Using Ray Tracing for Site-Specific Indoor Radio Signal Strength Analysis

by

Michael Nidd

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 1997

I hereby declare that I am the sole author of this thesis.

I authorize the University of Waterloo to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Waterloo to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

As wireless networks are installed at more locations, it will become more important that they be organised in an efficient layout. This requires prediction tools that use site-specific information to give better results than the traditional statistical models. While counting the number of walls between the transmitter and receiver can provide a rough approximation, more comprehensive techniques can make better predictions.

This report explores ray tracing as a method for predicting signal strength in an indoor wireless radio environment. It presents a dynamic "cone tracing" algorithm that offers a reasonable compromise between execution time and accuracy. This technique bypasses the duplication of effort present in similar previous attempts to analyze the signal strength in an entire room, instead of just at a single receiver location. The more complete view gives the planner seemingly continuous information about the region of interest, providing more information about the areas of poor coverage, and allowing for more informed decisions to be made in attempting corrections. This improved quality is achieved without the sacrificing response time.

Acknowledgements

Wow! I hardly know where to begin. It seems that every week I spent in this program introduced me to another helpful person. I enjoyed the work, but it would have been much less fun if it were not for the support of all my friends. This being the case, I would like to thank first all the people whom I have not mentioned below. I couldn't possibly list you all.

Although he had moved on before I had gotten very far in writing my thesis, I would like to thank Dr. Michael Coffin for his help in developing the topic, what some people call the hardest part of the whole thesis process, and also for his helpful comments on the final product. The chronology of supervisors leads next to Dr. Jay Black, without whose motivation I might still be looking at a blank screen. Effectively composing technical reports such that they are both informative and comprehensible is a difficult skill, and Jay's patience during my efforts at development in this area have had a permanent effect on how I approach technical writing (and reading). The third addition to my chain-o-supervisors was Dr. Stephen Mann. As this report started looking more like ray tracing, and less like network analysis, outside advice was required. Steve was always available to bounce ideas off, and do reality checks on my wilder notions. While it was sometimes frustrating to drop in and ask him what he thought of some algorithm that I'd been developing for a couple of days, only to have him take thirty seconds to come up with a case that shot holes in it, I shudder to contemplate the hours that I might have wasted without his help. Thanks also go out to Dr. Peter Forsyth, who read the final draft, and pointed to important reading that had escaped my literature search. Some other people who were not my supervisors, but who made this paper possible, have been Dr. Ian McGee, whose mathematical advice was irreplacable; Wendy Rush, who taught me how University administration works; and Jane Prime, who

shielded me from so much paperwork.

Money is another important part of finishing up a degree, and I am indebted to the generosity of Motorola Canada Limited for their support of the mobile computing project at Waterloo. My work was also funded by a scholarship from the Institute for Computer Research (ICR) of the University of Waterloo.

Contents

1 Introduction **1**

2 Electromagnetic Wave Characteristics **3**

2.1 Superposition (Interference) 3

2.2 The High Frequency Assumption 5

2.3 Reflection 8

2.4 Refraction (Transmission) 9

2.5 Scattering 10

2.6 Diffraction 10

3 Related Work **13**

3.1 Solving for Signal Strength 13

3.2 Statistical Predictions 15

3.3 Ray Tracing Techniques 16

3.4 Existing Ray Tracing Implementations 18

4 Practical Solutions and Optimisations **21**

4.1 Cone Tracing vs. Ray Tracing 21

4.2 Cone Emission 24

4.2.1 Adaptive Algorithm 26

4.2.2 Exact Algorithm 30

4.3 Representing Different Antennas 31

4.4 The Storage of Interim Values 32

4.5 Intersection Calculations 34

4.5.1 Intersecting rays with walls 34

4.5.2 Shadows . . . 36
4.5.3 Identifying the area of intersection . . . 36
4.5.4 Restrictions . . . 37
4.6 Diffuse Scattering . . . 38

5 Prediction Tool Design **41**
5.1 Basic Functionality . . . 42
5.1.1 Initial Cones . . . 42
5.1.2 Identifying affected walls . . . 43
5.1.3 Limiting Cone Size . . . 47
5.1.4 Ordering the walls . . . 49
5.1.5 Adding a cone to a wall . . . 50
5.2 Future Work . . . 53
5.2.1 Diffraction . . . 55
5.2.2 Missing Walls . . . 56
5.2.3 Distance Determination . . . 57
5.2.4 Incident Angles and Coefficients . . . 57

6 Performance Results **59**
6.1 Agreement With Theory . . . 59
6.2 Execution Time . . . 62
6.3 Profile Results . . . 65

7 Conclusions **67**

A Test Run Layouts **69**

B Derivation of Equation 4.2 **73**

Bibliography **75**

Glossary **79**

List of Figures

2.1 Adding waves in complex space 5
2.2 The frequency regimes 6
2.3 The daughter rays most commonly caused by an incident signal . . 9
2.4 Knife-edge diffraction 11

4.1 Point sampling of a big chess board 22
4.2 Two-dimensional view of the reception sphere 23
4.3 Several cones striking a single signal-intensity grid square 23
4.4 Cross-section of a cone 27
4.5 Expanding error bounds to three dimensions 28
4.6 Occlusion of background walls (shadows) 36

5.1 A cone striking interesting wall geometries 46
5.2 Cones intersecting planes at different angles 47
5.3 Two patterns for subdividing cones 48
5.4 Determining order for processing walls 49
5.5 Method for determining possible occlusion between two walls 51
5.6 Method for determining possible occlusion between two half planes 52
5.7 Cone addition 53
5.8 The shape of diffracted cones 55

6.1 Cone tracing vs. Lecours et al 60
6.2 Transmission and reflection coefficients vs. angle of incidence 62
6.3 Execution times for four rooms 63
6.4 How bounce limits affect results 64

Chapter 1

Introduction

Signal strength prediction pre-dates the existence of radio. In 1866 James Clerk Maxwell published the four fundamental equations of electromagnetic field theory, now known as "Maxwell's equations." These were largely based on the earlier work of Ampere, Gauss, and Faraday, but Maxwell's addition of a displacement term [Max66] is what earned him a place in history. This dates the first radio signal propagation research twenty two years before Hertz confirmed experimentally that radio waves (a.k.a. Hertzian waves) existed.

Since Maxwell also unified the theories of radio waves and light waves (electromagnetic radiation), there is a great deal of overlap between physical optics and radio signal propagation. Unfortunately, while this overlap has resulted in a well-defined theory for predicting signal propagation, the existence of theory does not always imply a solution to the practical problem. In this case, an exact solution would be intractable. The problem, therefore, is to explore which approximations are acceptable, and to find a fast method for their application.

With that in mind, this thesis presents cone tracing as a potential solution to

the problem of signal strength prediction, and addresses the following questions:

1. Does this general class of technique (cone tracing) constitute a reasonable method for signal strength prediction?

2. What can be done to improve implementations of these concepts?

To solve these problems, algorithms are presented that allow this cone tracing technique to be implemented and examined in a practical context. The validity of the technique is demonstrated through a sample implementation, and various optimization strategies are presented.

The technique proposed represents the signal as a collection of solid, non-overlapping cones radiated from a transmitter. It then overlays a horizontal grid onto a scene, and assembles information about signal quality on a square-by-square basis. Signal interactions with walls are also recorded by dividing the wall into discrete squares, providing information which can later be used to simulate scattered radiation. Reflection and transmission are handled by generating new cones with different apparent intensities.

Before going into detail about the specifics of this technique, Chapter 2 covers the fundamental physics of radio frequency (RF) signal propagation, and Chapter 3 describes previous research in the area. Chapters 4 and 5 discuss the theoretical underpinnings of this new technique and the design decisions made in the sample implementation. Chapter 6 then discusses what was learned from this implementation, leading to conclusions in Chapter 7.

Chapter 2

Electromagnetic Wave Characteristics

Understanding what electromagnetic radiation is and how it behaves will be essential to understanding the methods for predicting its behaviour. With this in mind, this chapter reviews the physical characteristics of radio frequency radiation. This discussion touches on three very different areas. The first is the principal of superposition and some notation to simplify its later application. The second introduces how signals in different frequency ranges interact with solid matter. The third gives detail about interactions in the optical region, as defined in Section 2.2.

2.1 Superposition (Interference)

When describing wave motion, it is convenient to use trigonometric functions. For instance, $E = A\sin\left(\frac{2\pi t}{T_0} + \epsilon_0\right)$ describes the instantaneous amplitude at one point on a wave with amplitude A and period T_0. The ϵ_0 allows the wave to be non-zero

at time $t = 0$. This provides a compact description that is easily interpreted, but it does not describe what happens when waves interfere.

The principal of superposition states that the amplitudes of interfering waves add. That means that if waves $E_1 = \sin x$ and $E_2 = \sin x$ interfere, the resulting energy can be described by $E_{total} = 2 \sin x$. Unfortunately, the general reduction of $A_1 \sin(x + B_1) + A_2 \sin(x + B_2)$ is very awkward.[1] A much simpler technique uses "Euler notation."

Euler notation provides a consistent interpretation of complex exponents as follows:

$$e^{i\theta} = \cos\theta + i \sin\theta,$$

which can be interpreted as $e^{i\theta}$ having a value which is angle θ around a unit circle in the complex plane. This is useful because a sinusoidal wave of any amplitude or phase can now be represented by a circular path in complex space. The instantaneous sum of two waves will be the vector sum of the corresponding complex coordinates.

If we assume that all signals have the same carrier frequency, as will be the case in predicting the footprint of a single antenna, then the angles between vectors representing signal components arriving at a given point will be constant. Therefore their sum will also have a constant radius and angular velocity. Since the angular velocity (frequency) is the same for all signals being considered, and the radius and instantaneous angle are given by the sum, this sum completely describes the resulting wave.

Because of this property the signal resulting from a new wave arriving at a

[1] $A_1 \sin(\omega t - B_1) + A_2 \sin(\omega t - B_2) = A \sin(\omega t - B)$
where $A^2 = A_1^2 + A_2^2 + 2A_1A_2 \cos(B_1 - B_2)$ and $\tan(B) = \frac{A_1 \sin(B_1) + A_2 \sin(B_2)}{A_1 \cos(B_1) + A_2 \cos(B_2)}$

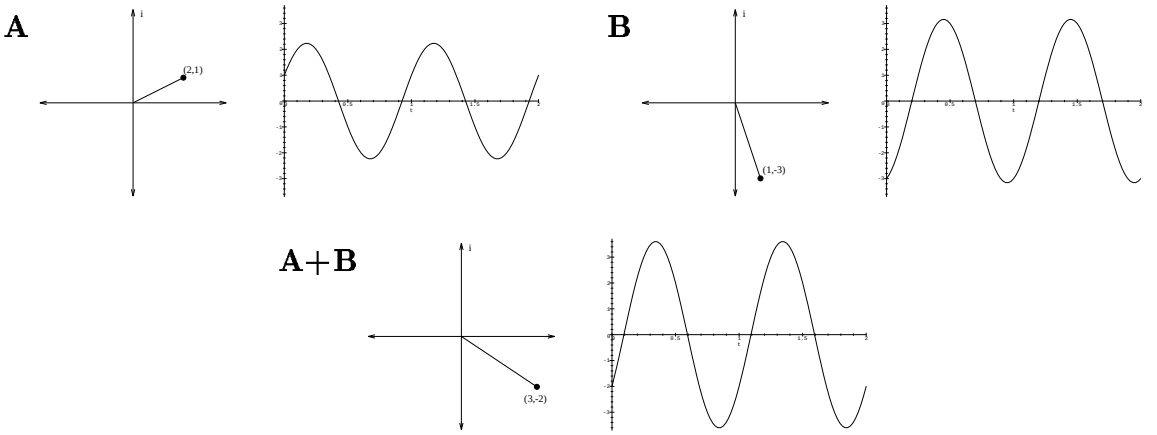


Figure 2.1: Adding Waves: For the waves A,B, and A+B, the complex value at t=0 is on the left, and the resulting wave is on the right.

point can be calculated with only two additions once the new signal is expressed in complex coordinates (Figure 2.1).

2.2 The High Frequency Assumption

The relation of angle of incidence to the amplitude of energy scattered, reflected, or otherwise returned from an object as a result of incident electromagnetic radiation is called the *radar cross section* of that object. Figure 2.2 compares the radar cross section σ (reflected field intensity) of a metallic sphere to the circumference. (σ is normalized to the geometric cross section (projected area) of the sphere, and circumference is in terms of wavelengths.) This figure shows that the radar cross section of an object is strongly dependent on the wavelength of the incident radiation, and can be categorised broadly into the following three regimes [KST85, p.53]:

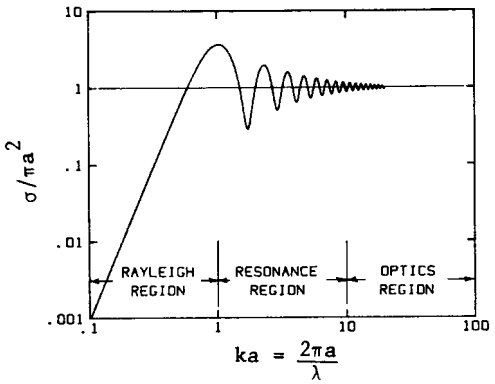


Figure 2.2: The frequency regimes

Rayleigh Region: (Low frequency region) Wavelength is long with respect to the object. The strength of the returned radiation grows quickly as the frequency increases.

Resonance Region: Wavelength is similar to the size of the object. The strength of the returned radiation oscillates as the frequency increases.

Optical Region: (High frequency region) Wavelength is short with respect to the object. The strength of the returned radiation is reasonably constant as the frequency increases.

While the evaluating the exact integral equations over the room would be a very difficult exercise, finite-difference and finite-element methods could be used to give good approximations of the results [PLL92]. Experimental results, however, still suggest that some 20 nodes would be required per wavelength. For 1 GHz carrier waves,[2] this would require that building floors be broken into cubes with a side-

[2]The 1 GHz region is currently a very popular frequency range for indoor radio communications (although some new research is also looking at the region near 20 GHz). In particular, this is the

length of about one centimetre. For an average building, this will be hundreds of millions of data points in the converging matrix. Clearly, a more memory-efficient alternative would be preferable.

By considering only the optical region, the radiation can be treated as rays (rather than as a field). This means that the computations of reflected directional intensity can be done discretely, instead of through evaluations over the entire field. Therefore, if we assume that the frequency is high, we can use ray tracing techniques to predict signal intensity. In this case, the rays have source and direction that change only during reflections. Also, the calculation is being done all at once, rather than in an attempt to reach a steady state. Because of these differences from the above-mentioned matrix technique, the final signal data needs only to be stored for areas of interest to the user. This means that it is no longer necessary to store the signal intensity for every discrete section of volume in the room to reach a solution. The resulting memory saving makes this a very useful restriction.

