Path finding based on Monte Carlo Techniques Compared with a Full Ray-Tracing Approach in Narrow and Wide Bands

L. Aveneau and P. Combeau
IRCOM-SIC, CNRS UMR 6615, SP2MI
Bd Marie et Pierre Curie, BP 30179, 80962 Futuroscope Chasseneuil Cedex – France
Email : aveneau@sic.univ-poitiers.fr

***Abstract*— Efficient propagation prediction requires a tool both fast and accurate, which are two opposite qualities. Generally, such a tool is fast using crude approximation, leading to miss some propagation paths. Moreover, its accuracy is usually validated by some comparisons to measurement. Then, only the statistical behavior of such a tool is available, but not its particular one for a given scene.**

In this paper we propose a new optimization for finding the propagation paths, based on the Monte Carlo techniques. It allows a theoretical knowledge of the error due to our approximation, so that any engineer can really choose between faster or accurate calculations.

Our approach is compared with a Full Ray Tracing tool : This allows to discuss about the validity of our model, and its gain in terms of computation time. Comparisons are produced both in narrow and wide bands.

I. Introduction

Ray tracing is an interesting technique for propagation prediction. Based on the Uniform Theory of Diffraction [1], a Full Ray Tracing implementation (or FRT) allows to find all the theoretical contributions between a transmitter and a receiver. Many confrontations with measurements have shown that such a solution gives precise predictions [2].

Nevertheless, the problem with FRT is the computation time, that may become very large even with a simple scene. Indeed, the calculation consists in testing all the possible contributions. By increasing the number of allowed interactions in path computation, the combinatorial explosion of the number of such tests leads to impracticable tools.

Of course, many optimizations exist, mainly with visibility pre-computation, or using propagation properties [2]. Moreover, very powerful optimizations are less precise, since they imply to miss valid propagation paths [3] [4]. Computation time becomes certainly acceptable for an engineering use, but two questions arise : The validity of the prediction, and the user control on it. While the first one is established with comparisons to measurements, the second one is generally not available, even if the error control is the most important point of any optimization.

In this paper, we present a new approximate approach with error prediction account, and then compare it to a full solution. This error control is achieved using a mathematical framework for the optimization purpose. Our solution is a two pass one, with an oracle construction, followed by a ray tracing computation guided by the oracle.

In the second section, we present our approach, based on Monte Carlo techniques. So, we recall the Monte Carlo basis, and then detail our two pass implementation.

The purpose of the third section is to compare this new approach with a full one. We propose some test scenes, and discuss about the number and the importance of the missed paths with a wide band approach, and the narrow band differences. Next, we show the computation times, which are drastically reduced with our approximate solution.

As a conclusion, we discuss about our approach capacity to tend to the full solution, and about future works.

II. The Monte Carlo Approach

Monte Carlo techniques are used in many research fields, from nuclear computations to financial simulations. The basis of these techniques is the ability to compute the estimation of integral functions. Of course, this is useful when such functions are non computable in a reasonable time, as it is the case in visibility computations.

This section begins with a short presentation of the Monte Carlo principles, followed by implementation details.

A. Monte Carlo principles

The main idea of Monte Carlo integration [5] is to use random sampling for evaluating the following integral :

$$I = \int_{\Omega} f(x)dx \tag{1}$$

It is done by computing the estimate :

$$F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)} \tag{2}$$

where the N independent random values $X_1, ..., X_N$ are sampled using a convenient density function p.

The convergence rate mainly depends on the number of samples N. The standard deviation is

$$\sigma[F_N] = \frac{1}{\sqrt{N}} V\left[\frac{f}{p}\right] \tag{3}$$

where V denotes the variance. Then, dividing σ by a factor 2 necessitates 4 times more samples, and a fine calculation requires a large number of samples.

To deal with this problem, many variance reduction techniques have been developed. In this paper we use mainly one, the *stratified sampling*. It consists in partitioning the domain Ω into several non-overlapping regions – or strata – $\Omega_1, ..., \Omega_n$. Then, the work is done on each stratum Ω_i by sampling n_i random variables, according to some given density function p_i. This technique greatly reduces the variance on Ω, which becomes the sum of the variances for every stratum.

B. Implementation

Our implementation is quite easy to understand. It is based on a two pass process.

