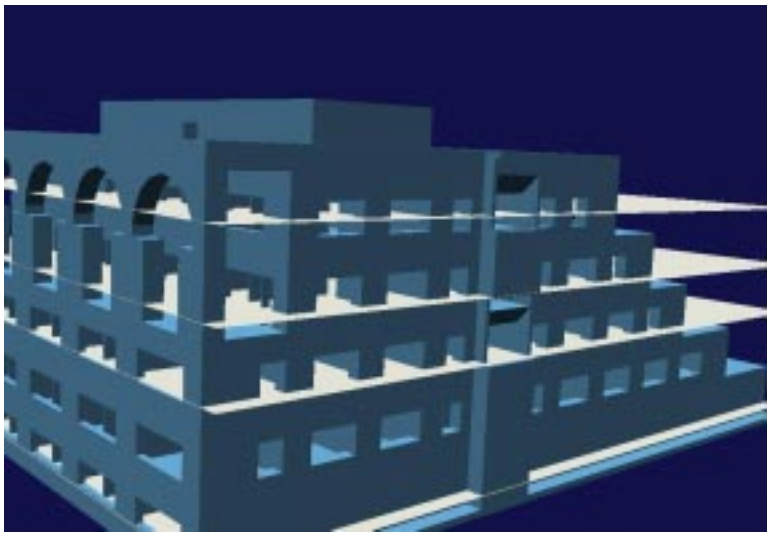

Figure 7.5: The image shows a large multi-floor building. For our simulations, a transmitter was placed on the third floor and power sampled on the first, third, fourth, and the fifth floors.

of the number of receiver locations as compared to the linear increase in time for an image based algorithm. The algorithm has sub-linear time complexity because it is not

Type of geometric environment	Number of receiver locations	Run time (normalized)	
		Linear-time image based algorithm	Broadcast algorithm
a five floor UC Berkeley, CS building	182x130x1	1.0	1.0
	182x130x2	2.0	1.07
	182x130x3	3.0	1.10
	182x130x4	4.0	1.13

Table 7.3: This table shows a runtime comparison between a standard image based algorithm and our algorithm based on the broadcast mode of propagation. The comparison is based on an increase in the number of heights at which power is being sampled in a large multiple-floor building. Each height is defined as sample plane having the same number of receiver locations. For easy comparison the run times have been normalized.

directly dependent on the number of receiver locations. The signal is propagated into the environment from a source without any knowledge of the exact location or the number of receivers. Typically, in cluttered environments the power level in a ray-beam attenuates below a desired threshold long before a particular receiver is encountered. and only if a ray-beam intersects a polygon (plane) representing a set of receiver locations does a ray-beam record the current power level for these receivers. Although it is important to note that the algorithm does take the distance between successive receiver locations into account before recording the power levels for each receiver.

The time complexity of the ray-beam tracing algorithm can also be measured as a function of the distance between the receiver locations. The runtime of the algorithm

increases sub-linearly with respect to the increase in the number of receivers for the same environment. This is because the algorithm is able to exploit spatial coherence of a beam as described in the next section.

7.4.1 Exploiting spatial coherence of a ray-beam

Most of the early attempts at proposing algorithms such as cone tracing, beam tracing, or pencil tracing [1, 26, 56] were explicitly designed to exploit the concept of spatial coherence and as a result the computational efficiency they provide. In our case, although the motivation was not to explicitly devise an algorithm to exploit spatial coherence, the ray-beam tracing approach addresses the issue of coherence quite effectively. When a ray-beam intersects a sample plane (a polygonal plane that geometrically represents the receiver locations) and projects upon a large area, the ray-beam in effect is exploiting coherence to contribute energy to a number of receiver locations simultaneously. This is possible, since each ray-beam maintains power-density per unit square area rather than the actual power in a beam. Figure 7.6 depicts this concept of simultaneously contributing to a number of receiver locations.

We evaluated the ray-beam tracing algorithm both by varying only the number of receiver locations and by varying the number of receiver locations along with a corresponding change in the ray-beam subdivision rate. The increase in the runtime is marginal as compared to the increase in the number of receiver locations as is shown in Table 7.4.