The expected demand for high data rates in indoor networks will require high-frequency carrier waves. Therefore, since the techniques proposed in this paper deal with indoor networks, I will focus on the prediction of high-frequency signal propagation [Fre91, LGBC93, THF92]. In addition to increasing the accuracy of treating the fields as collections of rays, the use of high-frequency transmissions also means that the signals will fade quickly. This means that signal propagation through walls and around corners is smaller than for the lower-frequency regions, decreasing the error associated with numerical techniques.

Assuming that all surfaces are large in comparison with the wavelength (by a factor of ten or more) allows surface interactions to be characterised in the optical

region being proposed for GSM, a system already popular in many parts of the world, offering 9600 bps digital channels [MP92].

region. This assumption is usually referred to as "the high frequency assumption." (Geometrical optics also requires that the principal radii of curvature for the surface be large with respect to the wavelength, but since the techniques discussed in this paper deal only with planes, this is not relevant.)

This assumption, that radio frequencies are high with respect to the sizes of the objects being considered, holds for the remainder of this thesis.

2.3 Reflection

Like light, radio signals reflect when they strike barriers. If the incoming signal is broken into components parallel to the surface and perpendicular to the surface, the reflected wave has the same parallel components, and the negative of the perpendicular component. In a two-dimensional world this can be viewed as the angle of incidence (angle between the surface and the incoming ray) equalling the angle of reflection (angle between the surface and the reflected ray), as shown in Figure 2.3. The only difference between these two views is that the second assumes a two-dimensional frame of reference parallel to the path of the ray.

The looming difficulty here is the principle of conservation of energy. It would be convenient to reflect the signal at full intensity, but the existence of other daughter signals implies that this would be incorrect. The ratio of incident energy to reflected energy is called the reflection coefficient. The value of this coefficient can be calculated exactly only if the material can be described by an exact mathematical model (either homogeneous, or a collection of smoothly varying strata [Jon79, §§8.5]), which is not the case for most building materials. The solution I chose uses experimentally determined functions for the reflection coefficients of the materials. Specialized equipment exists for determining these functions, or values

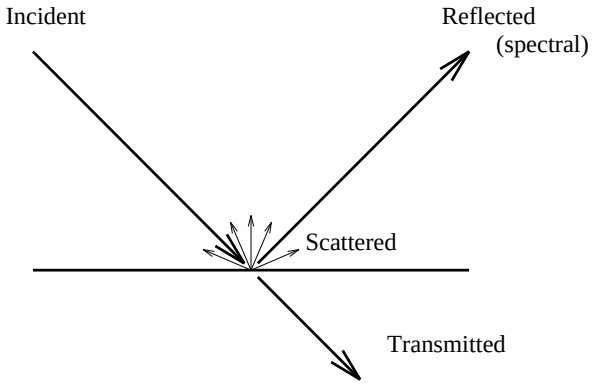


Figure 2.3: The daughter rays most commonly caused by an incident signal

found by researchers [Ale83, CMN83, CMN84] can be combined with generalized functions [HBD+92]. Theoretical calculation tells us that the reflection coefficient varies with the angle of incidence. Therefore, the functions used to approximate the exact coefficients should be capable of varying with this angle. (As discussed in Section 6.1, the effect of incident angle on reflection and transmission coefficients is not considered in my experimental implementation.)

2.4 Refraction (Transmission)

Most of the energy not reflected continues through the barrier medium. This transmitted signal changes direction in accordance with Snell's law, bending towards the normal as it enters a more dense medium, but resuming its previous direction of travel when it reenters the air. Waves inside walls are not relevant to this study, so the term "transmitted signal" refers here to a ray that has emerged from a wall as a result of an incident signal on the other side of that wall. Note that, since the walls in the test implementation (described later) are considered to be thin, a transmitted wave front will have exactly the same direction of travel and focal

point as the incident wave front.

Like reflected signals, refracted signals do not have the same energy as the corresponding incident signal. The refraction coefficient corresponds closely to the reflection coefficient, and has been approximated using the same technique.

2.5 Scattering

Reflection and refraction account for the dissipation of most of the incoming energy, but the barrier material is not a passive component of this interaction. Subjecting a conductive material to a changing electromagnetic field will induce a charge that alternates with the incoming field, and an alternating charge produces an electromagnetic field. The emission pattern of the resulting electromagnetic field depends only on the material itself, not the orientation of the incoming signal. Because its direction is unknown, the induced radiation is called "scattering."

2.6 Diffraction

To say that the angles of reflection and refraction are defined by the angle of incidence with respect to the surface normal is of very limited use when the surface normal is not well defined. At edges and corners of surfaces the surface normal is not defined in at least one dimension. In these cases, the signal behaves as if the normal were in all possible directions at the point of discontinuity; hence an infinitesimal part of an incoming signal (looking like a line) striking an edge will give rise to a conic shell (Figure 2.4).

Understanding superposition, the high frequency assumption, and RF signal

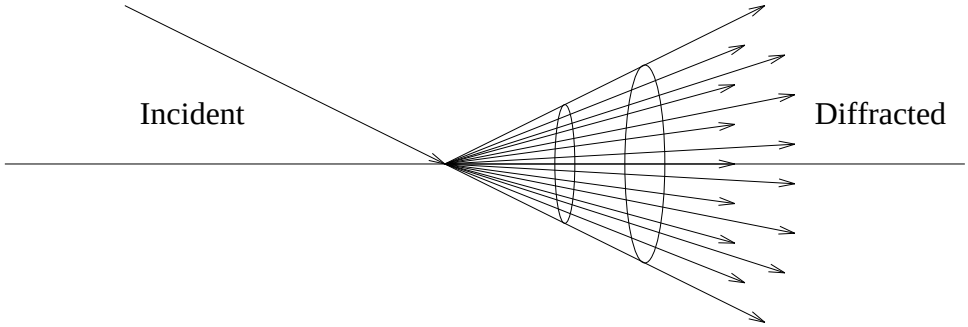


Figure 2.4: Knife-edge diffraction

behaviour leads to discussion about existing techniques for the prediction of this behaviour. This is the subject of Chapter 3.

Chapter 3

Related Work

The interactions presented in Chapter 2 are deceptively simple. As these inelaborate building blocks are stacked, the final result is complex (probably an intractable problem in the general case), but in its complexity it is also interesting. This fact, combined with the increasing use of radio in industry, has attracted a great deal of research into predicting signal strength. In this chapter I give introductions to both RF signal propagation research and the idea of ray tracing. Having introduced both topics, I explain how the problem of ray tracing for computer graphics has relevance to that of site-specific signal strength prediction. This will be summarized by a brief presentation of the attempts that have already been made to exploit this similarity.

3.1 Solving for Signal Strength

The problem physical optics leaves unsolved is that of calculating the specific characteristics of the propagated waves; even a small error can mean the difference

between constructive and destructive interference. Today this is a challenging problem, but in the early days of radio signal research it was not even worth considering, given the computational resources available.

The easiest way to avoid the complexity resulting from obstacles is to ignore them. Assuming ideal (free space) fading makes calculation easy: $A = A_0/r^2$ where A_0 is the energy at some representative distance from the antenna, and r is scaled to units of that distance.

One weakness of this model is the exponent on r. The exponent 2 comes from the surface area of the expanding spherical wavefront, but oxygen absorbs radio waves (and re-radiates the energy at other frequencies), interfering signals build up over distance, and miscellaneous objects can obstruct the path of the signal. These and other problems have resulted in a large number of researchers trying to establish good approximating exponent values for different environments [Ale83, MK88, CMN83, CMN84]. A "multiple slope" model has also been used, in which different exponents are chosen for the various distance ranges [Ake88].

Another modification that has been explored is adding attenuation factors to the propagation loss (in decibels). An example of this is the Keenan-Motley model [KM90] which assigns a fixed attenuation to walls and floors (W and F respectively), and requires a knowledge of the number of walls and floors separating the transmitter and receiver, giving

$$Loss(dB) = L(\nu) + 20 \log d + kF + pW$$

where $L(\nu)$ is the "clutter loss" which represents the path loss at 1 metre with variance ν, and the $20 \log d$ term is just the exponential fading discussed earlier. The factors k and p represent the fading caused by each floor and wall respectively. Keenan and Motley found that the incorporation of the floor and wall factors ac-

counted for enough fading that they could use $20 \log d$ (free space attenuation) instead of guessing independent n values to get a $10n \log d$ relation like the one discussed earlier.

These two modifications to the basic model are improvements because they incorporate *site-specific* information into the prediction. It makes sense that prediction is more accurate for a known site than for a generic one. Unfortunately, they both use little information about the site, and hence are still only coarse approximations. One way to handle this problem is simply to accept that they are approximate. Using the predicted signal strength as a base value, multipath interference can be approximated for a general signal of that strength. This refines the result somewhat, giving a reasonable signal strength estimate. Techniques for this type of approximation are discussed in Section 3.2.

Another method of dealing with this inaccuracy is to use more site-specific information. Ray tracing techniques attempt to divide the transmitted signal into discrete components and follow the propagation of each component. The accuracy of the result can be adjusted by using a finer granularity when creating the initial rays. This method is discussed in Section 3.3.

3.2 Statistical Predictions

The most popular treatment for general RF propagation is the Rayleigh distribution.[1] This is a probability distribution for the amplitude of the received signal, due to multipath interference from random scatterers. This model has a uniform distribution for the phase jitter, and a probability density function for amplitude based on

[1]Lord Rayleigh; discussed by Hashemi [Has93].

a given variance σ (the average energy is σ^2).

An improvement on the Rayleigh distribution is the Rician distribution [Ric44]. The Rayleigh distribution assumes that all signals have random phase and similar amplitude, as is the case when a direct ("line of sight") signal is not present. Rice skews this distribution to favour a particular phase and amplitude, handling the case where a line-of-sight signal exists.

While these techniques are good in general, they do little to aid in the small-scale analysis of specific sites. The remainder of this paper will suggest alternatives that are very different from Rayleigh and Rician analysis. For further reading on statistical signal analysis, the textbook by Lee [Lee82] provides an introduction with examples, and the survey paper by Hashemi [Has93] gives a broad overview of what has been done and proposed.

3.3 Ray Tracing Techniques

Ray tracing is frequently used in computer graphics for accurately placing shadows and highlights, but a few important issues must be considered when trying to adapt techniques from graphics to the prediction of signal strength. In graphics, the illumination of the scene is known; the problem lies in finding out which light is transmitted in the direction of a specified "eye point." In signal propagation prediction, the eye point is replaced with a transmitter antenna, and the signal pattern in the immediate area of the antenna is the well-known quantity. This reverses the problem; the former case analyses information about the whole room to examine the small area immediately in front of an eye point, while in the latter case information about a small area (the transmitter antenna) is being used to deduce information about a large area. The significant result of this difference is

that rays straddling objects in the scene, which in the former case cause no difficulty, correspond to objects that ignore parts of the signal for the latter case. Techniques for controlling this type of error are discussed later in Section 4.1.

The problem of identifying optical paths between arbitrary locations in a scene and signal (light) sources is analysed by James Arvo [Arv86]. Arvo begins his discussion of solutions with a summary of the problem:

> The positions of the light sources offer not the faintest clue where to look except in the case of direct (unobstructed) illumination.

The specific problem addressed by Arvo's paper arises in computer graphics when light sources are reflected or passed through lenses. The same requirement is found in radio signal propagation, in which signals are transmitted and reflected by most indoor surfaces.

In computer graphics, spectral reflections still require knowledge of the orientation of the observer, so traditional (forward) ray tracing techniques must be used to determine the final scene; the backwards part of the ray trace is used only as a first step. The indirect light arriving at each surface is found first, and then the highlights are found in a slightly modified version of forward ray tracing [HH84, Arv86]. Since the radio signal propagation characteristics are being recorded for all parts of the room, however, this second step is not required. The strength at every point automatically incorporates spectral reflection through the reflected daughter cones (Section 5.1.5). The implementation of this backwards ray tracing is the focus of Chapters 4 and 5, but in the remainder of this chapter I introduce some of the theory behind the interactions being modelled, and the types of ray tracing applications that are already in use.

In adapting ray-tracing techniques developed for light rays, it is important to consider the behaviour of radio frequency (RF) radiation, and how it differs from light. The two main techniques for describing or predicting this behaviour are geometric optics and the geometric theory of diffraction.

Geometric optics assumes that the wavelength is short with respect to the size of interfering objects. It calculates the angles of the daughter rays resulting from an incident ray striking a partially mirrored continuous surface (in the radio frequency domain, this describes walls, doors, windows, and similar objects.) Radar cross section work can be used to calculate the strength of the signal along these rays [KST85, Ch.3].

The interesting part of the geometric theory of diffraction deals with curved surfaces but is fairly complicated, and therefore implementations are slow. For this reason, and due to the fact that most building walls are flat, it is common to assume that all surfaces are planar [MH91, HPL92, SR92, Val93]. (Some authors further assume that all walls are perpendicular to Cartesian coordinate axes [MH91, Val93].) The planar assumption allows reflection/transmission angles to be calculated quickly and easily.

3.4 Existing Ray Tracing Implementations

At Globecom '92, Scott Seidel and Theodore Rappaport presented the results of a ray tracing technique [SR92] that predicts the path loss and delay spread (time between the first arrival of a particular part of the signal and the last significantly strong arrival) of indoor RF signals. It traces rays in three dimensions, and uses point transmitters and receivers. Reflections are calculated for each ray until the

power level falls below a given threshold. Scattered rays are not reflected. Diffraction is not considered, but is identified as the object of ongoing research by the authors (see also [RRB92]). The results presented are for a single floor only, and show path-loss predictions with an error (in decibels) less than 11% (5 dB) in most places.

In a 1992 paper, Holt, Pahlavan, and Lee [HPL92] present a technique that uses only two dimensions to predict path loss and delay spread of indoor RF signals. Point transmitters and receivers are used. (Although there is an "area mode" in which a field of receivers will be generated, the paper suggests no optimisations for dealing with multiple receiving locations.) Results are for a single floor, and show path loss predictions within 5 dB of the measured values, and delay spread within 15 ns. These results are presented as the mean received values for a room, rather than for specific receiver locations.

It should be noted that the environment used by Seidell and Rappaport was more complicated than that used by Holt, Pahlavan, and Lee. The former used transmitters and receivers around corners from each other in corridors, while the latter used a collection of rooms in which transmission attenuation would account for most of the signal loss.

