A beam tracing algorithm for prediction of indoor radio propagation

Steven Fortune
AT&T Bell Laboratories
Murray Hill, NJ 07974
sjf@research.att.com

February 27, 1996

Abstract

We describe a beam tracing algorithm that simulates radio propagation inside a building. With a triangulation-based spatial data structure, the algorithm is fast enough to provide propagation simulations for large buildings in a few minutes of computing time.

1 Introduction

Wireless communication networks using radio-frequency transceivers are important for many indoor applications. The design of such networks requires accurate prediction of radio propagation inside a building.

Radio propagation can be modeled fairly simply, using geometric optics. A building is a list of walls, each a polygon labeled with construction material. Radio propagates along a straight line, with power diminishing according to an inverse-square law. At a wall, part of the radio power passes through and part is reflected specularly, with proportions depending upon material and incidence angle. Diffraction is ignored in this model, as well as scattering at a wall or because of furniture and other clutter. To determine coverage, power is sampled at the lattice points of a regular grid.

Naive simulation of propagation is computationally challenging. In a large building there may be thousands of possible propagation paths between base transmitter and sample point, though only a few contribute significant power, and received power may need to be simulated at thousands of different sample points. The dynamic range of the simulation—the ratio between minimum and maximum received power—is 10^6 or more. The simulation is most useful if it is accurate over the entire dynamic range.

This paper describes a beam tracing algorithm that can be used to predict radio propagation. Each wall can be viewed as a partially reflecting mirror that illuminates a polyhedral cone, using reflected power from the base. If a sample point lies in an illumination cone, then there is a propagation path to the sample point that reflects off the corresponding wall; the actual path power may be reduced by the shadowing of intermediate walls. All propagation paths, up to some fixed number of reflections, can be found with such illumination cones.

The algorithm uses a spatial data structure based on triangulations. This data structure answers ray-shooting queries in constant time (as measured experimentally) and allows efficient detection of all environment obstacles that lie in a polyhedral query cone. Implementation experiments are reported below.

This paper does not address the accuracy of the propagation model itself. Many factors might affect prediction accuracy: the presence of furniture, inaccuracies in the building model, the modeling of reflection and transmission losses, other radio effects such as scattering or diffraction, etc. The literature contains some comparisons between predictions and measurements, for this and similar models [4, 10].

The algorithm in this paper have been incorporated into a wireless system engineering tool called WISE [1]. WISE includes a powerful graphical user interface that allows system engineers to choose base placements interactively, visualize propagation predictions, and optimize base placement automatically.

Other work. Beam tracing is a standard technique in computer graphics rendering [11]; its use for radio propagation has been independently proposed by Takahashi *et al* [9]. Propagation can also be predicted using ray tracing, i.e. by following paths in a large but discrete set of propagation directions[4, 5, 7, 10]. Beam tracing is competitive with ray tracing in performance and has fewer sampling artifacts; the full version of this paper [2] gives a detailed comparison.

Other practical spatial data structures can be used to answer ray-shooting and related queries. Held *et al* [3] decompose space into tetrahedra; Rajkumar *et al* [7] use binary space partition trees.

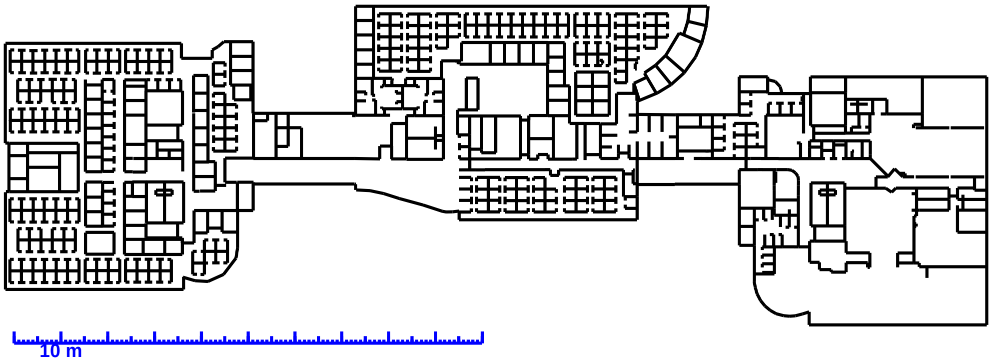

Figure 1: Plan view of AT&T Mt. Airy first floor; this model has about 2315 walls.

2 Radio background

The books by Parsons[6] and by Ramo *et al*[8] are general references for radio propagation.

