A Novel Ray Tracing Acceleration Method for Radio Propagation Modeling Based on Prior Environment Processing

Nima Sedaghat Alvar, Ayaz Ghorbani, Hamidreza Amindavar
Department of Electrical Engineering
Amirkabir University of Technology
Tehran, Iran
n.sedaghat@aut.ac.ir

Abstract— **Ray Tracing has been successfully used in prediction of wave propagation models in recent years. Although this method has its own obvious benefits, it suffers from a big problem: slow performance. In this paper, a novel method is proposed in which the main focus is on reducing the number of ray-facet intersections. It includes a light-weight pre-processing operation on the environment which is completely independent of the locations of source and target and is performed once for each environment. In this step, measures for the distances between entities in the environment are computed and saved. Later in the main procedure, theses data will come to help reduce the number of ray-facet intersection tests dramatically.**

Keywords-component; ray tracing, UTD, wireless channel, radio propagation

I. Introduction

Radio propagation modeling in urban and indoor environments is a complicated electromagnetic problem. Ray tracing and Uniform Theory of Diffraction (UTD) are already widely applied to radio propagation modeling for wireless applications [1]-[4]. In most of the cases, the modeling requires a large number of facets to be modeled. In a mobile communications problem, it is necessary to compute the field for a large number of points along a predefined path.

The most time and resource consuming operation in ray tracing method, when used to model the propagation for a complex environment, is the ray-facet intersection test [3]. This is a test to determine if a ray intersects with the specified facet in the environment or not. This test should be performed each time a new ray is generated in the simulation process (after each reflection, diffraction, etc.). And each time, it should be tested for all the facets in the environment! So, when the number of involved facets is large, the ray tracing procedure is very slow and almost impractical.

Many modifications and acceleration techniques have been proposed to speed up the ray tracing procedure [3], [5]-[7]. Some of the most important techniques are Binary Space Partitioning (BSP) [3], Space Volumetric Partitioning (SVP) [3], etc.

In this paper, a novel method is proposed in which the main focus is on reducing the number of ray-facet intersections. There is no approximation in this method. Instead, it eliminates the need to perform some part of the operation, which in fact is redundant and is performed each time a ray is launched. This is done by performing a light-weight pre-processing operation on the environment, before the main procedure is started. This step is independent of the main ray tracing procedure and even the location of transmitter and receiver, and depends only on the environment. Thus this process is in fact performed when the environment data is being gathered & generated (not by the ray tracing software).

In the mentioned pre-process step, two matrices are generated which later in the main process, will come to help speeding up the process. Detailed information about generating these matrices and the way they will be used in the ray tracing procedure is described in the following sections.

II. The Proposed Method

A. Prior Information Collection

In this method, prior information about the environment is collected at the initialization stage. To do this, all N facets in the environment are indexed in an arbitrary order. Then two NxN matrices are constructed. Let the names be D^1 and D^2. Now D^1_{ij} corresponds to the minimum distance between facets i & j and similarly D^2_{ij} corresponds to the maximum distance between facets i & j. These matrices are symmetric matrices with zeros on the main diagonal.

Without loss of generality, facets are assumed to be positioned so that no crossing pair exists. This is true as we can divide each of the crossing facets to two distinct parts. Now, the minimum distance between two facets, is the minimum of a set containing distances between all edges of each facet and the other facet itself. For triangular facets, as an example, the set contains 6 entries. One example is depicted in Fig. 1 where the bolded dashed line shows the minimum distance between the two facets. Note that the triangles represent facets in 3D space.

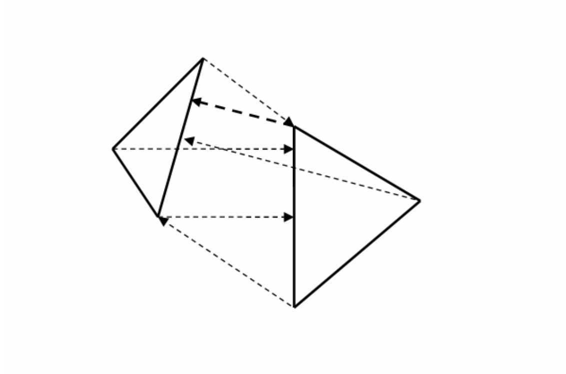

Figure 1. Example of finding minimum distance between two facets.

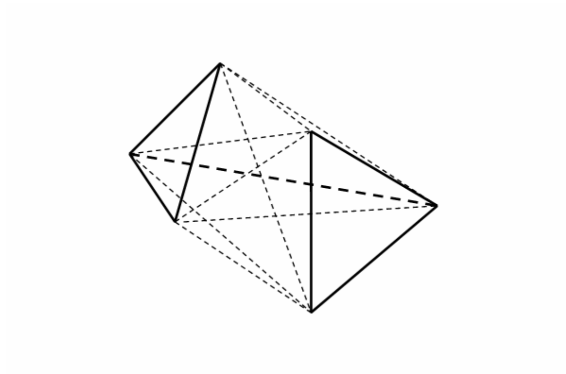

Figure 2. Example of finiding maximum distance between two facets.