In a 1991 article, John McKown and Lee Hamilton present their solution to ray tracing an indoor wireless network [MH91]. They use a three-dimensional simulation with variable antenna characteristics (the papers mentioned above both assumed vertically polarised dipole antennas). The rays are traced from each destination point back to the source in a grid pattern. If two corners of a grid square return different values (outside a tolerance), that square is subdivided and recalculated recursively. The results are not compared to actual measurements. It should also be noted that theirs is the only published solution that calculated coverage

maps rather than finding results for a single receiver location.

Other optimisations that have been proposed include a ray splitting model [KUG93] that converts each ray to a collection of smaller rays once the original has gone a certain distance from its source. This prevents the rays from getting too spread out, resulting in a more accurate solution. The solution presented in the following chapters avoids the need for this periodic increase in granularity by considering the volume contained between adjacent rays, instead of the rays themselves. Chapter 4 will explain this variation on ray tracing, and compare it with the applications presented above.

Chapter 4

Practical Solutions and Optimisations

Having provided an historical perspective on signal propagation research and introduced the idea of ray tracing, I now discuss the high-level decisions that I feel are important in the use of ray tracing to build an efficient signal-strength prediction system. Each section presents the alternative solutions that I considered for a particular aspect of the problem, and explains which I prefer.

4.1 Cone Tracing vs. Ray Tracing

Rays are thin. That means that as a regular group of rays gets farther from its source, it becomes less dense, resulting in some negative effects (or *aliasing*) with objects in the distance. This effect is visually obvious when the scene contains regular patterns, since an observer can identify the pattern, and know what it should look like in the distance. An example of such a regular pattern is a large

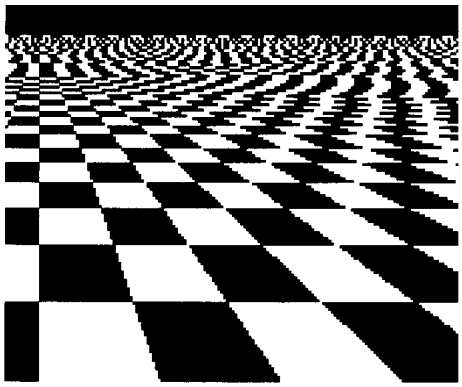


Figure 4.1: Point sampling of a big chess board [FvDFH91, Fig 17.12]

chess board, as in Figure 4.1. This effect is the result of rays spreading out so much that they miss some tiles. In forward ray tracing this is rarely a problem, since it results in a small inaccuracy for the receiver. In reverse ray tracing it is unacceptable. If areas are missed by transmitter rays, the prediction will be wrong for those areas.

One solution is to cast the initial rays very densely. This leads to a new set of problems when two or more rays, representing the same signal front, strike a single section of wall. Unless the algorithm has global knowledge, it cannot know that one of the rays is duplicating an earlier effect, so we are again faced with a technique that can give incorrect answers. A situation in which algorithms are able to make use of this technique is when there is a finite number of receivers, and scattered signals are not being considered. This means that intermediate values do not need to be stored since daughter rays are created when rays strike surfaces, but nothing is recorded. For each ray, the minimum perpendicular distance to each receiver is found. The total distance traveled by the ray at that point is combined with the initial transmission angles of separation to find how dense the wavefront is at that range. If a good (uniform) pattern was used for the initial transmission, then a

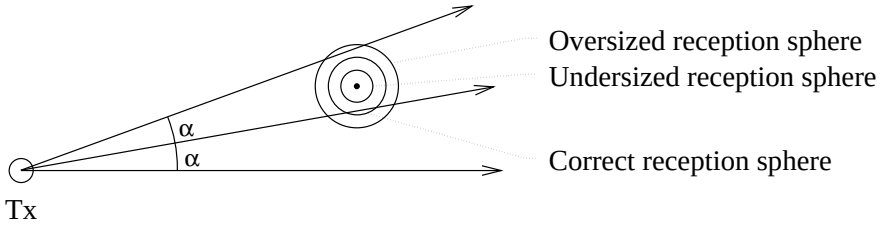


Figure 4.2: Two-dimensional view of a reception sphere. The total ray path length is d, producing a reception sphere radius of $\frac{\alpha d}{\sqrt{3}}$. [SIR92]

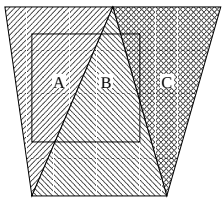


Figure 4.3: Several cones striking a single signal-intensity grid square

radius can be found for the antenna that will touch exactly one ray in the resulting field [SR92, SIR92, HBD+92] (see Figure 4.2). If the ray is within that distance, it will be the only one for which this test returns true. If the rays are less dense, larger spheres are used. In this way, each wavefront affecting the receiver will have exactly one ray used by this calculation. This does not help when the whole room (i.e. an infinite number of receiving antennas) is being analysed.

These problems with ray density are entirely due to the gaps formed as the field diverges. One solution comes from a computer graphics anti-aliasing technique that replaces the rays with solid angles (or *cones*). This ray tracing derivative is called cone tracing [Ama84]. Using cone tracing, the wave front is now composed of solid angles.[1] That means there are no longer any gaps.

If multiple cones from the same wavefront strike the same recording area, they

[1]For an introduction to solid angles and steradians, see [Ser86, §§24.6, pp.542–543].

will all have similar power and phase values. Therefore, multiplying the power of a cone striking an area by the fraction of the surface covered will cause the partial cones to add to the correct total value. For example, Figure 4.3 shows a case where three cones strike a single grid square. In this example, all three cones will reach similar conclusions about the strength and phase of their particular portion of the field at that square; call that strength x. If A strikes 50% of the square, B strikes 40%, and C strikes 10%, then the actual strength added to that square from all three cones will be $0.5x_A + 0.4x_B + 0.1x_C \approx x$. This solves the problem of multiple intersections with the same area in a way that cannot be applied in the context of thin rays. Solutions are now possible for the problem of missing areas, since the cones form a solid wavefront (some of the problems in actually implementing these solutions will be presented in Section 5.1.2.) Faster solutions to the problem of multiple intersections could be found by identifying the "first" intersection from a given wavefront, but these will not be explored here.

4.2 Cone Emission

Throughout this thesis, "cone" refers to the volume bounded by three rays diverging from a common source. The advantage of using cones instead of rays is that cones are infinitely dense. Since emissions spread out, rays can miss areas far away from the source, and so a large number of rays must be used to minimise this possibility. Since a cone sweeps out a volume, it can be large without missing areas. A cone can be traced by tracing its corners (in this case I use triangular cross-sections, so there are three rays to trace per cone). It is still important to note, however, that while the corner rays completely describe the cone, care is required to examine the volume they enclose; it is often a hard problem to identify and describe the

interactions between a cone and its environment.

One problem with using large cones comes from the spherical nature of the wavefront. Linear interpolation between two distances will give a linear change in *radius*, i.e. polar coordinates. This will not always be a line segment. For example, consider two points that are equidistant from the source. In this case, a linear variation of radius will be a constant, but the only curve with constant radius is a portion of a circle. The problem stems from the fact that the polar coordinates (radius and angle) for a straight line are not linearly related, but the cone has struck a plane so the actual region of intersection will be flat, therefore there will be error associated with a linear approximation. The difference between the linear interpolation between the corners of the cone and the actual distance to the region of intersection will be called the "bulge."

I propose two methods for limiting the inaccuracies caused by this bulge. The first is an adaptive algorithm based on the fact that, given the solid angle spanned by the cone and the distances traveled by the corners, the absolute error can be bounded. This allows a tolerance to be specified, and if an error above the tolerance is possible, the cone can be subdivided.

The second method is an exact technique. Since I consider only planar surfaces, all transformations will be affine. Therefore, lines that converge before the transformation will also converge after the transformation (due to the fact that affine transformations preserve ratios of distances). Furthermore, the point of convergence can be found by performing the same transformation on the original focus (source). This will give a virtual source that can be used in standard distance calculations to find exact distances to all parts of the cone.

4.2.1 Adaptive Algorithm

The bulge that results from bilinear interpolation of distances to interior points from the distances to the points of intersection of the boundary cones defining the cone, as described above, will not usually be large when compared to the total distance travelled. As explained in Section 2.1, however, the short term effects of systematic interference from multipath are very sensitive to distance. These effects vary with absolute differences in distance, regardless of total distance travelled to that point. Therefore it is important to have an absolute bound on the possible error associated with any distance calculation.

Bilinear interpolation removes the need for square root calculation when determining the distance to an interior point by assuming linear variation of distance along each edge of the intersection area, as well as along any straight line passing through it. In fact, distance is a second degree function, but a small section of a comparatively large ellipsoid is very close to a plane. If the error associated with this approximation can be bounded, then any time a cone is too large, it can be subdivided. In this way, the error in each intersection will be kept below an arbitrary pre-specified value. The actual tolerance will depend on the wavelength, since over short distances (on the order of one or two centimetres), the relative phases of incoming signals will usually have much greater effect on the combined strength than the free space attenuation over the same distance.

The problem lies in bounding the error associated with the bulge of a cone. Consider first the two dimensional case.

Call the distance determined by linear interpolation $\tilde{d}$ and the actual distance d (see Figure 4.4). Since the points are defined by distances only, the coordinate system is arbitrary. A convenient coordinate system has its origin at the source of

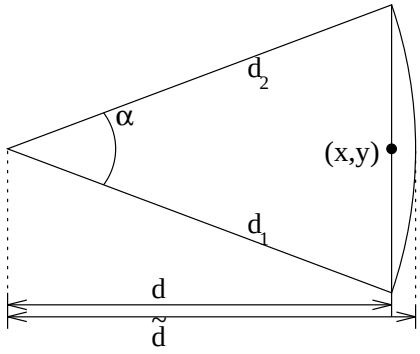


Figure 4.4: Cross-section of a cone

the cone, and takes the point defined by d_1 to be on the x-axis, i.e. located at $(d_1, 0)$. The second point is now at some angle α from the x-axis, and at a radius d_2 from the origin, i.e. at $(d_2 \cos\alpha,\ d_2 \sin\alpha)$. That means that the parametric interpolation (using $0 \leq \ell \leq 1$ as the arbitrary variable) is $((1-\ell)d_1 + \ell d_2 \cos\alpha,\ \ell d_2 \sin\alpha)$.

This means that the two distance values are given by

$$\begin{aligned} d &= \sqrt{[(1-\ell)d_1 + \ell d_2 \cos\alpha]^2 + (\ell d_2 \sin\alpha)^2} \\ \tilde{d} &= (1-\ell)d_1 + \ell d_2 \end{aligned} \tag{4.1}$$

At an arbitrary interior point ℓ between the endpoints the error will be $\tilde{d} - d$, but this is a complicated result, so some approximations help.

First, notice that the derivative of error, with respect to d_1, is given by the following:

$$\frac{\partial}{\partial d_1}\tilde{d} - d = (1-\ell)\left(1 - \frac{(1-\ell)\frac{d_1}{d_2} + \ell\cos\alpha}{\sqrt{[(1-\ell)\frac{d_1}{d_2} + \ell\cos\alpha]^2 + \ell^2\sin^2\alpha}}\right)$$

Inspection shows that for positive d_1, this derivative is positive (approaching zero for large d_1). Therefore, $\frac{\partial}{\partial d_1}\tilde{d} - d \geq 0$ for all $d_1 \geq 0$. That means that increasing d_1 will make the error bound larger, i.e. looser, regardless of the relative value of

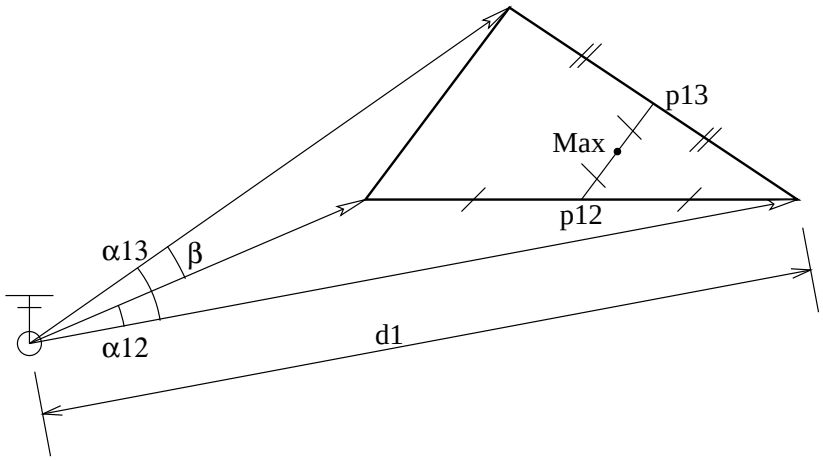


Figure 4.5: Expanding error bounds to three dimensions: p12 and p13 are the points for which error is bounded by e_{12} and e_{13}, and the point labelled "Max" is the point of maximum error.

d_2. By symmetry, d_2 also has this property. Therefore, replacing d_1 and d_2 with $d_{max} = \max(d_1, d_2)$ will still give an upper bound on error.

Second, differentiating $\tilde{d} - d$ shows that it is maximized at $\ell = \frac{1}{2}$ (see Appendix B). Using these approximations gives the following:

$$\begin{aligned} \tilde{d} - d &\leq d_{max} - d_{max}\sqrt{\tfrac{(1+\cos\alpha)^2+(\sin\alpha)^2}{4}} \\ &\leq d_{max}\left(1 - \sqrt{\tfrac{1+\cos\alpha}{2}}\right) \end{aligned} \tag{4.2}$$

By symmetry, the property that the error is maximised at the midpoint of the arc can be extended to the third dimension. Choose a base point to be d_1 from the source. Label the two adjacent angles of emission α_{12} and α_{13}. Call the angle of emission between the other two points β (see Figure 4.5). The error for this case will be the error of a two dimensional linear interpolation again, except this time the base points (p_{12} and p_{13}) have error associated with them. It makes calculation simpler to assume that both p_{12} and p_{13} have the same error, so let d_{max} be the

maximum distance to any corner, and α be the maximum of α_{12} and α_{13}. Since the error will be in the same direction as before (the approximation is too large), it will be maximised where the errors on the two sides are maximised. This means that the new error value will be limited by Equation 4.2, except using the base distance $d_{max} + \text{Error}_{2D}$. Since the largest angle of emission between these two edges is β, Equation 4.2 can be used to show that the maximum error is given by the following:

$$\begin{aligned} \text{Error}_\beta &\leq (d_{max} + \text{Error}_\alpha)\left(1 - \sqrt{\tfrac{1+\cos\beta}{2}}\right) \\ &\leq \left[d_{max} + d_{max}\left(1 - \sqrt{\tfrac{1+\cos\alpha}{2}}\right)\right]\left(1 - \sqrt{\tfrac{1+\cos\beta}{2}}\right) \end{aligned} \tag{4.3}$$

By again noticing that the bound gets looser as angles get larger, this can be further simplified by using $\gamma = \max(\alpha, \beta)$.

$$\text{Error}_\beta \leq d_{max}\left[2 - 3\sqrt{\frac{1+\cos\gamma}{2}} + \left(\frac{1+\cos\gamma}{2}\right)\right] \tag{4.4}$$

Since both Error_α and Error_β are absolute deviations from the "correct" value interpolated by their endpoints, their errors must be added not multiplied. Since the maximum errors from the first step were added to the base values used by the second step, Error_β will be too large, so addition will still produce a valid upper bound. The resulting maximum error (using the fact that $\cos\gamma \leq \cos\alpha$) is limited by the following:

$$\begin{aligned} \text{Error}_{3D} &\leq \text{Error}_\alpha + \text{Error}_\beta \\ &\leq d_{max}\left(1 - \sqrt{\tfrac{1+\cos\gamma}{2}}\right) + d_{max}\left[2 - 3\sqrt{\tfrac{1+\cos\gamma}{2}} + \left(\tfrac{1+\cos\gamma}{2}\right)\right] \\ &\leq d_{max}\left[3 - 4\sqrt{\tfrac{1+\cos\gamma}{2}} + \left(\tfrac{1+\cos\gamma}{2}\right)\right] \end{aligned} \tag{4.5}$$

Again using the fact that this is an approximation, it is acceptable to truncate the angle α to, for instance, two decimals. This will make the total number of possible values small enough to use a table look-up. The values of $3 - 4\sqrt{\frac{1+\cos\gamma}{2}} + \frac{1+\cos\gamma}{2}$

can be calculated at the beginning of the simulation for all possible values of γ, and looked up during execution. That means that once the longest corner ray is known, the error can be bounded with one multiplication.