1) Oracle Construction: The first step consists in exploring the visible part from a given transmitter, using Monte Carlo techniques, by sampling each face and each dihedron. The samples are produced using a stratified strategy, by fixing $n_i = 1$. It is similar to a ray launching strategy, but without the reception part.

A sample is a primary ray, that can be reflected or diffracted a given number of times. The reflection is obtained by searching the first face intersecting by the primary ray. The diffraction is more difficult to control: in a ray launching strategy, the diffracting edges are found by testing the intersection of a primary ray with cylinders. But their size must depend on the parameterization of the oracle. Since we search to obtain as many paths as possible, we used the discrete tube technique [6]. This solution allows to quickly find all the edges included into the *n* first Fresnel ellipsoids. In order to miss the minimum of edges, the number of ellipsoids directly depends on the number and on the lengths of the primary rays.

Then, each reflection or diffraction fact is stored on the corresponding face or dihedron. We do not keep the localization of diffraction or reflection, but the history of faces or dihedra that have been used to bounce on the current object.

Notice that the final user may specify either the total number of samples, or the standard deviation.

2) Propagation: The propagation step looks like a classical Full Ray Tracing (FRT) solution. The only difference is that we do not explore all the combinations of faces and dihedra, but only those predicted by the oracle. Then, at a given reception location, for each history stored into each face or dihedron, we compute the corresponding path that may add some propagation information.

Since we do not explore all the theoretically possible paths, then the computation time is really reduced. We obtain a better ratio between the combinations that lead to a valid path, and those that are explored for nothing.

Notice that this technique is different from Tan's one [7], mainly in two points:

- We can use standard deviation to ensure a given precision,
- We do not use a sphere around a given receiver. In fact, the receiver location does not have any importance in the first step.

C. Error control

As detailed in the next subsection, the oracle construction can be computed with either a number of primary rays or a maximum standard deviation. With this last solution, the oracle must decide itself when to stop the computations.

With a classical Monte-Carlo approach, this is usually implemented in the following way. The computation is made two times, with the same number of samples. The results are compared, in order to verify that their difference is under a given precision. If it is the case, then the calculation is stopped. In the other case, then these two results is merged into one, and another Monte Carlo simulation is made with the same number of samples (*i.e.* twice the initial number). This process is repeated until the convergence is ensured.

In our oracle construction, the main question is how to compare the results. In fact, the results is the set of faces and dihedra visible from the transmitter, after some reflection and diffraction. So, the convergence of the Monte-Carlo process can be established by comparing these sets: our tests have shown that this can be obtained quickly, and allow an auto parameterization of the oracle construction.

In the ideal case, the computation leads to two identical sets of faces and dihedra. Of course, this do not imply that we obtain all the desired faces and dihedra. Nevertheless, this solution ensure that this kind of error have a small probability to arise.

III. Comparison with Full Solution

The comparison of our proposed solution with a Full Ray-Tracing is made with three different test scenes. We give here some results in narrow bands, by comparing the number of covered receivers, and the mean number of paths at each receiver locations. Then, in wide band we propose some comparisons of the power delay profil for some receivers.

All the computation made with the presented solution are performed with a mean of 1024 primary rays for each face or dihedron, and 32 diffracted rays at each diffraction points. The computation was performed on a Pentium IV at 2.6 Mhz, with 512 Mo of RAM, under Linux.

A. The test scenes

The first scene – scene A – represents a small part of the campus at the Poitiers University, in France. It contains 194 faces and 206 dihedra. As it can be observed on the Fig. 1, it is a classical suburban scene, with a poor building density.

As depicted in Fig. 2, the scene B is a classical downtown environment, with only 1352 building faces, and 1956 dihedra. The transmitter is below the rooftop at a street corner.

The last scene – scene C – is an extended view of the scene B, with 4038 building faces and 5999 dihedra, as it is shown in Fig. 3.