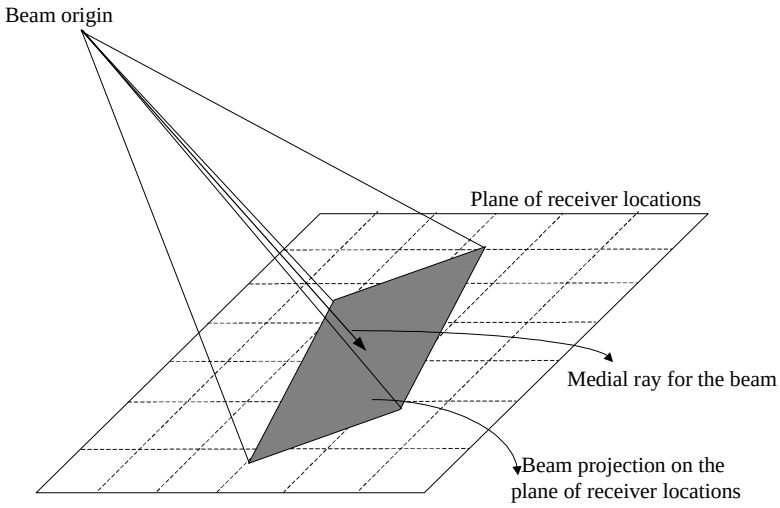

Figure 7.6: Projection of a ray-beam on a plane of receiver locations. The beam intersection with the plane is determined by the beams medial ray. Due to the spatial coherence of the ray-beam all receiver locations covered by the projection receive the same power density per unit area.

Type of geometric environment	Number of receiver locations	Relationship of receiver span w/wavelength	Run time (in sec) varying	
			receiver span only	receiver span w/change in ray-beam subdivision rate
a five floor UC Berkeley, CS building	5,915	4λ	3953.4	3757.2
	23,660	2λ	4141.8	4117.0
	94,640	1λ	4925.6	5236.6
	378,560	0.5λ	8036.4	7670.7

Table 7.4: The table shows a comparison of run times with a change in the number of receiver locations and the rate of ray-beam subdivision.

7.5 RF engineering applications

Several RF engineering applications can be built around a tool that efficiently computes power distribution in an environment from a given transmitter location with specific antenna properties. One of these applications is for the study of better antenna systems. Empirical studies have shown that antenna diversity can be used to counter small-scale fading effects observed in multipath environments. It is also an important technique to obtain greater antenna efficiency which can lead to higher capacity in a wireless network. Antenna diversity can be in the form of spatial diversity, polarization diversity, time diversity, and angular diversity. Some of these diversity schemes can be simulated using our system.

7.5.1 Delay spread computation

One of the techniques in which time diversity can be used is by computing delay spread experienced by signals in an environment. In a multipath environment two signals reaching a sample point from the same source may have different path lengths, which results in different arrival times. In an analog system, this time delay is not desirable. But for a digital system such as code division multiple access (CDMA), this delay (within a specified limit) can be used to an advantage by combining these signals.

Our system facilitates this computation quite easily by recording individual signals received at a receiver location. Multiple paths are recorded by a simple enhancement to the data structure that either records the path length along with the received signal

strength or indexes the received signal strength based on the time delay of the received signal. The mean time delay, $\overline{\tau}$ is computed as

$$\overline{\tau} = \frac{\sum_k a_k T_k}{\sum_k a_k} \tag{7.3}$$

where a_k is the amplitude of the k^{th} signal and T_k is the relative time delay of the k^{th} signal from the first signal received at the location. The delay spread can then be computed as

$$\sigma_\tau = \sqrt{\overline{\tau^2} - \overline{\tau}^2} \tag{7.4}$$

where $\overline{\tau^2}$ is given by the following equation.

$$\overline{\tau^2} = \frac{\sum_k a_k T_k^2}{\sum_k a_k} \tag{7.5}$$

7.5.2 Angular spread computation

In some cluttered environments antennas do not receive any signal from certain angular directions. If this can be predicted with a certain level of confidence then antenna systems could be built and deployed with specific beam patterns having an angular spread that are tailored to specific environments. That is, the gain pattern of an antenna could be altered such that there is more gain in some directions than others to match the needs of an environment.

Angular diversity is mostly used at the base station (both for a transmitter as well as for a receiver) instead of at the mobile because of the cost of providing larger antennas on a mobile phone. To simulate angular diversity for a receiver at the base station the signal

has to be transmitted from the receiver locations. This can be extremely inefficient. In such scenarios we use the principle of reciprocity to redefine the problem.

The principle of reciprocity states that if a transmitting antenna and a receiving antenna interchange their location without changing any other antenna parameters such as the transmitted power and the antenna gain pattern then the received signal strength remains the same at the receiver. The result can directly be derived from the Maxwell's equations in electromagnetic theory. This principle is very useful to model angular diversity.

To compute the angular spread for an environment, the idea is to model a transmitter at the base station as a directional antenna with a predefined beam width. At the same time each mobile location is modeled as an omni-directional isotropic antenna. Power is transmitted from each of these directional antennas for all 360°, and coverage computed as a function of the angle at which the signal was propagated. Then a metric can determine the extent of the coverage from each of these angles. This can determine a good approximation of the beam width for an antenna as well as the preferred azimuthal angle at which power is to be transmitted for a given environment.

7.5.3 Spatial antenna diversity

Another antenna diversity scheme that can be modeled by the system is the spatial antenna diversity. In heavily built urban environments deep fades can occur at spatially close locations. To alleviate such problems, service providers typically install more than

one sectored antenna at the base station. These receivers can either be placed at a horizontal spatial separation or a vertical spatial separation with respect to each other. The idea is that if there is a narrow fade then at least one of these antennas would receive a strong ray even if the other receives a faded ray.