This bound can be justified by considering a triangle of constant size at an increasing distance from the source. As the distance approaches infinity, γ approaches 0. Since $\frac{d}{d\gamma}\cos\gamma$ at $\gamma = 0$ is zero, $3 - 4\sqrt{\frac{1+\cos\gamma}{2}} + \left(\frac{1+\cos\gamma}{2}\right)$ approaches zero faster than d_{max} approaches infinity, i.e. Error_{3D} approaches zero at large distances. At an infinite distance from the centre, a finite portion of the surface of a sphere is flat. Between points on a flat surface, linear interpolation is exactly correct, so the error is zero. Therefore, this bound is exactly correct at large distances from the source.

If the error cannot be bounded to be less than the predefined tolerance, the cone can be divided into three cones by casting one new ray in the centroid of the cone. These new cones are handled recursively.

For this method, it should be noted that subdivision is a source of error, since the ray cast from the centroid may not have the correct length. If the source of this cone is a wall, the distance to the centroid will have been interpolated from the corners of that intersection. This will add a small error to each of the daughter cones.

4.2.2 Exact Algorithm

The advantage of the previous technique is that the distances from the source to all sections of the wall can be calculated with linear complexity. This advantage is rapidly eroded by large or highly reflective environments. In these situations, the cones may require frequent subdivisions, and each subdivision of a cone requires

new rays to be intersected with the room. This is very slow.

Taking advantage of the fact that all transformations are linear reflections allows an alternative solution. After a cone is reflected in a plane, the point at which the edges intersect is the same distance from the point of reflection as the actual origin. Call this point the *virtual origin*. By storing the virtual origin with the other information about the cone, finding the exact distance to any point is just a standard distance calculation.

With this technique, the only limit on cone sizes is that they must be small enough to not have the edges straddle a wall (discussed in Section 5.1.2,) so this is a minimal restriction to the technique. Since small objects interfere with the theoretical accuracy of the technique anyway (see Section 2.2), this limit should be less restrictive than the one imposed by the algorithm in Section 4.2.1.

Despite this less restrictive size limitation, I believe that the adaptive algorithm will be faster since it uses only one calculation (which turns out to be a very fast one, as will be shown later in this section) to determine the quality of the linear approximation. The exact algorithm uses three multiplies, two adds, and a square root for each interior point used. At this time, however, only the exact algorithm has been implemented.

4.3 Representing Different Antennas

As much as it would help signal strength prediction, all antennas are not perfect isotropic radiators. If directional antennas are to be employed as transmitters, a transmission characteristic relating the angle of emission to the signal intensity must be provided. Emitted cones will take their intensity from this information,

and propagate as usual. The only difficulty here is that the cones will have to be small enough to give a reasonable resolution for the emission intensity.

If directional antennas are to be used as receivers, the situation becomes even more difficult. Since all incoming signals are added together, the original directions are lost. This means that a directional receiving antenna could be simulated, but the direction would have to be specified ahead of time, and would remain constant for the entire room. This could be implemented by sending incoming rays through a function that attenuated them based on direction before adding their values to the stored volume totals.

A possible future feature would be a two-pass analysis in which the first pass identifies the strongest incoming signal to each volume, and the second pass assumed that the antenna in each volume was oriented in the appropriate direction. This might give an idea of the best possible reception in each area.

The usefulness of representing directional receiving antennas is an interesting area for future research, but is beyond the scope of this investigation.

4.4 The Storage of Interim Values

The attributes of waves arriving at any space or surface must be stored in a compact manner, and must be easily accessible during the ray trace. This is because there is a large number of discreet areas being examined, and they are constantly being updated. The simple rooms discussed in Chapter 6, for example, are 6×12 metres with a resolution of 40 per metre in each direction. This means that 115,200 discrete values are required to represent the signal-intensity grid (a two-dimensional data plane discussed below). The walls are stored at a lower resolution (10 per metre in

each direction), but still require a significant amount of memory.

The location of the data is also important. Ray tracing distributes well across multiple computers, resulting in multiple separate processes updating the data. The natural worry in such a situation is about bottlenecks on data locks, but since waves affect each other only locally, and results are commutative, signal information can be kept independently on all machines being used for the trace. The results of each machine's contribution can be combined at the end to get the total picture (before the scattered signals are computed).

Ideally, this technique could be used to predict the signal strength at all points within the room(s) of interest. However, it should be sufficient to predict the signal strength on a single plane in the room (parallel to the floor) because the wireless devices using the network are usually either desktop or handheld, and can be expected to be at a roughly constant distance above the floor. The array used to store the signal intensities in this plane is called the signal-intensity grid. This partitions the portion of the plane to be measured into rectangular blocks. As new portions of the signal are identified, they are added to the sum of all the components already identified in the same area. When the simulation is complete, this array will hold the total predicted field intensity at each point in the specified plane.

Walls also accumulate signals by superposition, but the addition of new signals to wall sections has side effects: incoming signals cause daughter rays to be cast into the room and traced recursively. Daughter rays were discussed in detail in Chapter 2.

4.5 Intersection Calculations

Intersection calculations are slow. To characterise the behaviour of one ray, it must be intersected with all elements of the scene to find the closest intersection.

Under some circumstances, it may be possible to model this intersection calculation in two dimensions [SIR92]. This would make the solution much more computationally tractable. An intersection region would no longer be required; instead, the two sides of the cone would be traced through a regular grid. (This is an appropriate application for Bressenham's line algorithm, which draws sloped lines on grids using only integer arithmetic, as discussed in [FvDFH91].) Some squares would be passed through easily, and others would suggest that a collision is possible, so a check would be made, and the line would either end (hit a wall), or miss the object and continue. This would also make the intersection calculation much easier, since only one box would be checked at a time. This is a ray tracing technique known as "spatial partitioning" [FvDFH91, pp.664–665]. It greatly increases the efficiency of the intersection calculations, and it has natural application here, since the room must be partitioned in order for the two-dimensional signal data to be stored (the signal-intensity grid). Unfortunately, Lecours et al [LGBC93] show that significant areas of poor signal strength can be caused by reflection from the ceiling and floor. Whether or not this is a problem in practice is unknown.

4.5.1 Intersecting rays with walls

When defining a new room, a convenient representation for the walls is to first specify the location of one corner as a 3-D coordinate. With this point fixed, identify unit vectors from that corner along each of the adjacent edges of the wall, defining its orientation, and specify how long each of those edges is. This is how my

sample implementation accepts room definitions (see Appendix A). For improving the efficiency of intersection calculations, it is helpful to derive from this information the three-dimensional equation which represents the plane of the wall. Therefore, for purposes of intersection calculations, I will assume that a wall object contains the following information:

- one corner: (x_0, y_0, z_0)
- unit vectors defining the wall-space: $\hat{\mathbf{x}}', \hat{\mathbf{y}}'$
- far corner in the wall-space: (x_1', y_1')
- constants from $ax + by + cz + d = 0$ defining the plane: (a, b, c, d)

For intersecting rays with walls, a ray extends from a source $(x_s,\ y_s,\ z_s)$ through a point $(x_t,\ y_t,\ z_t)$. This leads to the parameterised ray equation:

$$(x_s + \ell(x_t - x_s), y_s + \ell(y_t - y_s), z_s + \ell(z_t - z_s)). \tag{4.6}$$

Let $\Delta x = x_t - x_s, \Delta y = y_t - y_s, \Delta z = z_t - z_s$;
now the ray is given by $(x_s + \ell\Delta x, y_s + \ell\Delta y, z_s + \ell\Delta z)$.

Intersecting this ray with the plane $ax + by + cz + d = 0$ gives

$$a(x_s + \ell\Delta x) + b(y_s + \ell\Delta y) + c(z_s + \ell\Delta z) + d = 0$$

$$\therefore \ell = -\frac{ax_s + by_s + cz_s + d}{(a\Delta x + b\Delta y + c\Delta z)}. \tag{4.7}$$

Substituting Equation 4.7 into Equation 4.6 gives an intersection point (u, v, w). Transforming this into wall-space coordinates gives

$$(u', v') = ((u, v, w) \cdot \hat{\mathbf{x}}', (u, v, w) \cdot \hat{\mathbf{y}}') \tag{4.8}$$

If $0 \leq u' \leq x_1'$ and $0 \leq v' \leq y_1'$ then the ray has intersected the wall segment. For each intersection, the square of the distance $d^2 = (u - x_s)^2 + (v - y_s)^2 + (w - z_s)^2$ from the source to the intersection point is recorded. Once all intersections have been performed, the smallest distance is the intersection that is used.

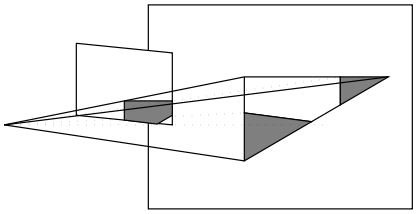


Figure 4.6: Occlusion of background walls (shadows)

4.5.2 Shadows

If a cone strikes multiple surfaces, one surface may partially occlude another. In order for this to be properly incorporated into the model, the various walls struck by a cone must first be processed in depth-first order (see Section 5.1.4). If part of the cone does not strike a given wall (as must be the case if multiple surfaces are struck), standard clipping algorithms can be used to find out where the edges of the cone left the wall in question. By tracing these intersections and any corners in the foreground object to the background object, the resulting area can be shaded. The actual values will still come from interpolations from the original rays; these subsequent intersections just find the portion of the surface affected. This technique is applied recursively.

4.5.3 Identifying the area of intersection

Characterising the behaviour of the rays defining a cone is only the first step towards characterising the behaviour of the cone. The results that matter are the interactions between the interior of the cone and the scene. Once a cone has struck a wall, two fills (area-effect calculations) are required: one to record the cone's interaction with the signal-intensity grid (Section 4.4), and one to record its interaction with the wall (see Section 4.6).

When a cone is added to the signal-intensity grid, it is important that the algorithm first knows which wall it is hitting. The wall being struck and the wall that is the real source of the emission (if one exists) are used to clip the intersection area of the cone with the plane of the grid. That clipping will produce a polygon that must be "filled," much like a regular polygon fill, but with one difference. The edges need to be shaded for partial intersections so that when the next cone shares an edge and adds its strength, the result will flow smoothly from one polygon to the other. The net result should be as if the signal intensity for each square was calculated (using one of the methods discussed in Section 4.2), then multiplied by the fraction of the square inside the region of intersection.

I found no published solutions to this problem, and implemented a solution that is not very efficient. A source of difficulty is squares that are partially covered by the cone. Calculating the covered fraction of the square is expensive. One idea for a better solution is this: when a cone is cast, if one edge is considered "inside," and the other "outside" the cone (already covered by the previous cone), then partially covered boxes on the inside edge are "hits," and partially covered boxes on the outside edges are "misses." This would remove the ambiguity, and provide an important optimisation, but might not be possible without some global knowledge of the cones. More work is needed on such optimisations.

4.5.4 Restrictions

By storing only two corners for each wall, wall segments are restricted to be parallelograms. This allows the intersection calculation to be very fast, since in wall-space the wall segment is a square parallel to the axes and based on the origin. If speed were sacrificed in this intersection calculation, more general shapes could be used.

In fact, the current implementation requires walls to have right-angle corners. This is to ensure that the ratio of areas in the two wall-spaces is unity, so an appropriate power can be radiated during the calculation of scattered signal. General parallelograms could be handled by finding the world space area a of what is a unit square in wall space, then multiplying the length of both basis vectors by $\frac{1}{a}$. This would affect only the setup time.

4.6 Diffuse Scattering

Every time a section of wall is struck, there is potential for diffuse scattering. Determining the effect of diffuse scattering on the signal-intensity grid is an expensive operation, so tracing this signal component required a better solution than the obvious one before it could be practical. I have chosen to leave it until all the primary rays have been traced, and the total signal is known for all parts of all walls. This trades memory for speed, since the diffuse scattering from each wall section need only be calculated once, instead of every time a section is struck by an incident cone. The memory cost is caused by the additional storage required to maintain the composite signal at every wall section.

Diffuse radiation is added in a sphere about the centroid of the wall section. The base intensity and phase of the diffuse radiation from each wall section is given by the complex vector sum of the radiation received at that section, and the intensity fades as the square of the distance from the secondary emitter (see Section 2.5). These signals are not reflected. Attempting to do so would lead to an iterative solution similar to a progressive refinement radiosity technique used in computer graphics [CsECWG88], but would probably not help significantly, since the diffuse component of most reflections is small.

How these algorithmic design decisions, methods for data representation, and techniques for intersection and reflection are actually implemented in a sample application is discussed in Chapter 5. The analysis of their usefulness, incorporating data gathered from the sample application, is then presented in Chapter 6.

Chapter 5

Prediction Tool Design

This chapter will explain the prediction tool. Knowledge of the broad concepts, such as cone tracing and radio frequency interference have already been covered in Chapters 2 and 4, and will not be repeated.

The general model on which the simulator is built involves three stages:

1. Generate a model of the scene.
2. Predict the behaviour of the signals emitted from sources identified in step one.
3. Present the result.