The environment consists of a set of *walls*, each of which is a polygon with a small number of edges, e.g., a rectangle. Each wall has a type that indicates its construction material and hence radio properties. A typical environment might have between 100 and 10000 walls (figure 1).

A *propagation path* is a directed polygonal chain such that at each internal vertex, the path is generated by reflection off a wall using Snell's law, that is, the direction along the path after the vertex is the reflection in the plane of the wall of the direction before the vertex. A propagation path may pass through a wall.

A propagation path p has an associated *path loss*

$$l_p = \frac{\prod_i \rho_i \prod_j \tau_j}{|p|^2} \tag{1}$$

where $|p|$ is the length of the path in meters; ρ_i is the ith *reflection coefficient*, i.e., loss due to reflection; and τ_j is the jth *transmission coefficient*, i.e., loss due to a wall intersection that is not a reflection. Measured in dB, path loss is $10 \log_{10} l_p$.

The (free-space) received power at one meter is given by

$$\beta_{1m} = P \left(\frac{\lambda}{4\pi} \right)^2$$

where P is the power input to the transmitter and λ is the wavelength in meters. The *received power* at a sample point is

$$\beta = \beta_{1m} \sum_p l_p \tag{2}$$

where the sum runs over all propagation paths from base transmitter to sample point. Measured in dBm, received power is $10 \log_{10} \beta$, assuming transmitter power is measured in milliwatts (mw).

Typical transmitter power is 34 mw; at 2 Ghz this yields a free-space received power of about -23 dBm at one meter. A received power of -70 dBm is typically required for reliable communication, implying a free-space coverage radius of about 220 meters. It is appropriate to model paths with even lower power, say down to -90 dBm, since several paths could add together to meet the -70 dBm threshold, and since it is desirable to model interference between transmitters operating on the same frequency.

A *coverage calculation* determines received power at a set of sample points. Typically there are infinitely many propagation paths to a sample point. Hence a coverage calculation is usually an approximation, with propagation paths restricted to those of relatively high power or with a limited number of reflections.

Theoretical transmission and reflection coefficients can be determined using a multilayer dielectric model [8] derived from estimated properties of construction materials. For example, a sheetrock wall might be modeled as a half inch of plaster, 3.5 inches of air, and a half inch of plaster. For a fixed wall material and wall orientation, the coefficients depend upon the angle of incidence of path to wall and can be determined to adequate accuracy with table lookup indexed by discretized angle. Figure 2 plots the coefficients for sheetrock and 'wood over concrete' as a function of incident angle; both walls are vertical. For most walls, the transmission coefficient is maximized if the path is perpendicular to the wall; the reflection coefficient approaches 1 if the path

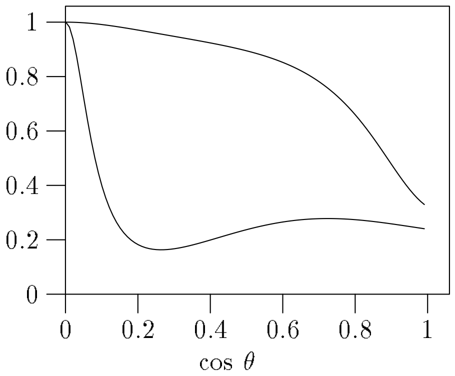

Figure 2: Theoretical reflection coefficient as a function of incident angle: sheetrock (top) and wood over concrete (bottom). Transmission coefficient is 1 minus reflection coefficient.

is almost parallel to the wall. In practice, theoretical reflection and transmission coefficients may need to be altered to model losses due to scattering, construction inhomogeneities, furniture, etc.

3 The beam tracing algorithm

Fix the position of the base transmitter. The *reflection cone* $C(w)$ for a wall w consists of all points reachable by a single-reflection path off wall w. Let v be the reflected image of the base in the plane of the wall; v is a *virtual transmitter*. It is immediate that a sample point is reachable by a single reflection off wall w exactly if the line segment from v to the sample point intersects w at the reflecting point. Thus the reflection cone is a truncated polyhedral cone. It has apex v; each plane through v and one of the edges of w bounds the cone; the cone is truncated by the plane of w, deleting the portion of the cone from v to w. See figure 3.

In general, for a sequence $w_1, \ldots, w_k$ of walls, the *reflection cone* $C(w_1, \ldots, w_k)$ is the set of all points reachable by paths that reflect off walls $w_1, \ldots, w_k$ in order. It has apex the *iterated virtual transmitter* $\hat{v}$, obtained by reflecting the transmitter successively in the planes of w_1, …, w_k. $C(w_1, \ldots, w_k)$ is bounded by each plane through $\hat{v}$ and an edge of the polygon $w_k \cap C(w_1, \ldots, w_{k-1})$. As before, $C(w_1, \ldots, w_k)$ is truncated by the plane of w_k.