The maximum distance is similarly the maximum of another set which is constructed in a simpler way. This set contains the distances between vertices of the first facet and the vertices of the other one. For the triangular facets example, it will have 9 members. Fig. 2 depicts an example for this.

B. Improved ray-facet intersection test

As mentioned previously, in a standard shooting-and-bouncing ray method, a ray, regardless of type of its source (a transmitter, a reflection point, etc.) is tested for intersection with all facets in the environment. After finding all intersecting facets, all distances between the source point and the intersection point on all facets should be computed and the nearest facet is selected. In this new method, however, a ray is tested with only a few facets, resulting in a dramatic improvement in speed.

The algorithm is as follows. The first ray which is originated from the transmitter is tested just like before, and no optimization is performed. From this step on, all the sources are placed on some facet. So the optimization can be applied. Let the ray be originated from some point on facet number n. All existing facets are ordered with increasing distances (near to far) according to the entries of row number n of matrix D^1. This helps to perform the test on nearer facets before the

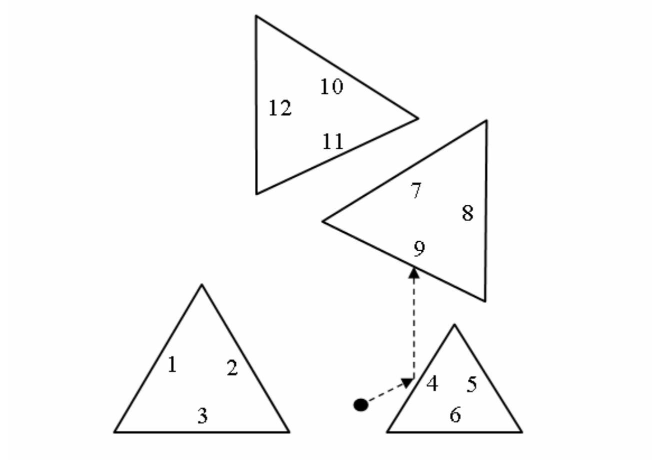

Figure 3. A simplified presentation of the method.

others. This step is paused when the first facet having intersection with the ray is found. Let the index of this facet be k, so the corresponding entry in D^1 would be D^1_{nk}.

The next step is to find those few neighbors which are probable candidates for "nearest intersecting facet". The necessity of this final check is explained with the fact that the distance obtained here, is a lower bound for the exact point-to-point distance. Meaning that the distance between ray source and intersection point might be, and usually is, greater than D^1_{nk}. So there might be other facets for which the exact point-to-point distance is less than the achieved one. These are the facets whose corresponding entry in D^1 (minimum distance) is less than D^2_{nk} (the maximum distance for the first candidate) AND have intersection with the launched ray. There is no other facet that the ray might intersect with, before a member of this set (visible to the ray).

A simplified example for this approach is given in the following section.

III. A Simplified Presentation Of The Method

Fig. 3 depicts a simple 2D scene. Note that in the 2D case, facets are replaced with lines. In this simple case, the ray originated from facet number 4 is to be considered. We have:

$$D^1_{4,5} = D^2_{4,6} = 0$$

So the facets number 5 and 6 are tested before all other ones. Since no intersection is detected between the ray and these facets, one step farther facets will be tested, which in this example are the number 8 and 9. Number 9 intersects with the ray. So number 9 will be the "first candidate". Here the first step is completed and the algorithm moves on to the next step.

In this step, a set of facets, satisfying the following condition should be found and tested:

$$D^1_{4,n} < D^2_{4,9}$$

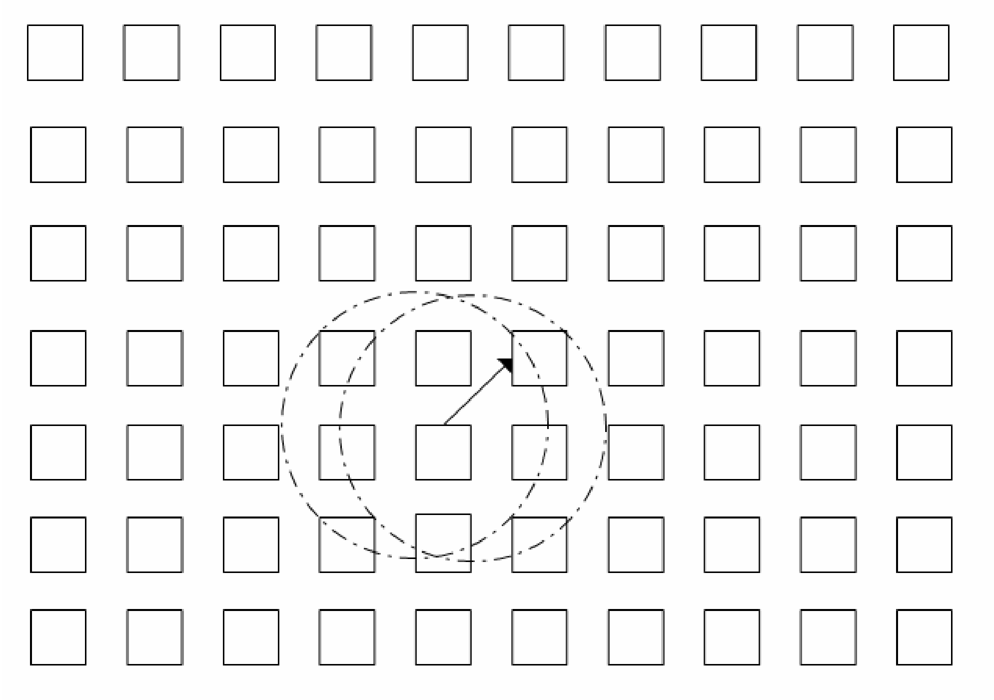

Figure 4. An example of effectiveness of the method in a 280-facet scene.

According to the figure, facets 2,3,7 are the members of this set. Finally these facets are subjected to intersection test, and since there's no intersection, number 9 will be the first-intersecting facet.

It can be seen that just some probable candidates have been tested in this approach. Although in this simplified example, which contains only 12 facets, there is no big achievement; but this method will have its best effect in crowded environments, such as urban areas. This will be considered in following sections where the effectiveness of the method is considered using computational approaches.

IV. Effectiveness Of The Method

A. Pre-processing Time

The preprocessing step, is performed only once for an environment. For example, if the ray tracing software is to be applied to a 3D city map, there's no need to perform the preprocessing each time. However this step is really light-weight. To investigate it, a simulation has been performed. 1000 facets have been distributed in 3D space uniformly. The preprocessing algorithm has been coded in C++, using Object Oriented coding schemes, without any special optimizations. The personal computer used for this simulation was an Intel® Pentium® Mobile processor 1.86GHz, running Microsoft Windows XP Professional™ with 504MB of RAM. Also none of the routine processes belonging to the OS have been stopped. The interesting result was that it only took about 21 seconds to generate both matrices!

B. Main Procedure Speed Improvements

The proposed approach has variable effects on speed of ray tracing procedure, according to the geometry of the scene. It appears in its best performance when applied to crowded scenes such as urban areas. It can be seen well in Fig. 4. In this simplified 2D example, there are 280 facets(line) in the scene. As it is shown, for this particular ray, only 36 facets should be tested. This is only about 13% of the standard case. Which proves the great improvement achieved using this method.

It should be noted that, in a real example, there are much more facets in the scene; thousands and even more. In such cases, the number of tested facets will remain equal to 36 which leads to interesting results. The computation time can be reduced even down to 1%!

V. Conclusion

In this paper, a novel fast ray tracing method was presented. The method's main focus is on reducing the number of ray-facet intersection tests in a ray tracing procedure, based on a preprocessing step which saves prior information about distances between pairs of facets. A simulation was performed and the results stated that the preprocess step takes a very short time for a scene containing 1000 facets (about 21 seconds).

Also, another example was designed to consider the efficiency of the method in the main process time. It was shown that this method can bring the process time down to even 1% of that in standard SBR method. This method can be combined with other known methods to achieve even better results.

ACKNOWLEDGMENT

The authors wish to gratefully thank the financial support provided by Iran Telecommunication Research Center (ITRC).

REFERENCES

[1] M. C. Lawton, and J. P. McGeehan, "The application of a deterministic ray launching algorithm for the prediction of radio channel characteristics in small-cell environments," IEEE Trans. Veh. Technol., vol. 43, no. 4, pp. 955-969, 1994.

[2] C. F. Yang, B. C. Wu, and C. J. Ko, "A ray-tracing method for modeling indoor wave propagation and penetration," IEEE Trans. Antennas Propagat., vol. 46, no. 6, pp. 907-919, 1998.

[3] Catedra et al., "Cell Planning for Wireless Communications" pp. 7-107, Feb. 1999.

[4] T. Fügen, J. Maurer, W. Sörgel and W. Wiesbeck, “Characterization of Multipath Clusters with Ray-tracing in Urban MIMO Propagation Environments at 2 GHz”, IEEE Proceedings of the International Symposium on Antennas and Propagation, Washington DC, USA, July 2005.

[5] F. A. Agelet, A. Formella, J. M. H. Rábanos, F. I. Vicente, and Fernando Pérez Fontán, "Efficient Ray-Tracing Acceleration Techniques for Radio Propagation Modeling," IEEE Trans. Veh. Tech., vol. 49, no. 6, 2000.

[6] Z. Yun, M.F. Iskander and Z. Zhang, "Fast ray tracing procedure using space division with uniform rectangular grid," IEEE Electronics Letters, vol. 36, no. 10, pp. 895-897, May 2000.

[7] I. Wald, H. Friedrich, G. Marmitt, P. Slusallek and H. Seidel, "Faster Isosurface Ray Tracing Using Implicit KD-Trees," IEEE Trans. Vis. Comput. Graph. vol. 11, no. 5, pp. 562-572 , 2005.