The first step is implemented as a simple parser that accepts descriptions of the walls to be modelled, room size, an altitude and resolution for the signal-intensity grid, and a bounce limit. It allocates memory for the arrays which will store the discretized information about incoming signals at all points on walls and the signal-intensity grid, and sets up a linked list for searching through all walls in the scene. Sample input files are included in Appendix A.

The third step just interprets the arrays generated in the first two. The current tool dumps a two-dimensional picture of the room to an X11 server, generates a graph of a single linear slice of the room in a format suitable for xgraph, or dumps a three-dimensional image of the room (including signal intensities for wall sections) in an in-house format suitable for viewing on an SGI console.[1]

The second step is the subject of the remainder of this chapter.

5.1 Basic Functionality

The simulator divides the problem into the following sub-problems:

1. Emit the initial cones.
2. For each cone, identify which walls will be affected by that cone.
3. Choose an order in which to process these walls
4. For each wall, create daughter cones for the parts of the cone that interacted with the wall, and intersect the other parts of the cone with the remaining walls.

These subproblems are dealt with as follows:

5.1.1 Initial Cones

As I described in Section 4.2, the word "cone" refers to a volume bounded by three rays. Since any three-dimensional collection of points that defines a surface can be

[1]The three-dimensional viewer used was written by Stephen Mann; this file-format will not be discussed here.

divided into triangles (tessellated), cones can describe any emission pattern from a point source with no gaps or overlap in the volumes spanned. Thus, when I say that an antenna "emits the initial cones," I mean that a triangular mesh has been constructed around the antenna location (point source), and the rays from the source through the vertices of this mesh have been arranged into groups of three to form a collection of adjacent non-overlapping cones that pass through every part of the surface. Note that if these cones were extended infinitely, they would span three-space with no gaps.

For an isotropic radiator, this step is not difficult. All emitted cones will be of the same intensity, so it is sufficient to emit them through the faces of any polyhedron centred on the antenna. Since the entire cone is traced, it does not matter that this division is non-uniform.

The problem is more difficult for complicated antennae. A resolution must be chosen for dividing the initial signal into a collection of cones that can be considered each to have a constant effective radiated power. This has not been implemented.

5.1.2 Identifying affected walls

Since a cone is just a way to describe a particular volume, it has no physical meaning by itself. The value of cones lies in their ability to define the part of the total signal that is being considered in any part of the solution. A cone becomes meaningful only when it is used to find the interaction between its portion of the signal and some object in the scene. What I call here "affected walls" are those walls that interact with the portion of signal described by a particular cone. This does not include indirect interactions (such as those resulting from a reflection or transmission); those will be handled by the daughter cones discussed later.

Identifying affected walls is an old problem in ray tracing applications. In this case I chose to maintain two descriptions of each wall. One is the a, b, c, and d values that specify the plane ($ax + by + cz + d = 0$) in real space; and the other is the location of an origin, two basis vectors, and the top right corner (in the coordinates of those basis vectors). In the local coordinate system all walls are squares with a diagonal on (0,0) and the given top right corner.

One of the fundamental assumptions of the implemented solution is that the interactions of the corners characterise the interactions of the cone. That is to say that the set of walls struck by the cone is the same as the set of walls struck by the corners of the cone.[2] Therefore, the problem of identifying cone interactions can be broken into two parts: identifying corner-ray interactions, and combining these interactions to characterise the cone.

A single ray interacts with at most one wall. In general, the wall with which the cone will interact is the one for which the intersection point is closest to the source of the ray. However, if the cone containing the ray has been reflected or transmitted (or both), then the ray has a "virtual source" (convergence point for the rays) that is not the actual source of the signal. Anything between this point and the last wall encountered is ignored. In the actual implementation, the rays that define daughter cones are the difference between the point of interaction with the wall causing the new cone and the reflected base of the parent cone. This means that all cones that are the result of one or more previous interactions with walls will ignore further interactions that are less than distance 1.0 (normalized to the length of the ray) from their source.

Ray interactions are detected by examining all walls in the scene, and using

[2]The validity of this assumption, and methods for maintaining it at runtime are discussed in Sections 5.1.3 and 5.2.

their global coordinate representations to identify the distance to an intersection (if one occurs). If that intersection is the closest so far, then the location is found in local coordinates (for a global 3-space point P and the corresponding local 2-space point P', $P' = (P \cdot \hat{\mathbf{x}}, P \cdot \hat{\mathbf{y}})$, where $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ are the basis vectors for the wall) to determine if the intersection is actually in the portion of the plane describing the wall. Since that portion is just a square based on (0,0) and a point in the positive quadrant of the plane, this comparison is simple. If the intersection is on the wall, then that wall becomes the new best candidate for closest.

The goal of finding the interactions of the corner rays, of course, is to draw conclusions about the interactions of the cone. Once a candidate set of walls is found, it is the function of the intersection routine to subdivide the incident cone into cones that strike exactly one of those walls each. If parts of the cone strike none of the candidate walls, then those parts are returned as a set of "external cones." This is achieved with a loop that maintains a set of cones (initially the single incident cone). For each candidate wall, these cones are passed, one at a time, to a clipping function. The cones that strike the wall are passed on to the wall object, which creates whatever daughter cones are appropriate (see Section 5.1.5), and the cones that do not strike it are returned. This set of returned cones is added to the external list, and passed on to the next candidate wall. After the set of candidate walls has been exhausted, the remaining set of external cones represents the portion of the initial signal that did not strike any of the candidate walls. These are the external cones passed back to the top-level trace routine.

Since the candidate walls are those struck by a cone corner, a maximum of three walls will be identified. If a wall is missed by all three corners, and the portion of the cone that should have struck that wall did not strike any of these three, the interaction will be detected in the next iteration, as explained above. This is

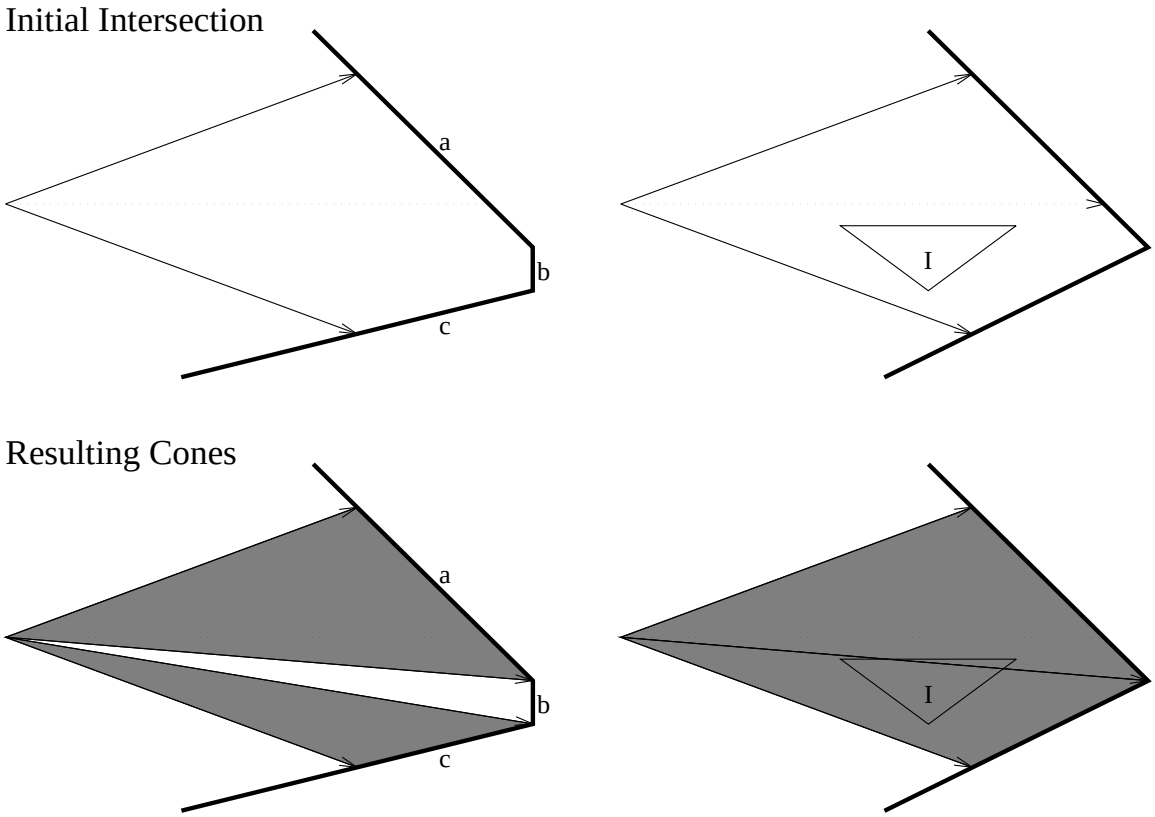


Figure 5.1: A cone striking interesting wall geometries

shown in the left example of Figure 5.1, in which the shaded areas show the parts of the cone that will be filled, and the unshaded area will be returned to the calling routine as one or more external cones. The interaction with wall *b* will be detected when these cones are passed recursively to the trace procedure.

If a portion of the cone that should have interacted with the wall also intersects one of the walls that was detected, as with all the walls of object I in the right example of Figure 5.1, that portion will not be returned as an external cone, since it was used. Therefore, that interaction will never be detected.

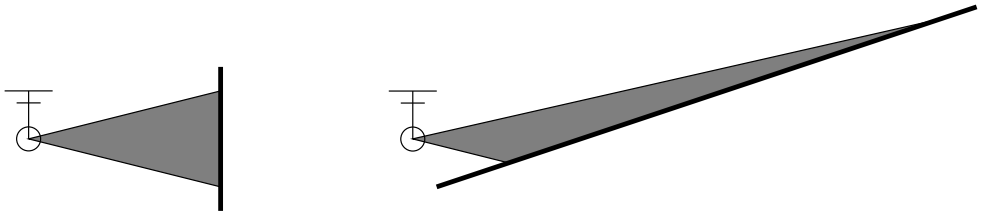


Figure 5.2: Cones intersecting planes at different angles

5.1.3 Limiting Cone Size

The undetected interactions described above occur only when the corners of a single cone straddle a wall. An obvious conclusion that can be drawn is that the undetected interactions are symptoms of cones that are too big. Unfortunately, "too big" is relative, and its definition is not obvious.

The problem with identifying a cone that is too big is that what actually needs to be limited is the size of its interactions with planes. When the three-dimensional cone is intersected with a two-dimensional plane, the area of intersection can be infinite (see Figure 5.2). The implemented solution combines two methods. If a ray cast through the middle of the cross section (defined below) strikes a wall other than one already detected, the cone is divided into three as in the left example in Figure 5.3. The second condition compares the length of the line segment connecting the wall intersections of two corners, projected into the plane perpendicular to the ray through the middle vector (middle is defined the same as in the first check), to a defined maximum length. If the line segment is too long, the cone will be subdivided along the long edge as in the right example in Figure 5.3.

The former measure is based on the hope that if a significant fraction of a wall was missed, then it will exist in the middle of the cone. It is worth noticing that if the middle ray strikes nothing at all, but the corners all struck actual walls, this

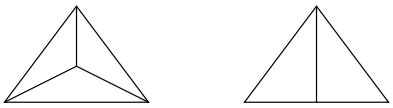


Figure 5.3: (Left) When a new wall is discovered in the centre of the cone, and (Right) When one side of a cone is too long

will also cause a subdivision. The middle of the cone is defined as the ray through the base and the average of the three corner points. If the plane defined by these corner points is not perpendicular to the direction of travel (i.e. not parallel to the plane for which the cone has minimum surface area), the transformation between the two planes is just a shear, and therefore affine. This means that the ray found by this method will be the same for all choices of points on the three corner vectors.

The latter check tries to minimize the need for the former. Projecting the vector into the plane perpendicular to the middle vector (multiplying by the dot product) avoids cases where a cone is subdivided just because it struck a steeply sloped wall. The magnitude of this vector, where the edge and middle vectors are $\vec{e}$ and $\vec{m}$, and $\hat{m} = \frac{\vec{m}}{||\vec{m}||}$, is $||\vec{e} - (\vec{e} \cdot \vec{m})\hat{m}||$. If the largest of the three values (one for each edge) is greater than an arbitrary constant, the cone is subdivided.[3] If more than one edge was too long, then the new cone containing that edge will be subdivided by the recursive call.

[3]In the actual implementation, the value used is the square of the magnitude. This avoids a square root calculation, and costs nothing, since a constant that limits the square of a value is still a constant.

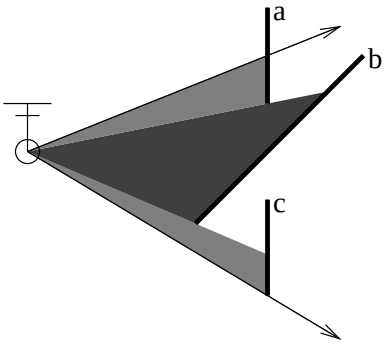


Figure 5.4: Determining order for processing walls

5.1.4 Ordering the walls

When a cone is intersected with a wall, some of the cone might not directly affect that wall. When this happens, the remaining part of the cone is tessellated (converted into a group of triangular cones), and returned to the calling procedure to be passed on to the next candidate wall, as discussed in Section 5.1.2. It is therefore important that if wall A causes a shadow on wall B, then wall A should be processed first. Otherwise, portions of the signal will be added to B that should have struck wall A.

If all parts of one wall are closer to the source than all parts of another, then the order in which they should be processed is clear: closest first. The tricky part is when the walls overlap in space. Consider the two-dimensional case presented in Figure 5.4. In this example, walls a and c are equidistant from the antenna, yet wall a occludes wall b, and wall b occludes wall c. The problem lies in geometrically characterising the interrelationship between the walls and the source such that these orderings can be identified.

The first step in the solution is to appreciate that processing a before b does not necessarily mean that a must occlude b. The only important requirement is

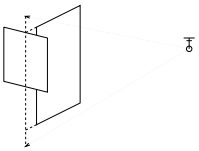
that it must be impossible for b to occlude a. Therefore, the walls are sorted such that no wall could possibly occlude a wall earlier in the list. Sorting a list is a well researched problem, requiring only a method for consistently comparing two values. Figure 5.5 is an algorithm for determining the order of two walls. For simplicity, this description uses the graphics term "eyepoint" to mean the source of the cone.

Grouping the half-planes by the side of P_e on which they fall divides the problem into two areas, each with a unique order of precedence. In each of these groups, one half-plane is always closer to the eye than the other, and there is no line through the eyepoint intersecting both groups so they do not interfere with one another. This explains 3b. If there is only one pair in which order actually matters, determine which is the closer plane of this pair, and identify it as the only one that might occlude the other.