The beam tracing algorithm assumes an *a priori* bound on the number r of reflections. It can be given simply as

for each reflection cone C
  for each sample point p inside C
    compute power of path to p.

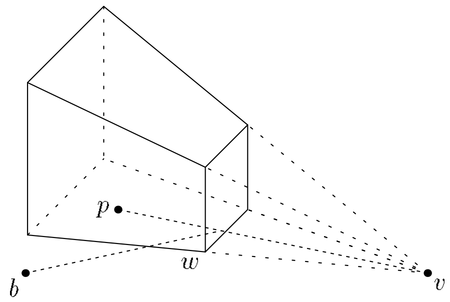

Figure 3: Reflection cone (outlined in solid lines). b is base, v is virtual transmitter. Reflecting path from b to p intersects wall w at same point as line segment vp.

The reflection cones can be generated by a simple recursive process: there is a one-reflection cone $C(w)$ for each wall w, and a cone $C(w_1, \ldots, w_k)$ for each wall intersecting $C(w_1, \ldots, w_{k-1})$. Section 4 describes an efficient method that enumerates the walls inside a reflection cone.

In the worst case, the number of nonempty reflection cones grow as n^r, where n is the number of walls. More typically a reflection cone intersects about $n^{1/2}$ walls, so the number of cones grows as $n^{(r+1)/2}$. See section 5 for experimental data.

Given sample point p lying in a reflection cone, it is easy to find the corresponding propagation path to p. The power computation using formulas (1) and (2) requires the discovery of the walls that intersect the propagation path. Section 4 describes a spatial data structure that can be used to trace a propagation path, finding the intersected walls in time that is approximately constant per wall.

Even with the spatial data structure, path tracing is the dominant cost of the algorithm. Consequently, it is useful to avoid a path power computation if it will not contribute significant power to a sample point; here significant power is defined as a fraction, say .01, of the power already accumulated at the sample point.

Let w be the last wall in the reflection sequence of a reflection cone C, and let m be an interior point of $C \cap W$, say the centroid of the vertices of $C \cap W$. Cone C is assigned an *estimated power* β_C, defined as the power from the base to m (including the final reflection coefficient off wall w).

The value β_C can be used to filter path power computations, in two ways. First, if β_C is less than an absolute threshold, say -90 dBm, then C is ignored (as are any reflection cones produced by walls intersecting C). Second, suppose β_C is above threshold and sample point p lies inside C. An upper bound estimate of the path power to p is $\beta_C(d_m/d_p)^2$, where d_m and d_p are propagation path distances to m and p, respectively. If this upper bound cannot add significant power to p, then the path is ignored. Otherwise the path is traced and path power computed; notice that β_C does not affect the actual path power computation.

It is possible that a filtered-out path in fact contributes significant power to a sample point. This can happen if there is a wide variation on the path power to points of $C \cap w$, and β_C happens to be a low estimate. The probability of such erroneously-filtered paths is minimized by restricting filtering to narrow cones, where a cone C is *narrow* if the maximum difference between any two direction vectors from the virtual transmitter of C to points in C is small, say ten degrees.

The full paper [2] discusses an algorithm that sweeps through each reflection cone, maintaining the 'shadow partition' induced by wall transmission losses. This algorithm eliminates the need to trace each propagation path to a sample point. However it is more expensive to maintain the shadow partition (roughly n^3 per reflection cone) than to do the path tracing.

4 Data structures

We describe a triangulation-based spatial data structure that can be used to answer two geometric queries: report the walls intersected by a query line segment, and report all walls inside a convex polyhedral query cone. To use the data structure, each wall polygon must either be horizontal, of arbitrary shape, or a vertical rectangle, i.e., with two sides parallel to the z-axis and top and bottom horizontal. Most building walls can be modeled with polygons of this type, though e.g. cathedral ceilings cannot.

The triangulation data structure. A *distinguished z-value* is a z-coordinate of a wall vertex. The data structure consists of a set of triangulations. For each distinguished z-value, there is a triangulation of the edges of the horizontal walls at that z-value. Any triangle interior to a horizontal wall is marked with the wall that contains it. For each open interval between two adjacent distinguished z-values, there is a triangulation of the horizontal cross-section; notice that the horizontal cross-section consists of an edge for each vertical wall spanning the z-interval, and that the cross-section is constant over the interval. Each edge of the triangulation is marked with the wall that contains it, if any.