Instead of modeling these antennas as receivers and the mobiles as transmitters, because of the *principle of reciprocity* we can again model the receivers at the base station as transmitters and the mobiles as receivers. With this redefinition, the problem is the same as our standard propagation problem. A simulation is performed for each of the antennas (at spatially separated locations) at the base station. Since our system records all the time-delayed multipaths, a correlation between different multipaths received by the individual antennas can be computed.

7.5.4 Modeling co-channel and adjacent-channel interference

According to wave theory, an important wave effect is coherence, which measures the correlation between the phases of two beams. Interference in the classical sense is defined such that if there is a positive correlation between the phases then the interference is constructive else it is destructive. If two beams when superimposed are completely incoherent then the two intensities get added linearly (where intensity is defined as the mean squared amplitude). This was further confirmed by the famous two-slit experiment [5].

Co-channel[2] and adjacent-channel interference in wireless communication does not refer to interference in the classical sense of wave theory. We model co-channel interference

[2]Here channel refers to a predefined frequency at which the signal is being transmitted.

by computing the received signal strength at each receiver location from all transmitters in the environment. Carrier-to-interference at a receiver location with respect to a transmitter T_0 is defined as a ratio of the signal strength received from T_0 and the total power received from all other sources. The relationship is given by the equation (7.6)

$$\frac{C}{I} = \frac{T_0}{\sum_{i=1}^{m} T_i} \tag{7.6}$$

where m is the number of other transmitters in the environment, T_i is the received signal strength from the i^{th} source, and I is the total interference. It is worth noting that computationally there is no difference between co-channel and adjacent-channel interference. However, depending on the setup of a network of transmitters, specific transmitters may be transmitting on the same channel or adjacent channel [48]. This is taken into consideration to determine the type of interference being computed.

7.6 Network-wide load balancing

One of the applications for such a radio propagation prediction tool is in the management of a network of base stations for optimal capacity utilization. A typical base station in any geographic area can serve a certain number of users based on a few predefined parameters. On the other hand, traffic densities in any geographic region and at any time is neither uniform in nature nor can be predicted *a priori*. For example, traffic density can drastically change on specific days depending on arbitrary events, such as traffic accidents, a ball game, or a concert somewhere in town; or even during the course of a day depending on certain traffic patterns. No network can be optimized for all of

these eventualities. Therefore, methods have to be devised to mitigate user congestion scenarios, especially those that cause overloading in one part of the network while under utilizing another part of the network.

7.6.1 Network load balancing based on antenna downtilt/uptilt

One method of achieving this load balancing is by dynamically downtilting some antennas and simultaneously uptilting other adjacent antennas. An antenna downtilt implicitly reduces a particular base stations coverage area which correspondingly reduces the number of users that have to be served by the base station (assuming a uniform distribution of users within the area of influence). Similarly, an antenna uptilt implicitly increases the coverage area and the number of users it can potentially serve. Our system can be used to effectively model such a scenario.

We conducted simulations in a terrain based environment to simulate the effects of an antenna downtilt and its impact on the ground coverage. Figure 7.7 shows images of coverage when the antenna is pointed at the horizon, that is, 0° downtilt and has a downtilt angle of 5°. Figure 7.8 shows coverage with a downtilt angle of 10° with respect to the horizon. One can notice the reduction in the coverage area by observing shrinking power levels shown as color bands in the individual images. The irregular coverage is due to the terrain effects as well as the particular antenna gain pattern being used. We compute the new gain pattern of a downtilted antenna by using the original gain value at an angle and modifying it by the downtilt angle instead of constructing a completely