Case 3a presents a special challenge. When both walls exist on both sides of the line of intersection, it is not immediately obvious which pair of half planes might cause occlusion. To determine which pair is representative of the whole, an edge on one wall is first found that is closest to the other wall (such as the line marked A in the picture next to step 3a). If any occlusion exists, this edge will be involved. As in the other cases, the half-planes are grouped by which side of the line of intersection they are on. If the constructed plane intersects the other wall (for either half-plane pair), then that pair will give the occlusion for the two walls. Otherwise, no occlusion takes place, either order will do.

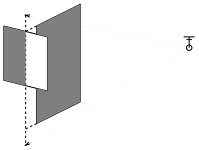
5.1.5 Adding a cone to a wall

Once it has been determined that a cone has a clear path from its source to a particular wall, it is "added" to that wall. The operation for adding a cone to a



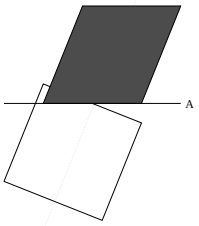
1 Intersect the planes of the two walls.

- If the planes are parallel, use the method described in Figure 5.6 to determine order. In this case, the points selected can be anywhere in either plane (may as well use origin points.)



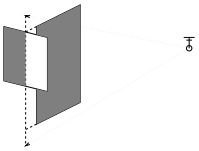
2 Divide the planes into half-planes at the line of intersection, and divide those into pairs (in this example, only one wall has pieces in both half-planes).

- Construct a plane P_e through the eyepoint and the line of intersection.
- Divide the half-planes into two pairs, one pair on each side of P_e.



3a If both walls exist in both half planes, determine which pair of half planes can be used to determine order.

- Find the edge of one wall that is closest to the other wall. This will be one of the edges cut by the plane constructed in 2 (e.g. the edge marked 'A' in this diagram).
- Construct a plane on this line through the eyepoint
- For the first pair:
  1. If this plane intersects the other wall, there is occlusion in this pair. Determine order as in Figure 5.6.
  2. If no occlusion was discovered, repeat for the second pair.
- Skip 3b.



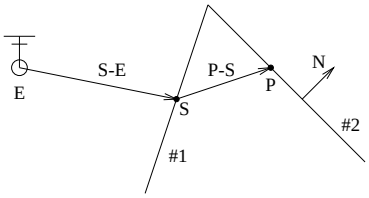
3b Consider only the pair of half-planes in which both walls have pieces (the shaded sections in this example).

- There is at most one such pair, because of step 3a
- If each pair contains only one wall, the walls do not occlude one another, and order does not matter.
- This pair is the only one in which occlusion is possible. Use the algorithm from Figure 5.6 to determine order.

Figure 5.5: Method for determining possible occlusion between two walls

wall encapsulates several distinct components which are all executed once a cone is known to intersect exactly one wall. These encapsulated components are the creation of daughter cones, identifying and recording contributions to the measured electric field in the signal-intensity grid, and adding the intensity of the incoming field to the recorded total received energy in the affected portion of wall.

- Pick a point in each half-plane. Let S be in half-plane 1, P be in half-plane 2, E be the eye point, and N be a normal to half-plane 2
- Iff ((S-E) dot N) * ((P-S) dot N) $>$ 0 then 1 might occlude 2.



(In this example, #1 might occlude #2.)

Figure 5.6: Method for determining possible occlusion between two half planes

The signal-intensity grid is a two-dimensional array that spans part of a horizontal plane (see Section 4.4). Until the extent of a clear path is identified for a cone, its interaction with this grid cannot easily be characterised. Once a portion of a cone actually strikes a wall, that portion is intersected with the signal-intensity grid, the complex vector describing the signal at each grid point is found ($\frac{\sigma}{d^2}\{\cos\theta + i\sin\theta\}$ where d is distance, σ is source intensity, and $\theta = \frac{d}{\lambda}$ is phase), and then the signal is added to what is already known about signals received at the grid in the area of intersection. This is a two-component vector value (real and complex components of the signal) in a two-dimensional array that represents the signal in all squares in the signal-intensity grid.

The two fundamental types of daughter cones are the reflected and the transmitted cones. The transmitted cone is exactly the same as the incident cone, except that the ERP for the signal is multiplied by the transmission coefficient associated with the wall's building material. Similarly, the reflected cone's ERP is attenuated, but it also has the location of the base of the cone reflected in the wall.

Figure 5.7 shows a typical "addition." A single incident cone has struck two

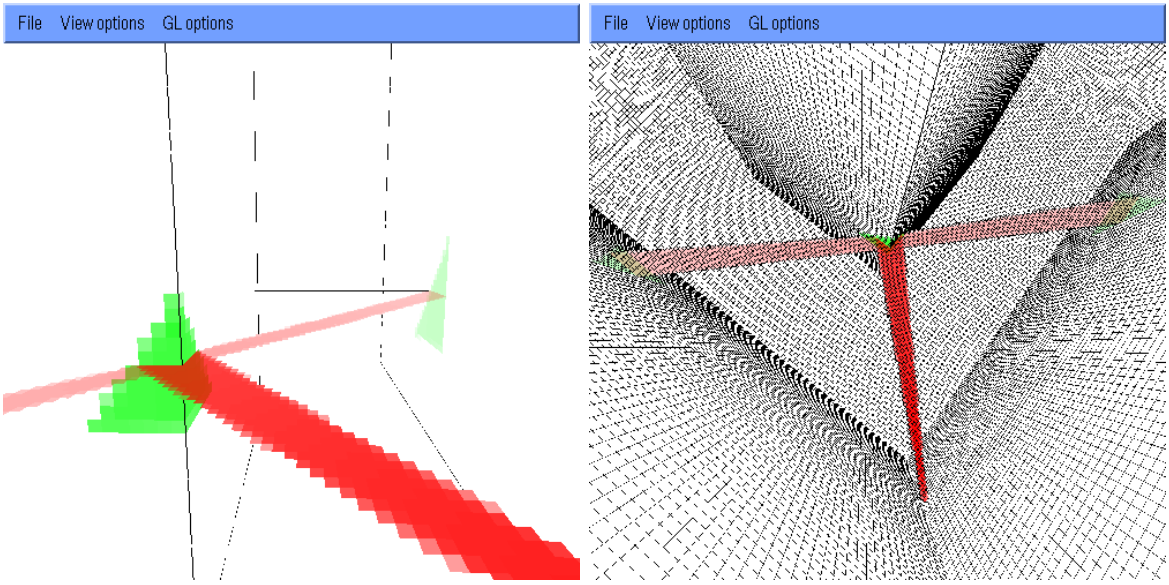


Figure 5.7: Cone addition

walls. The cone has been subdivided into two cones, one hitting each wall. These cones have been added separately to their respective walls, and have resulted in a single daughter cone each (the reflection). The red squares each represent a single square in the signal-intensity grid, and the green squares are the recorded intensities at the walls. Darker colours indicate a stronger signal.

All daughter cones are passed to new instantiations of the trace function. The parts of the cone that do not directly interact with this wall are returned in the form of a group of triangular cones. These are traced by the calling function.

5.2 Future Work

The reason for writing an implementation of these ideas was to verify their practical applicability. The facts that results show agreement with other work, and that

execution times are on the order of a minute or two show that the basic concept is practical. Having shown that, there remains more work to be done. There are inefficient areas in the code, areas in the theory that should be extended, and one notable theoretical idea that was presented but not implemented.

Under the heading of areas in the theory that should be extended, I feel the most significant topics are the incorporation of diffraction into the model, developing a more correct (and efficient) technique to avoid missing walls, completely revising the distance determination calculations, and taking incident angle into consideration when finding radar cross section coefficients. These are the subject of the remaining sections in this chapter.

The technique that was presented but not implemented is the incorporation of scattered signals. As explained in Section 2.5, surfaces in the presence of an electric field can become secondary emitters. The strength of the wall components in this respect will vary with the field intensity along the wall. To handle this, the wall is divided into sections. Each section is treated as a secondary emitter. When cones are added to the wall, the field strength at each of these points is found, and added in the same way as field strength is added to the signal-intensity grid.

After all the cones have been cast, the effects of all secondary radiators are added to the signal-intensity grid. This emission is considered to be in a uniform sphere based at the centroid of each wall section.

The signal strength along the walls is currently being gathered, but the scattered signal is not being added to the signal-intensity grid. This should be a small change to the code, but will require some investigation to explore the usefulness of the technique.

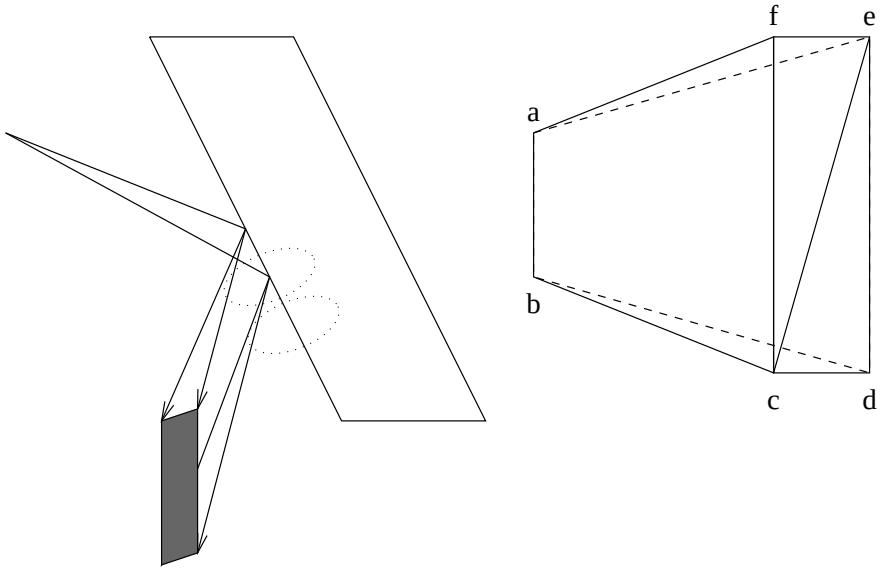


Figure 5.8: When a cone strikes an edge, the diffracted signal can no longer be described in terms of cones with a point source.

5.2.1 Diffraction

A theoretical challenge in this technique is the ability to handle diffraction[4] and other line-sources of signal. The difficulty with representing the diffracted signal pattern radiated from the edge of a wall is that it requires a series of wedge-shaped cones (as in Figure 5.8). The same pattern is required to describe signals from some types of antennas (such as "leaky coaxial"). However, as was explained in Section 5.1, cones are defined by three rays with a common source. The representation of cones could be changed to give each ray its own source, instead of having a single "base" value. This would allow the wedges to be described by two "modified cones." Using the labeling from Figure 5.8, one cone would be (ae,af,bc), and the other would be (ae,bc,bd).

[4]introduced in Section 2.6

Once the new cone style can be handled, techniques for calculating the various loss coefficients are well researched under the heading of "geometric theory of diffraction"[BB83].

5.2.2 Missing Walls

Another problem is that walls can be missed in the tracing procedure. The solutions I see to this problem apply to two parts of the interaction detection function. First, some facility must be provided for choosing a reasonable set of walls to consider. The current method uses the set of walls struck by the corners of the cone. It would also be possible (although probably unreasonable) to consider all walls in planes intersected by any of the corner rays as candidates. Somewhere between these extremes are options like clipping a perspective projection of the room (with the eyepoint at the base of the cone) on the cone, and using all walls inside that region. This would be very similar to what is done now, except that all three rays would be traced at the same time for each wall, and the 2-space location of the wall would be overlapped with the image of the triangle (clipping region). This would mean that cones would often be traced to walls that are completely occluded, but they wouldn't miss any non-occluded walls.

The second set of modifications would be to the method used for sorting the walls involved in the intersection. Right now the algorithm uses a bubble sort because there are never more than three walls to sort (only the "closest" intersections with corner rays are considered). Wall comparisons are expensive, so the complexity of the sort is an important factor in total efficiency. It is this complexity that makes it important to limit the options as much as possible in step one.

5.2.3 Distance Determination

The two techniques presented in Chapter 4 both have drawbacks. The linear interpolation method provides a good approximation, but inaccuracy is unavoidable, and the exact method is slow. Using the fact that all transformations are affine (as in the exact method discussed) also allows the conclusion that the distances can always be interpolated by an ellipsoid.[5] Since ellipsoids have only six degrees of freedom, it seems likely that there is a way to cast extra rays or calculate specific distances, to the centre of each edge of the triangle for example, such that the ellipsoid is always constrained by these values. If its equation could be found easily, the square root calculation could be removed from the exact method. This would increase its speed, but would still give exact distances.

5.2.4 Incident Angles and Coefficients

The current calculation of the reflection and transmission coefficients ignores the angle of incidence. This should be addressed, but will require careful implementation, since different parts of the incident cone are striking the surface at different angles. It will probably be necessary to allow the cone to be split if there is significant variation in attenuation over the affected surface. The relation of incident angle to reflection coefficient is discussed by Shaeffer [KST85, p.74].

Some other changes would also be beneficial, such as distributing this problem across multiple processors. Ray tracing distributes well, and an almost linear speedup can be achieved. The results of testing what was actually implemented, however, will now be presented in Chapter 6. Analysis and interpretation of these results is given there, leading to a final summary of conclusions in Chapter 7.

[5] Spheres and ellipsoids always map to ellipsoids through affine transformations.

Chapter 6

Performance Results

This chapter attempts to answer the first of the the two questions presented in Chapter 1, namely "Does this general class of technique (cone tracing) constitute a reasonable method for signal strength prediction?" Chapters 4 and 5 presented the various design decisions I feel are important for implementing an efficient ray tracing solution to the problem of site-specific signal propagation prediction, but these techniques are useful only if they produce correct results. I demonstrate the correctness of the results by comparing them with those of Lecours et al [LGBC93], and show the practicality of cone tracing by presenting actual execution durations.