The space required to store the triangulation data structure is linear in the number of horizontal walls plus $\sum n_i$, where n_i is the number of vertical walls that span the ith z-interval. In the worst case $\sum n_i$ is quadratic in the total number of vertical walls. However, most building models have only a few distinguished z-values per building floor, perhaps one for the floor, one for the false ceiling, and one for the true ceiling; and most walls span at most a single floor. Hence in practice, the sum is linear in the number of walls.

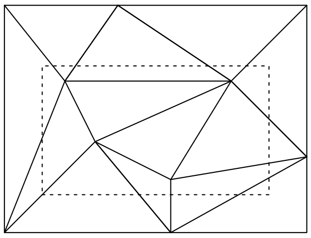

Figure 4: Query polygon (dashed) separates triangulation into inside and outside; graph search can report triangles and edges inside.

Line segment queries. To report the walls intersected by a query line segment, first split the line segment into subsegments so that each subsegment lies within a single z interval. The vertical walls intersected by a single subsegment can be determined using a walk in the corresponding triangulation, using the xy-projection of the subsegment. The horizontal wall crossed at each distinguished z-value, if any, can be discovered by a point location in the corresponding triangulation.

Polyhedral cone queries. The following strategy reports all walls intersected by a convex polyhedral query cone; it is repeated for each distinguished z-value or interval between z-values.

The query cone intersects the horizontal plane through a distinguished z-value in a convex polygon. The algorithm walks each edge of the polygon in turn, marking the triangulation edges intersected by polygon edges. See figure 4. The edges partially or fully inside the polygon can then be enumerated by a standard graph search, starting with the marked edges. Each triangle incident to an enumerated edge is examined to see if it is marked with a horizontal wall.

The approach is similar for an interval of z-values. Consider the subset of the query cone that lies within the z-interval. The orthogonal projection onto the xy-plane of this subset is a convex polygon. The edges of this polygon are used for the walk, as before. In this case, the graph search examines edges of the triangulation for intersected walls, rather than triangles.

Performance. For a line segment query, the number of triangulation edges traversed per reported wall varied between 2.5 and 6, depending upon the uniformity

Building	Floors	Walls	size (m)	Sample points
Crawford	1	81	15×130	418
Hynes	4	1436	130×170	5240
Indiana	2	278	120×250	7060
Middletown	1	1367	200×200	3651
Mt Airy	3	2316	60×220	2449
Prudential	2	1175	65×85	1366

Figure 5: Sample buildings.

	reflections				
Building	1	2	3	4	5
Crawford	:01	:04	:10	:27	:51
Hynes	:12	:53	2:39	6:18	14:08
Indiana	:11	:37	1:44	4:31	10:24
Middletown	:14	1:09	4:04	13:09	39:20
Mt Airy	:11	1:40	8:03	32:52	103:03
Prudential	:07	:44	4:25	19:46	66:31

Figure 6: Running time of beam tracing, mm:ss.

of building walls. For the polyhedral cone query, the number of triangulation edges visited per reported wall usually varied between 6 and 20, though sometimes it was more than 100.

For point location queries, a rectangular grid was superimposed on the triangulation; each grid cell stored the triangle containing the center of the cell. A query point can be located by a triangulation walk from the stored triangle of the cell containing the query point. On average, the walk traversed between .5 and 2.2 triangulation edges, indicating that this simple grid is quite adequate.

5 Experimental results

The beam-tracing algorithm was implemented in C++ using the geometric data structures described above. All times are for an SGI R4400, running at a clock-doubled rate of 100 Mhz (for an effective clock rate of 200 Mhz).

Figure 5 gives a capsule description of the buildings used in the experiments. For each building, a representative location of the base was chosen and used throughout. The sample points were the lattice points of a regular grid with 2 meter spacing on a horizontal plane inside the building. For all runs, the frequency was 1.92 Ghz and the transmitter power was 34 mw.

The beam tracing algorithm was run with the filtering described in section 3, and in some cases without the filtering. For the filtered runs, a reflection cone was ignored if its estimated power was -90 dBm or below; a propagation path to a sample point was ignored if it was estimated to contribute less than .01 times the power already accumulated at the sample point. Figure 6 gives the running times of the filtered beam tracing algorithm. For one or two reflections, more than 50% of running time is spent doing triangulation walks; for higher reflections, the manipulation of reflection cones is dominant. The unfiltered running times are not given but are much worse, up to 30 times slower in some cases.