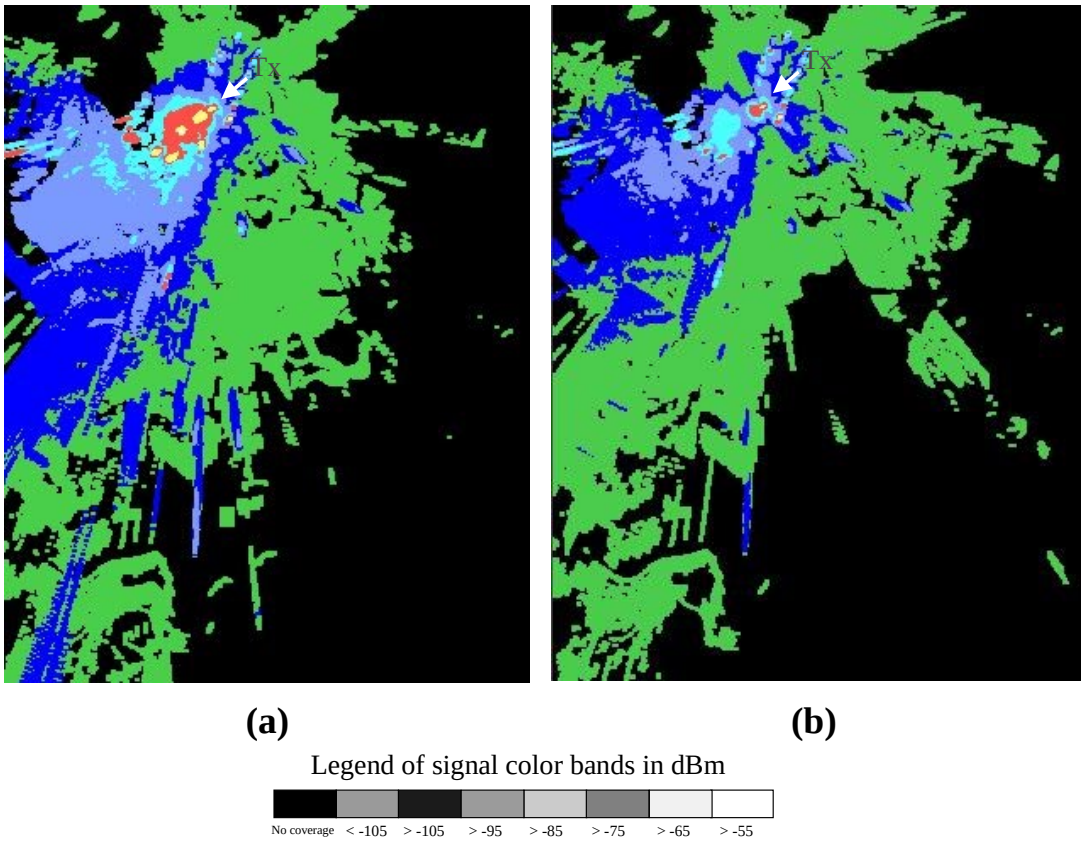

Figure 7.7: The image shows coverage in a terrain-based environment from a transmitter location marked as 'Tx'. The coverage is from an antenna that is downtilted to control intercell interference by reducing the area under coverage as well as the signal strength at a particular receiver location. (a) This shows the coverage when the transmitter is pointing at the horizon. (b) This shows the coverage with a downtilt angle of 5° with respect to the horizon. Notice the reduction in the power levels as compared to the corresponding locations for the other downtilt angle.

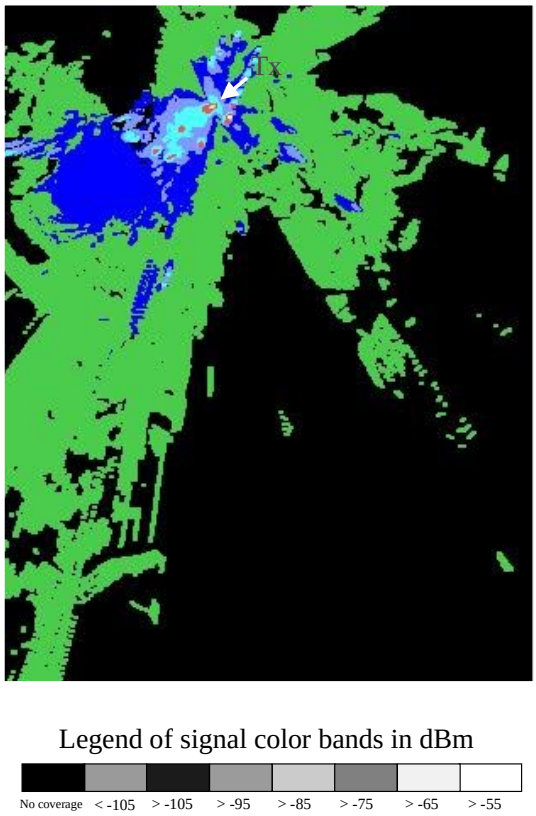

Figure 7.8: This image shows the coverage from a transmitter marked 'Tx' with a downtilt angle of 10° with respect to the horizon. All the other simulation parameters are the same as the previous figure with 0° and 5° downtilt. Notice the reduction in the power levels as compared to the corresponding locations for the other two downtilt angles along with the shrinkage in coverage area. The irregular coverage is due to the terrain effects as well as the antenna gain pattern being used.

new gain pattern. The former approach is more desirable and corresponds to an electrical downtilt whereas the latter approach corresponds to a physical downtilt of an antenna.

Chapter 8

Conclusions and Future Work

We conclude by again presenting the main contributions of this dissertation. We have proposed a unified approach to propagation for different types of site-specific geometric environments. The thesis proposes a broadcast mode of propagation and empirically shows that it is computationally more efficient for radio propagation applications than an image based approach. It also presents an adaptive ray-beam tracing algorithm, and an algorithm to dynamically compute edges in an environment for diffraction purposes.