6.1 Agreement With Theory

Lecours et al [LGBC93] developed a ray tracing-based simulator, and published its behaviour in both real and imaginary situations. The real situations demonstrated good agreement, and that was presented as an argument that the analysis of imaginary situations also reflected a realistic approximation of the actual signal

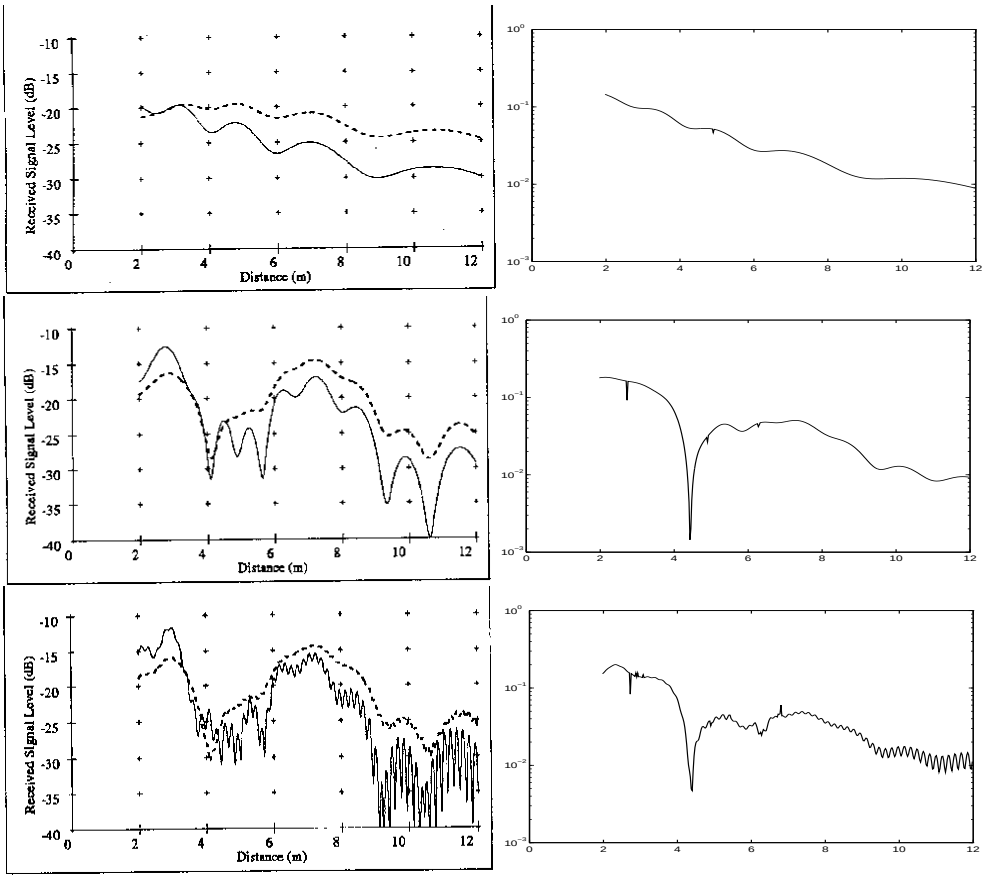


Figure 6.1: Cone tracing vs. Lecours et al Left side is from [LGBC93] (reprinted with permission from the author), right is from cone tracing. Top: floor only; Middle: floor, ceiling, side walls; Bottom: fully enclosed.

behaviour.

Figure 6.1 compares the results from Lecours et al to mine for an imaginary room.[1] These are the signal strength predictions for an isotropic radiator one metre from the "front" wall, and one metre below the ceiling. The graphs show the signal strength 2 metres above the floor, along a line one metre from the left

[1]The data file used to describe the room in the cone tracing simulation is reproduced in Appendix A.

wall. Both sets of results show an area of poor reception in the region from four to six metres along the side wall. This develops when the ceiling and side walls are added.

There are two major differences in these figures. First, there is an apparent 10 dB shift (at two metres, the Lecours et al room has an attenuation near -20 dB, and mine has one of about -10 dB). This is just from the definition of signal strength at 1 metre from the antenna, and is strictly a calibration issue. The shape of these graphs is much more significant than the point values.

Second, the effect of the ceiling reflection near the transmitter is smaller in my model. This is apparent in the top pair if graphs, in which the first peak of the Lecours et al model (at 3 metres) is about the same intensity as the signal at 2 metres, while mine shows a drop of one or two decibels. The same effect is obvious near the 12 metre point on the bottom pair of graphs, in which my model shows significantly less short term fading caused by the reflection from the back wall. These are both examples of my model underestimating the strength of a reflected signal resulting from an incoming signal at a steep (near 90°) incident angle. These differences can be explained by the failure of the current implementation of cone tracing to consider angle of incidence when finding reflection coefficients. The values I used were the coefficients for a 45 degree angle of incidence. This is not always a good approximation (see Figure 6.2).

If these differences are allowed for, one can see the general agreement on large-scale behaviour. Both predict an area of unusually poor signal strength in the region from four to six metres, a more stable signal from six to eight, and a reasonably steady decline at more than eight metres.

Theoretically, this result is not a surprise. Although blocks of signal are con-

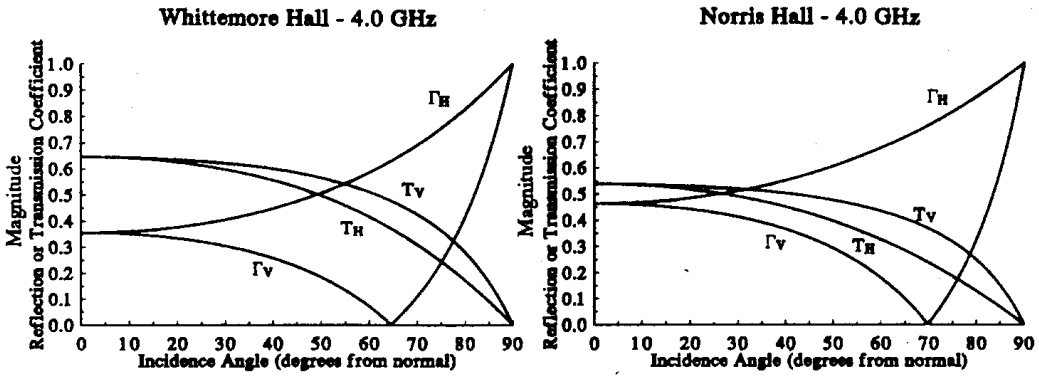


Figure 6.2: Transmission (T) and reflection (Γ) coefficients vs. angle of incidence in two buildings [SR94]. (Subscripts indicate the polarization of the incoming signal) Note the large, non-linearly varying, in some cases non-differentiable, effect of incident angle on loss coefficients.

sidered, instead of infinitely thin components, the same reflections are used, so the total interference should be similar. What this experiment proves, however, is that the cone tracing technique can be applied in practice. The graphs in Figure 6.1 were generated by a post-analysis loop through the signal strength matrix, and represent only one strip of the room. In fact, by the time they were known, the analysis for the whole room was complete. This particular analysis took thirty seconds on a lightly loaded RS/6000.

6.2 Execution Time

Timings were done for four models. Room 1 was the Lecours et al room. Room 2 was for the same room with a wall added 7.0 metres from the back wall (transmission coefficient 0.1, reflection 0.38). Room 3 also had a similar wall 2 metres from the

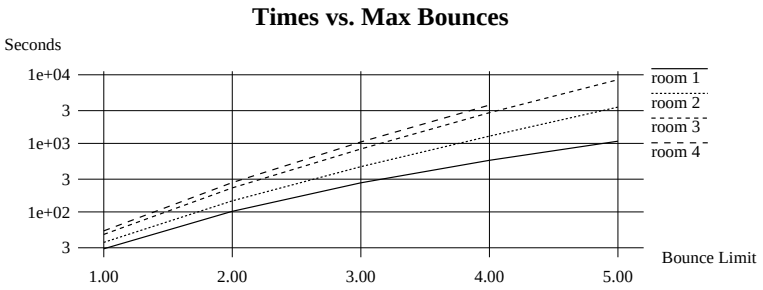


Figure 6.3: Execution times for four rooms

left wall, running from the back wall to the wall added for Room 2. Room 4 added a wall that extended the one added in Room 3 all the way to the front wall.[2] Overviews of the room layouts, along with the one-bounce signal strength estimates are given in Appendix A. Each room was analysed for various limits on the number of bounces permitted. Ten trials were used for each bounce limit in each room, and the averages were used. The user time elapsed was roughly exponential with respect to the bounce limit (Figure 6.3). Exponential growth seems natural, since each cone gives birth to some group of daughter cones, and the average number of daughter cones from an incident cone should be independent of the generation numbers of the cones involved.

The results (shown in Figure 6.4) demonstrate that, while the time cost of each step increases, the effect on results decreases. This also makes sense, since cones usually become weaker when they are reflected, so including cones that have been reflected several times has little effect.

Combining these results suggests that setting a low limit on the number of bounces permitted will greatly benefit overall efficiency.

[2]A new wall was added because extending the size of the wall added in Room 3 would cause it to cross the wall added in Room 2, and walls that cross can cause the occlusion algorithm to give incorrect results.

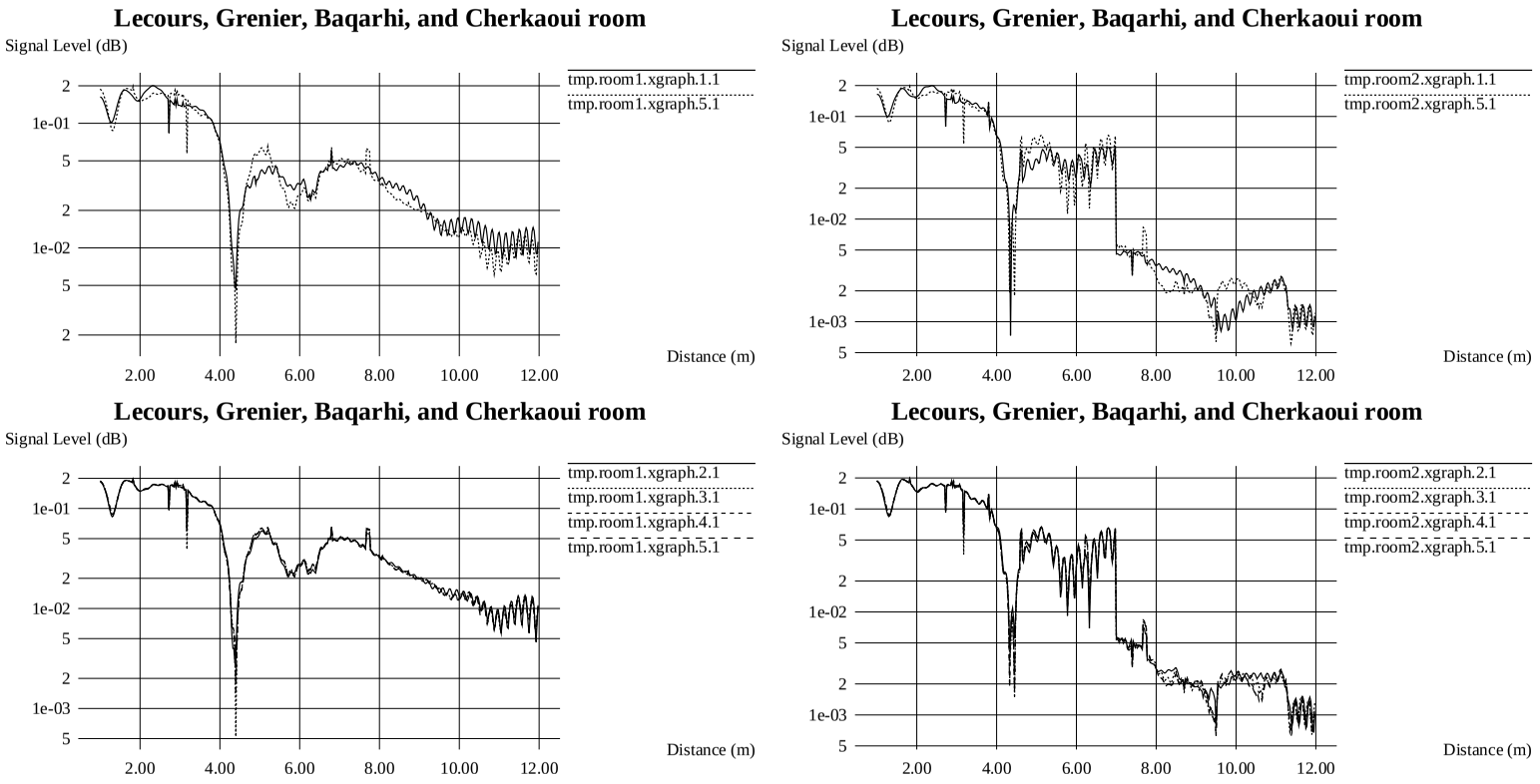


Figure 6.4: How bounce limits affect results: room one is shown on the left, and room 2 on the right. The top row compares 1 vs. 5, and the bottom shows 2, 3, 4, and 5. Notice that there is very little qualitative difference between limiting the results to two, and limiting them to five. Compare this with the execution times shown in Figure 6.3.

6.3 Profile Results

Profiling the code demonstrated some interesting information. On a typical run (using the exact technique) cosine, sine, and square root calculation used 37% of the time spent simulating the signal. This includes only calls due to signal phase calculations, and supports the argument for an adaptive algorithm, as presented in Section 4.2.

Another useful number, though not surprising, is that the fill routine used 58% of the total simulation time. This suggests that improvements to that routine could significantly improve overall performance.

These two areas are closely related, since it is the fill routine that usually calculates the phase of signals at particular points, and could most reasonably be improved (or replaced) together.

Chapter 7

Conclusions

In 1991, McKown and Hamilton [MH91] wrote

> "Our major requirement was that our code execute fast enough to be useful for generating two-dimensional maps. This ruled out the most simple approach to ray tracing, which is to spew out a cone or pin cushion of rays from the transmitter, trace them until they either hit the receiver or become attenuated beyond some threshold, add together the fraction that reach the receiver, and increment the receiver position and repeat the process to build up the desired map."

The cone tracing technique that I have presented here makes this simpler approach to the problem a practical one.

This thesis has shown an efficient application of cone tracing, which produces useful approximations of signal behaviour in specific indoor environments. Using the technique described avoids the mathematical complexities of approximating a solid wavefront with a collection of rays, the redundancies of analysing the same

room for multiple receiver locations, and the inefficiencies of choosing arbitrary initial signal resolutions. It does so by using cones instead of rays, recording signal intensity for the entire room in one pass, and dynamically subdividing cones that become too big.

A sample implementation was presented, and usage profiles show some areas that need improvement. For example, the fill algorithm should be rewritten. Other useful work would be to experiment with the diffuse signals. The values are already being collected, but they are not added to the total. This should be a very small change to the code, but it would be interesting to see how much difference this component makes.