The average difference between filtered and unfiltered runs was typically .1 dB for a one reflection map, increasing to between .4 and .7 dB at three reflections; the standard deviation varied between .2 and .4 dB. The maximum difference was 2 to 5 dB, with larger values at higher reflections.

Figure 7 gives more details on selected runs. The number of reflection cones roughly fits the prediction of $n^{(r+1)/2}$, while the number of cones with estimated power above the -90 dBm threshold is much smaller. The column labeled 'paths' gives the number of propagation paths, i.e., the number of times a sample point lies inside a reflection cone (ignoring the threshold). The column labeled 'walks' gives the actual number of path power computations performed; clearly, the filtering has dramatically reduced this number from the number of propagation paths. The number of path computations per sample point gives an absolute measure of the efficacy of filtering.

6 Conclusions

The beam tracing algorithm is many orders of magnitude faster than a naive algorithm [10]; useful predictions can be obtained within a few minutes. The speedup is obtained by exploiting the geometry of the problem and by utilizing appropriate geometric data structures.

Similar techniques may be applicable for prediction in outdoor urban or suburban environments. The radio theory is somewhat similar, though diffraction becomes an important propagation mechanism. Computationally, the prediction problem is somewhat different as walls are modeled as opaque, with reflections only.

7 Acknowledgements

Thanks to Brian Kernighan, Margaret Wright, and Reinaldo Valenzuela for helpful comments.

Building	refl	cones ($\times 10^3$)	cones > -90dBm ($\times 10^3$)	paths ($\times 10^6$)	walks ($\times 10^3$)	walks/ sample	walks/ cone
Crawford	2	3	.6	.2	19	47	31.0
Crawford	3	61	2.2	3.4	32	78	14.2
Crawford	4	1269	4.5	57.2	41	98	9.1
Mt Airy	2	131	19.1	1.2	230	102	12.0
Mt Airy	3	4873	110.6	23.1	614	273	5.5
Mt Airy	4	155212	407.8	436.5	1213	539	3.0
Prudential	2	54	18.3	.7	114	84	6.2
Prudential	3	1434	105.9	11.9	276	202	2.6
Prudential	4	30886	363.8	189.9	537	393	1.5

Figure 7: Beam tracing: reflection cones, total and with estimated power above threshold; number of propagation paths; number of path power computations ('walks') performed; and path power computations per sample point and per cone.

References

[1] S. J. Fortune, D. M. Gay, B. W. Kernighan, O. Landron, R. A. Valenzuela, M. H. Wright, WISE design of indoor wireless systems: practical computation and optimization. *IEEE Computational Science and Engineering*, Spring 1995, pp. 58–68.

[2] S. J. Fortune, Algorithms for prediction of indoor radio propagation, manuscript, 1995.

[3] M. Held, J. Klosowski, J.S.B. Mitchell, Evaluation of collision detection methods for virtual reality fly-throughs, *Proceedings 7th Canadian Conference on Computational Geometry*, 1995, pp. 205–210.

[4] P. Kreuzgruber, T. Bründl, W. Kuran, R. Gahleitner, Prediction of indoor radio propagation with the ray splitting model including edge diffraction and rough surfaces, *Proceedings of the 1994 IEEE 44th Vehicular Technology Conference*, IEEE, 1994, 878–882.

[5] J.W. McKown, R.L. Hamilton, Jr. Ray tracing as design tool for radio networks, *IEEE Network Magazine*, Vol 5, pp. 27–30, 1991.

[6] J.D. Parsons, *The Mobile Radio Propagation Channel*, John Wiley and sons, 1992.

[7] A. Rajkumar, B. Naylor, F. Fiesullin, L. Rogers, Predicting RF coverage in large environments using ray-beam tracing and partitioning tree represented geometry, Technical Memorandum, AT&T Bell Laboratories, 1995.

[8] S. Ramo, J.R. Whinnery, T. Van Duzer, *Fields and Waves in Communication Electronics*, 3rd edition, John Wiley and Sons, 1993.

[9] S. Takahashi, K. Ishida, H. Yoshiura, A. Nakagoshi, An evaluation point culling algorithm for radio propagation simulation based on the imaging method, *Virginia Tech Fifth Symposium on Wireless Personal Communications*, pp. 13.1 – 13.11, 1995.

[10] R. A. Valenzuela, A ray-tracing approach to predicting indoor wireless transmission, *Proc. 1993 IEEE 43rd Vehicular Technology Conference*, IEEE Press, Piscataway, NJ, 1993, pp. 214–218.

[11] A. Watt, M. Watt, *Advanced Animation and Rendering Techniques: Theory and Practice*, Addison-Wesley, 1992.