A unified approach to radio propagation in all types of environments is extremely important if terrain effects have to be understood along with other obstructions such as buildings in urban and rural environments. The system and algorithms described in this thesis can act as a comprehensive platform for propagation studies. It can serve as a platform to study antenna diversity issues in site-specific environments as well. The adaptive ray-beam tracing algorithm can also be applied to simulating audio propagation

in three-dimensional environments.

We used the proposed algorithms to develop a prediction system. The prediction system was used to model propagation signal strength and other parameters in several different types of geometric environments. The prediction results were then compared with field measurements from these specific environments. The correlation between the predicted and measured results were good and within acceptable bounds.

Although the main focus of this thesis was to develop efficient algorithms for energy propagation in geometrically defined environments the usefulness of such a system largely depends on the easy availability of site-specific geometric models. Obtaining a geometric model of a site-specific environment and assigning the correct material properties to each of the surfaces or geometric entities is still a challenge. One of the hopes is that with time more buildings and urban areas will be available in some electronic form or the other such that they can easily be imported into such prediction systems for wider usage.

8.1 Future Work

There are a number of areas in which this work can be extended. One of the areas is to find a better ray-beam subdivision approach and investigate if a quadtree partitioning approach is more efficient than uniform subdivision. Though the current algorithms have addressed the issue of propagation for certain bands of frequencies a more thorough investigation is required for higher frequencies where small geometric entities such as leaves play an important role in attenuating the signal significantly to necessitate a detailed

modeling of vegetation.

Many other useful modeling problems can be defined on top of a system that predicts radio propagation in a site-specific environment. One such problem is the simulation of power control in a CDMA system. Currently it is extremely expensive to conduct any field trial that involves a network consisting of a large number of base stations and many mobiles. Similarly, understanding mobile hand-off (as a mobile moves from one cell to another) in a site-specific environment is an important problem in the optimal utilization of a network of base stations. A system that pre-computes propagation distribution from a set of transmitters could be a good starting point in modeling such problems.

Appendix A

Importing building databases

These days most of the modern buildings are first designed using some type of an architectural software package before the actual construction begins. This helps not only in better designs but future maintenance of the building as well. AutoCAD is a popular architectural design and drawing package. The software package has an ASCII file format called *data interchange format* (dxf) [40], to output drawing files. The format is a *de facto* industry standard for architectural and many other types of drawings. We found that most of the architectural floor plans of buildings are available in the dxf format either because they were designed using AutoCAD or a package that supported the dxf format. One of the simplest method of acquiring this large pool of building floor plans was to convert these files available in dxf format into our internal binary space partitioning tree (bspt) format. We used a dxf parser[1] that reads these ASCII files to generate an

[1]The initial version of this parser was written by Michell Potmesil and later modified by the author at Bell Labs to suit the needs of extracting useful entities from the dxf files.

intermediate polygonal representation, which is then converted into the bspt file format.

However, these dxf files have a lot of unnecessary information, which does not always correspond to their physical representation in the building. For example, short line segments stacked together represent stairs; crossed line segments within a rectangle represent elevator shafts and building columns; some arcs represent door sweeps; plumbing fixtures are represented by simple line drawings, and so on. It is interesting to note that most of these symbolic representations are merely used because of the convenience for the draftsman or sometimes are a standard convention in the drafting world. Since both walls (actual geometric entities) as well as these symbolic representations use lines and arcs, it is hard to differentiate between them automatically. Thus a dumb automated conversion does not work. Some heuristics and some manual pre-processing is necessary before these drawings can be used for modeling purposes. Finally, the problem is that these floor plans are two-dimensional, whereas, the actual geometry they represent is three-dimensional. That is, the floor plans do not have any information about the height of the individual entities. We used several heuristics to alleviate these problems.

First, we observed that some of these symbolic representations, text, and notations are usually placed in separate layers that are different from the actual geometry. Although it is difficult to identify these layers by name across all dxf files (in absence of any naming convention for these layers), it still is an effective first step to prune these entities based on the layers. We used color as an attribute that differentiates types of walls, for example, concrete, sheet-rock, or mere partitions between offices. Some heuristics based the size

of the line segments were also used to extract relevant information but no standard generalizations can be drawn. Several similar heuristics have been proposed in [29] to resolve some of these extraneous features in these databases.

Once a dxf file has been cleaned of these extraneous features, a file is read by the parser. Specific values are used to filter different attributes. Each polyline representing the walls is extruded in the z-direction (height). Then a floor and a ceiling is added to the extracted building database. In case a building has a false ceiling, another polygonal entity is added to match the database with the actual environment.

Bibliography

[1] Amanatides, John. Ray Tracing with Cones, *Computer Graphics,* Vol. 18, No. 3, July 1984, pp. 129–135.

[2] Appel, A. Some techniques for shading machine renderings of solids, *AFIPS 1968, Spring Joint Computer Conference,* Vol. 32, pp. 37–45.