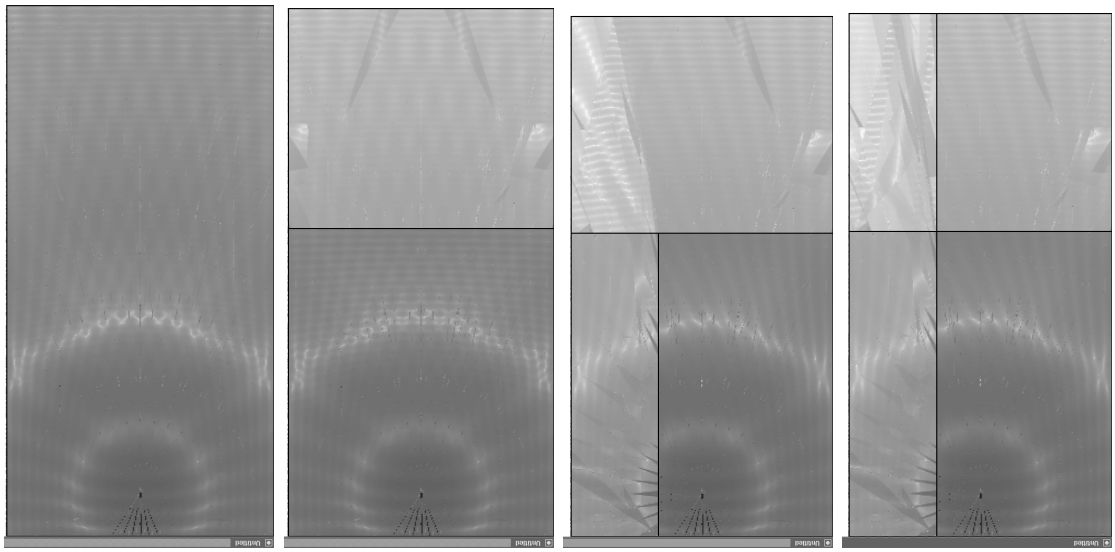
From a more theoretical perspective, future work was recommended in the handling of diffracted signals, and in the incorporation of angle of incidence into the strength of daughter cones. Some ideas were also briefly presented that might lead to the development of a better technique for not missing walls.

Exploring the applications of this cone tracing implementation has also been left incomplete. Collecting information at several altitudes, for example, could be done by locating walls with 0.0 reflection, and 1.0 transmission horizontally inside particular rooms. This might be useful to some users. Using the results to automatically tune transmitter location may also be possible by adding another layer to the program structure. It may also be possible to handle non-polygonal regular geometric figures by using a slightly more general routine for locating the "virtual source" resulting from a reflection or transmission. I see these extensions as both possible and useful; the specific ideas and implementation techniques discussed here form only one more step towards a more full understanding of how best to represent the complicated systems found in radio signal propagation analysis.

Appendix A

Test Run Layouts

The following rooms are discussed in Chapter 6.



From left to right, these are rooms one, two, three, and four. The figures show an overview of each room, in which darker areas indicate regions of better signal strength.

The "long line" tolerance described in Section 5.1.3 is 900. The antenna is in the centre of the back wall, 10 units away, and at an altitude of 30 (the nominal

scale is 10 units = 1 metre). The wavelength is 3.0 (30 cm), which corresponds to 900 MHz. The signal-intensity grid was at resolution = 10, i.e. 10 grid squares for each unit, therefore the data squares were 1 cm on a side. The walls were at resolution 1, which means that they used 10 cm squares. These values were chosen to agree with those analysed by Lecours et al [LGBC93].

The files which define the remainder of the room have the following format:

Line 1: room size X, room size Y, room resolution, volume altitude, bounce limit

Line 2-n: base location $< x, y, z >$, 'X' basis vector $< x, y, z >$, 'Y' basis vector $< x, y, z >$, top corner (local coordinates) $< x, y >$, transmission coefficient, reflection coefficient

The first line describes the room, and the remaining lines describe one wall each.

The actual files used follow:

Room 1:

```
60, 120, 4, 12, 1
0,0,0,    1,0,0, 0,1,0,  60,120, 0.0,0.2
0,0,35,   1,0,0, 0,1,0,  60,120, 0.0,1.0
0,0,0,    0,1,0, 0,0,1, 120, 35, 0.0,0.26
60,0,0,   0,1,0, 0,0,1, 120, 35, 0.0,0.26
0,0,0,    1,0,0, 0,0,1,  60, 35, 0.0,0.38
0,120,0,  1,0,0, 0,0,1,  60, 35, 0.0,0.38
```

Room 2:

```
60, 120, 4, 12, 1
0,0,0,    1,0,0, 0,1,0,  60,120, 0.0,0.2
0,0,35,   1,0,0, 0,1,0,  60,120, 0.0,1.0
0,0,0,    0,1,0, 0,0,1, 120, 35, 0.0,0.26
60,0,0,   0,1,0, 0,0,1, 120, 35, 0.0,0.26
0,0,0,    1,0,0, 0,0,1,  60, 35, 0.0,0.38
0,120,0,  1,0,0, 0,0,1,  60, 35, 0.0,0.38
0,70,0,   1,0,0, 0,0,1,  60, 35, 0.1,0.38
```

Room 3:

```
60, 120, 4, 12, 1
0,0,0,    1,0,0, 0,1,0,  60,120, 0.0,0.2
0,0,35,   1,0,0, 0,1,0,  60,120, 0.0,1.0
0,0,0,    0,1,0, 0,0,1, 120, 35, 0.0,0.26
60,0,0,   0,1,0, 0,0,1, 120, 35, 0.0,0.26
0,0,0,    1,0,0, 0,0,1,  60, 35, 0.0,0.38
0,120,0,  1,0,0, 0,0,1,  60, 35, 0.0,0.38
0,70,0,   1,0,0, 0,0,1,  60, 35, 0.1,0.05
40,0,0,   0,1,0, 0,0,1,  70, 35, 0.1,0.05
```

Room 4:

```
60, 120, 4, 12, 1
0,0,0,    1,0,0, 0,1,0,  60,120, 0.0,0.2
0,0,35,   1,0,0, 0,1,0,  60,120, 0.0,1.0
```

```
0,0,0,    0,1,0, 0,0,1, 120, 35, 0.0,0.26
60,0,0,   0,1,0, 0,0,1, 120, 35, 0.0,0.26
0,0,0,    1,0,0, 0,0,1,  60, 35, 0.0,0.38
0,120,0,  1,0,0, 0,0,1,  60, 35, 0.0,0.38
0,70,0,   1,0,0, 0,0,1,  60, 35, 0.1,0.05
40,0,0,   0,1,0, 0,0,1,  70, 35, 0.1,0.05
40,70,0,  0,1,0, 0,0,1,  50, 35, 0.1,0.05
```

Appendix B

Derivation of Equation 4.2

To keep this derivation short, only the major steps are shown. This is done using the "Maple V" symbolic algebra package. The variable ℓ used in the text is represented here as p. The lack of script characters in Maple makes it difficult to differentiate between ℓ and 1.

```
> # p is an arbitrary variable on interval [0,1]
  m := sqrt(((1-p)*d1 + p*d2*cos(alpha))^2 + (p*d2*sin(alpha))^2);
  n := (1-p)*d1 + p*d2;
```

$$m := \sqrt{((1 - p)\, d1 + p\, d2 \cos(\alpha))^2 + p^2\, d2^2 \sin(\alpha)^2}$$

$$n := (1 - p)\, d1 + p\, d2$$

```
> # find the value of p which maximizes the difference between m and n
  solve(diff(subs(d1=d2,n-m),p)=0,p);
```

$$-\frac{-1 + \cos(\alpha)}{1 - 2\cos(\alpha) + \cos(\alpha)^2 + \sin(\alpha)^2}$$

```
> # remove the common factor and use cos^2 + sin^2 = 1
  simplify(");
```

$$\frac{1}{2}$$

Bibliography

[Ake88] D. Akerberg. Properties of a TDMA pico cellular office communication system. In *IEEE GLOBECOM*, pages 649–653, Hollywood, Florida, U.S.A., December 1988.

[Ale83] S. E. Alexander. Characterizing buildings for propagation at 900 MHz. *Electronics Letters*, 19(18):860, September 1983.

[Ama84] John Amanatides. Ray tracing with cones. In *Computer Graphics (SIGGRAPH '84 Proceedings)*, pages 129–135, July 1984.

[Arv86] James Arvo. Backward ray tracing. In *SIGGRAPH '86 (Course Notes #12)*, 1986.

[BB83] W. D. Burnside and K. W. Burgener. High frequency scattering by a thin lossless dielectric slab. *IEEE Transactions on Antennas and Propagation*, 31(1):104–110, January 1983.

[CMN83] D. C. Cox, R. R. Murray, and A. W. Norris. Measurements of 800 MHz radio transmission into buildings with metallic walls. *The Bell System Technical Journal*, 62(9):2695–2717, November 1983.

[CMN84] D. C. Cox, R. R. Murray, and A. W. Norris. 800 MHz attenuation measured in and around suburban houses. *The Bell System Technical Journal*, 63(6):921–954, July-August 1984.

[CsECWG88] Michael Cohen, shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. In *Computer Graphics (SIGGRAPH '88 Proceedings)*, pages 75–84, August 1988.

[Fre91] Thomas A. Freeburg. Enabling technologies for wireless in-building network communications – Four technical challenges, Four solutions. *IEEE Communications Magazine*, pages 58–64, April 1991.

[FvDFH91] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics (second edition).* Addison-Wesley, 1991.

[Has93] Hamayoun Hashemi. The indoor radio propagation channel. *Proceedings of the IEEE*, 81(7), July 1993.

[HBD+92] W. Honcharenko, H. L. Bertoni, J. L. Dailing, J. Qian, and H. D. Yee. Mechanisms governing UHF propagation on single floors in modern office buildings. *IEEE Transactions on Vehicular Technology*, 41(4):496–504, November 1992.

[HH84] Paul S. Heckbert and Pat Hanrahan. Beam tracing polygonal objects. In *SIGGRAPH '84*, pages 119–127, 1984.

[HPL92] T. Holt, K. Pahlavan, and J. F. Lee. A graphical indoor radio channel simulator using 2D ray tracing. In *IEEE International Symposium on PIMRC*, pages 411–416, October 1992.

[Jon79] D. S. Jones. *Methods in Electromagnetic Wave Propagation.* Clarendon Press, Oxford, 1979.

[KM90] J. Keenan and A. Motley. Radio coverage in buildings. *British Telecom Technology Journal*, 8(1):19–24, January 1990.

[KST85] Eugene F. Knott, John F. Shaeffer, and Michael T. Tuley. *Radar cross section: Its prediction, measurement, and reduction.* Artech House radar library. Artech House, Dedham, MA, USA, 1985.

[KUG93] Peter Kreuzgruber, Peter Unterberger, and Rainer Gahleitner. A ray splitting model for indoor radio propagation associated with complex geometries. In *43nd annual IEEE Vehicular Technology Conference*, pages 227–230, 1993.

[Lee82] W. C. Lee. *Mobile Communications Engineering.* McGraw Hill, New York, 1982.

[LGBC93] M. Lecours, D. Grenier, M. Baqarhi, and S. Cherkaoui. Measurements and simulation of received signals in rooms and corridors at 900 MHz and in the 20-60 GHz band. In *43nd annual IEEE Vehicular Technology Conference*, pages 871–874, 1993.

[Max66] J. Clerk Maxwell. On the elementary relations between electrical measurements. *The Philosophical Magazine*, 24:436–460, 1866.

[MH91] John W. McKown and R. Lee Hamilton Jr. Ray tracing as a design tool for radio networks. *IEEE Network Magazine*, pages 27–30, November 1991.

[MK88] A. Motley and J. Keenan. Personal communication radio coverage in buildings at 900 MHz and 1700 MHz. *Electronics Letters*, 24(12):763–764, June 1988.

[MP92] Michel Mouly and Marie-Bernadette Pautet. *The GSM System for Mobile Communications*. published by the authors, 1992.

[PLL92] Keith D. Paulsen, Daniel R. Lynch, and Weiping Liu. Conjugate direction methods for helmholz problems with complex-valued wavenumbers. *International Journal for Numberical Methods in Engineering*, 35:601–622, 1992.

[Ric44] S. O. Rice. Mathematical analysis of random noise. *Bell System Technical Journal*, 23 & 24:287–332 & 46–156, 1944. Continued from vol. 23 to vol. 24.

[RRB92] Thomas A. Russell, Theodore S. Rappaport, and Charles W. Bostian. Use of a building database in prediction of three-dimensional diffraction. In *42nd annual IEEE Vehicular Technology Conference*, volume 2, pages 943–946, Denver, Colorado, USA, 1992.

[Ser86] Raymond A. Serway. *Physics for scientists and engineers with modern physics (second edition)*. CBS College Publishing, 1986.

[SIR92] Kurt R. Schaubach, Nathaniel J. Davis IV, and Theodore S. Rappaport. A ray tracing method for predicting path loss and delay spread in microcellular environments. In *42nd annual IEEE Vehicular Technology Conference*, volume 2, pages 932–935, Denver, Colorado, USA, 1992.

[SR92] Scott Y. Seidel and Theodore S. Rappaport. A ray tracing technique to predict path loss and delay spread inside buildings. In *Globecom 92*, pages 649–653, 1992.

[SR94] Scott Y. Seidel and Theodore S. Rappaport. Site-specific propagation prediction for wireless in-building personal communication system design. *IEEE Transactions on Vehicular Technology*, 43(4):879–891, November 1994.

[THF92] V. R. M. Thyagarajan, H. M. Hafez, and D. D. Falconer. Traffic study of TDMA-TDD in millimeter wave indoor wireless communications. In *IEEE Symposium on PIMRC*, pages 566–570, Boston, Mas., USA, October 1992.

[Val93] Reinaldo A. Valenzuela. A ray tracing approach to predicting indoor wireless transmission. In *43nd annual IEEE Vehicular Technology Conference*, pages 214–218, 1993.

Glossary

aliasing An undesirable side effect of a computer graphics technique

bulge difference between the approximate distance to a cone's interior point, arrived at by linear interpolation between the corners, and the actual distance

cone tracing an alternative to ray tracing which follows expanding volumes, rather than infinitely thin rays

clipping algorithm an algorithm which receives a representation of a graphic object and a viewport, and produces the representation of the portion of the original object which appears inside the viewport

delay spread the length of time after the first receipt of a portion of a signal during which that same portion might still arrive with significant strength via a different path.

external cone a cone which, after intersection calculations have been completed, has struck none of the candidate walls

footprint area of useful signal coverage

GO Geometric Optics

GTD Geometric Theory of Diffraction

phase jitter the range of expected phase variations from the norm

RF Radio Frequency

signal intensity grid the array used to store signal strength for the interior of a scene

solid angle a dimensionless quantity representing the ratio of surface area of the spherical section spanned by a particular angle to the square of the radius (units of measurement are steradians)

viewport a bounding polygon (usually in world coordinates) which limits the area being displayed

wall space the two dimensional space defined by $\hat{x}'$ and $\hat{y}'$ for a wall segment. Used during intersection calculations