[3] Arvo, James. Backward Ray Tracing, *Development in Ray tracing, Siggraph Course Notes,* Vol. 12, Aug. 1986.

[4] Balanis, C.A. *Advanced Engineering Electromagnetics,* John Wiley, 1989, New York.

[5] Born, M., and Wolf, E. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light,* Sixth edition, Pergamon Press, New York, 1986.

[6] Brekhovskikh, L.M., Godin, O.A. *Acoustics of Layered Media I, Plane and Quasi-Plane Wave,* Springer-Verlag, Berlin, Germany, 1990.

[7] Chattopadhayay and Fujimoto. Bi-directional ray tracing, *Computer Graphics: Proceedings of CG International 1987,* Springer Verlag, Tokyo, 1987, pp. 335–343.

[8] Chamberlin, K.C., and Luebbers, R.J. An evaluation of Longley-Rice and GTD propagation models, *IEEE Transactions Antenna Propagation,* Vol. AP-30, November 1982, pp. 1093–1098.

[9] Clark, James. Hierarchical Geometric Models for Visible Surface Algorithms, *Communications of the ACM*, Vol. 10, No. 10, 1976, pp. 547–554.

[10] Cohen, M.F., Greenberg, D.P. The Hemi-Cube: A Radiosity Solution for Complex Environments, *Computer Graphics,* Vol. 19, No. 3, July 1985, pp. 31–40.

[11] Cohen, M.F., Shenchang, E.C., Wallace, J.R., Greenberg, D.P. A Progressive Refinement Approach to Fast Radiosity Image Generation, *Computer Graphics,* Vol. 22, No. 3, August 1988, pp. 75–84.

[12] Conrady, A.E. *Applied Optics and Optical Design,* Part I, Oxford University Press, 1929. (reprinted by Dover Publications Inc. New York, 1957).

[13] Cook, Robert L., Porter, Thomas, and Carpenter, Loren. Distributed Ray Tracing, *Computer Graphics,* Vol. 18, No. 3, July 1984, pp. 129–135.

[14] Dadoun, Norm, Kirkpatrick, David G., and Walsh, John P. Hierarchical Approaches to Hidden Surface Intersection Testing, *Proceedings of Graphics Interface,* May 1982, pp. 40–56.

[15] Ebert D. et al. *Texturing and Modeling: A Procedural Approach,* Second Edition; AP Professional; Cambridge 1998, chapter entitled: Noise, Hypertexture, Antialiasing, and Gestures.

[16] Edwards, R., and Durkin, J. Computer Prediction of Service Area for VHF Mobile Radio Networks, *Proceedings of the IEE,* Vol. 116, No. 9, pp. 1493–1500, 1969.

[17] Elber, Gershon. Low Cost Illumination using an Approximation of Light Wavefronts, *Computer Graphics,* Vol. 28, No. 3, SIGGRAPH 1994, pp. 335–342.

[18] Feisullin, F., Naylor, B.F., Rajkumar, A., and Rogers, L. Simulation of RF Propagation in Large Scale Indoor and Outdoor Environments, *AT&T Workshop on Wireless Communications and Mobile Computing,* AT&T Bell Labs, Holmdel, NJ, Oct. 1994.

[19] Felsen, L.B., and Marcuvitz, N. *Radiation and Scattering of Waves,* Prentice Hall, New Jersey, 1973.

[20] Fortune, Steven J., Gay, David M., Kernighan, Brian W., Landron, O., Valenzuela, R. A., and Wright, M. H. WISE Design of Indoor Wireless Systems: Practical Computation and Optimization, *IEEE Computational Science and Engineering,* Vol. 2, No. 1, Spring 1995, pp. 58–68.

[21] Fuchs, Henry, Kedem, Zvi M., and Naylor, Bruce F. On Visible Surface Generation by A Priori Tree Structure, *Computer Graphics,* Vol. 14, No. 3, SIGGRAPH 1980, pp. 124–133.

[22] Funkhouser, T., Carlbom, I., Elko, G., Pingali, G., Sondhi, M., and West, J. A Beam Tracing Approach to Acoustic Modeling for Interactive Virtual Environments, *Computer Graphics,* Vol. 32, No. 3, SIGGRAPH 1998, Orlando, FL, July, 1998, pp. 21–32.

[23] Goral, C.M., Torrance, K.E., Greenberg, D.P., and Battaile, B. Modeling the Interaction of Light Between Diffuse Surfaces, *Computer Graphics,* Vol. 18, No. 3, SIGGRAPH 1984, pp. 213–222.

[24] Glassner, A. ed. *An Introduction to Ray Tracing,* Academic Press, London, 1989.

[25] He, X.D., Torrance, K.E., Sillion, F.X., and Greenberg, D.P. A comprehensive physical model for light reflection, *Computer Graphics,* Vol. 25, No. 4, July 1991, pp. 175–186.

[26] Hecbert, Paul S., and Hanrahan, Pat. Beam Tracing Polygonal Objects, *Computer Graphics,* Vol. 18, No. 3, July 1984, pp. 119–127.

[27] Kajiya, James T. The Rendering Equation, *Computer Graphics,* Vol. 20, No. 3, August 1986, pp. 143–150.

[28] Keller, J.B. Diffraction by an Aperture, *Journal of Applied Physics,* Vol. 28, No. 4, April 1957, pp. 426–444.

[29] Kernighan, Brian W., and Van Wyk, Christopher J. Extracting geometrical infor-

mation from architectural drawings, *Proceedings of the Workshop on Applied Computational Geometry,* May 1996, pp. 82–87.

[30] Knott, Eugene F., Shaeffer, M.F., Michael, T. *Radar Cross Section,* 2nd Edition, Artech House, Norwood, MA, 1993.

[31] Kouyoumjian, R.G., and Pathak, P.H. A Uniform Geometrical Theory of Diffraction for an edge in a perfectly conducting surface, *Proceedings of IEEE,* Nov. 1974, Vol. 62, No. 11, pp. 1448–1461.

[32] Kreuzgruber, P., Unterberger, P., and Gahleitner, R. A Ray splitting model for indoor radio propagation associated with complex geometries, *Proceedings of the 43rd IEEE Vehicular Technology Conference,* May 1993, pp. 227–230.

[33] Kreuzgruber, P., Brundl, T., Kuran, W., and Gahleitner, R. Prediction of Indoor Radio Propagation with the Ray Splitting Model Including Edge Diffraction and Rough Surfaces, *Proceedings of the 44th IEEE Vehicular Technology Conference,* June 1994, pp. 878–882.

[34] Lee, William C.Y. *Mobile Communication Design Fundamentals,* John Wiley & Sons, 1993.

[35] Lewis, E. E., and Miller, Jr., W.F. *Computational Methods of Neutron Transport,* John Wiley & Sons, 1984.

[36] Luebbers, R.J. Finite Conductivity Uniform GTD Versus Knife Edge Diffraction in

Prediction of Propagation Path Loss, *IEEE Transactions on Antenna and Propagation,* Vol. AP–32, No. 1, Jan. 1984, pp. 70–76.

[37] Luebbers, R.J. Propagation Prediction for Hilly Terrain Using GTD Wedge Diffraction, *IEEE Transactions on Antenna and Propagation,* Vol. AP–32, No. 9, Sept. 1984, pp. 951–955.

[38] Moravec, H.P. 3D Graphics and the Wave Theory, *Computer Graphics,* Vol. 15, No. 3, SIGGRAPH 1981, pp. 289–296.

[39] Murali, T.M., and Funkhouser, Thomas A. Consistent Solid and Boundary Representations from Arbitrary Polygonal Data, *1997 Symposium on Interactive 3D Graphics,* Providence, RI, pp. 155–162.

[40] Murray, J.D., and vanRyper, W. *Encyclopedia of Graphics File Formats,* O'Reilly & Associates, 1996.

[41] Naylor, B.F. Binary Space Partitioning Trees as an Alternative Representation of Polytopes, *CAD,* Vol. 22, No. 4, May 1990, pp. 250–253.

[42] Naylor, B.F. Constructing Good Partitioning Trees, *Graphics Interface*, Toronto Canada, May 1993, pp. 181–191.

[43] Nelder, J.A., and Mead, R. A Simplex Method for Function Minimization, *Computer Journal,* Vol. 7, 1965, pp. 308–313.

[44] Nishita, T., and Nakamae, E. Continuous Tone Representation of Three-Dimensional

Objects Taking Account of Shadows and Interreflections, *Computer Graphics,* Vol. 19, No. 3, SIGGRAPH 1985, pp. 23–30.

[45] Pharr, M., and Hanrahan, P. Geometry caching for ray tracing displacement maps, *Eurographics Workshop on Rendering,* 1996, pp. 31–40.

[46] Pharr, M., Kolb, C., Gershbein, R., and Hanrahan, P. Rendering Complex Scenes with Memory-Coherent Ray Tracing, *Computer Graphics,* Vol. 31, No. 3, SIGGRAPH 1997, pp. 101–108.

[47] Rajkumar, A., Naylor, B.F., Feisullin, F., and Rogers, L. Predicting RF Coverage in Large Environments using Ray-Beam Tracing and Partitioning Tree Represented Geometry, *Wireless Networks,* Vol. 2, No. 2, 1996, pp. 143–154.

[48] Rappaport, T.S. *Wireless Communications Principles & Practice,* Prentice Hall Inc., 1996.

[49] Reif, J., Tygar, D., and Yoshida, A. The Computability and Complexity of Optical Beam Tracing, *Proc. 31st Annual Symposium on Foundations of Computer Science,* Vol. 1, 1990, pp. 106–114.

[50] Reif, J., Tygar, D., and Yoshida, A. Computability and Complexity of Ray Tracing, *Discrete and Computational Geometry,* Vol. 11, No. 3, April 1994, pp. 265–288.

[51] Rubin, S.M., and Whitted, T. A three-dimensional representation for fast rendering

of complex schemes, *Computer Graphics,* Vol. 14, No. 3, SIGGRAPH 1980, pp. 110–116.

[52] Schaubach, K.R., Davis IV, N.J., and Rappaport, T.S. A Ray Tracing Method for Predicting Path Loss and Delay Spread in Microcellular Environments, *42nd IEEE Vehicular Technology Conference,* Denver, Co, May 1992, pp. 932–935.

[53] Seidel, Scott Y., and Rappaport, Theodore S. A Ray Tracing Technique to Predict Path Loss and Delay Spread Inside Buildings, Virginia Polytechnic Institute and State University, Blacksburg, VA.

[54] Shade, J., Lischinski, D., Salesin, D., DeRose, T., and Snyder, J. Hierarchical image caching for accelerated walkthroughs of complex environments, *Computer Graphics,* Vol. 30, No. 3, SIGGRAPH 1996, pp. 75–82.

[55] Shao, M.Z., Peng, Q.S., Liang, Y.D. A New Radiosity Approach by Procedural Refinements for Realistic Image Synthesis, *Computer Graphics,* Vol. 22, No. 4, SIGGRAPH 1988, pp. 93–101.

[56] Shinya, M., Takahashi, T., and Naito, S. Principles and Applications of Pencil Tracing, *Computer Graphics,* Vol. 21, No. 4, SIGGRAPH 1987, pp. 45–54.

[57] Siegel, R., and Howell, J.R. *Thermal Radiation Heat Transfer,* 3rd edition, Hemisphere Publishing Corp., Washington D.C., 1992.

[58] Sillion, F., Puech, C. A General Two-Pass Method Integrating Specular and Diffuse Reflection, *Computer Graphics,* Vol. 23, No. 3, SIGGRAPH 1989, pp. 335–344.

[59] Snyder, John M. and Barr, Alan H. Ray Tracing Complex Models Containing Surface Tessellations, *Computer Graphics,* Vol. 21, No. 4, SIGGRAPH 1987, pp. 119–128.

[60] Sommerfeld, A. Lectures on Theoretical Physics, *Optics,* Vol. 4, Academic Press, New York, 1964.

[61] Teller, Seth., Fowler, Celeste., Funkhouser, Thomas., and Hanrahan, Pat. Partitioning and Ordering Large Radiosity Computations, *Computer Graphics*, Vol. 28, No. 3, SIGGRAPH 1994, pp. 443–450

[62] Thibault, W., and Naylor, B.F. Set Operations on Polyhedra Using Binary Space Partitioning Tress, *Computer Graphics,* Vol. 21, No. 4, SIGGRAPH 1987, pp. 153–162.

[63] Triolo, A. personnel communications.

[64] Valenzuela, R.A. Ray-Tracing Approach to Predicting Indoor Wireless Transmission, *Proceedings of the 43rd IEEE Vehicular Technology Conference,* May 1993, pp. 214–218.

[65] Vaughan, R.G. Polarization Diversity in Mobile Communications, *IEEE Transactions Vehicular Technology,* Vol. 39, No. 3, August 1990, pp. 177–186.

[66] Veach, E., and Guibas, Leonidas J. Metropolis Light Transport, *Computer Graphics,* Vol. 31, No. 3, SIGGRAPH 1997, pp. 65–76.

[67] Veach, E. *Robust Monte Carlo Methods for Light Transport Simulation,* Ph.D. thesis, Stanford University, 1998.

[68] Verbeck, C.P., and Greenberg, Donald P. A Comprehensive Light-Source Description for Computer Graphics, *IEEE Computer Graphics and Applications,* July 1984, pp. 66–75.

[69] Vinco, Erceg., Rustako, Jr., A.J., and Roman, R.S. Diffraction Around Corners and its Effect on the Microcell Coverage Area in Urban and Suburban Environments at 900 MHz, 2 GHz, and 6 GHz, *IEEE Transactions on Vehicular Technology,* Vol. 43, No. 3, August 1994, pp. 762–766.

[70] Wallace, J.R., Cohen, M.F., Greenberg, D.P. A two-pass solution to the rendering equation: a synthesis of ray-tracing and radiosity methods, *Computer Graphics,* Vol. 21, No. 4, SIGGRAPH 1987, pp. 311–320.

[71] Wallace, J.R., Elmquist, K.A., Haines, E.A. A Ray Tracing Algorithm for Progressive Radiosity, *Computer Graphics,* Vol. 23, No. 3, SIGGRAPH 1989, pp. 315–324.

[72] Whitted, T. An Improved Illumination Model for Shaded Display, *Communications of the ACM,* Vol. 23, No. 6, 1980, pp. 343–349.