B. Narrow band results

Clearly, a good computation of the coverage area for these three scenes requires to process many reflections and

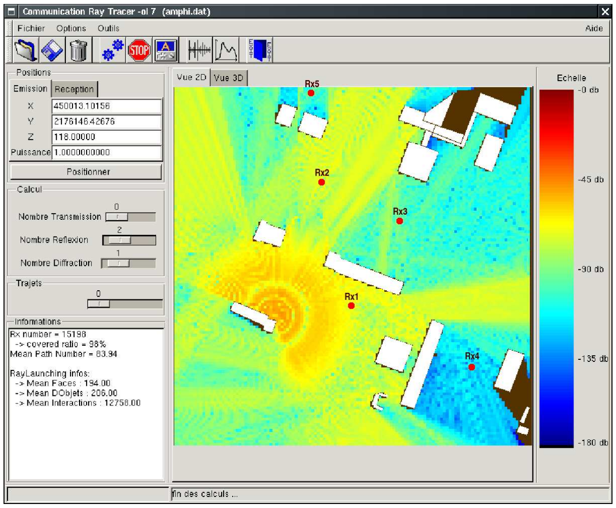

Fig. 1. The test scene A, for 1 diffraction and 2 reflections

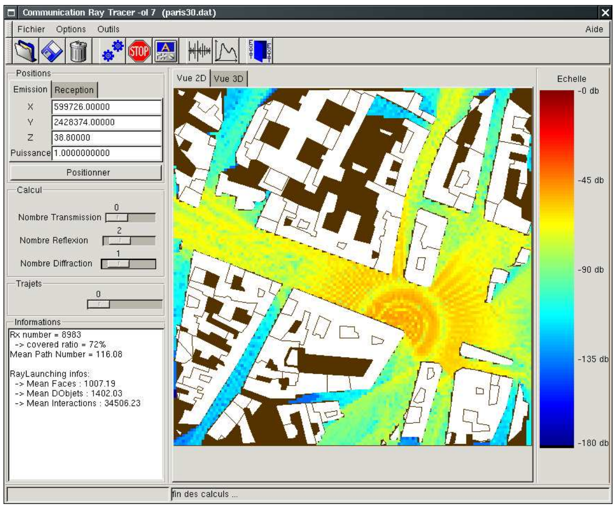

Fig. 2. The test scene B, for 1 diffraction and 2 reflections

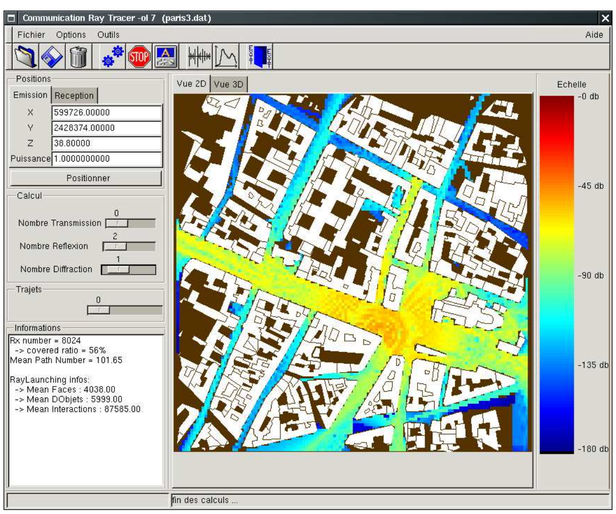

Fig. 3. The test scene A, for 1 diffraction and 2 reflections

diffractions. Then, with a full ray-tracing implementation, the computation time becomes very large and can not be achieved.

Since, with a FRT, the computation times are very important, we sometimes used an accelerated version base on a visibility graph construction – the italic computation times –. For information, the real time with 2 reflections and 0 diffraction for the test scene B is estimated to more than 218 minutes, and to more than 4926 hours with 3 reflections.

Inter.	Classical FRT		With oracle	
	Comp. Time	Mean PN	Comp. Time	Mean PN
1R 0D	2"93	2.73	2"31	2.73
2R 0D	3'21	4.36	5"20	3.90
3R 0D	2h19'31"	5.84	8"46	4.64
4R 0D			11"24	4.73
0R 1D	16"87	11.90	13"45	11.90
1R 1D	17'36"	52.89	1'31"	44.74
2R 1D	3h 18'	122.22	4'35"	83.94
3R 1D			9'22"	109.41

TABLE I

NARROW BAND RESULTS FOR THE SCENE A : THE COMPUTATION TIMES AND THE MEAN PATH NUMBERS

Inter.	Classical FRT		With oracle	
	Comp. Time	Mean PN	Comp. Time	Mean PN
1R 0D	9"71	3.06	3"58	3.06
2R 0D	*51'04"*	*5.85*	6"04	5.60
3R 0D			11"41	7.04
4R 0D			19"03	8.17
0R 1D	16"17	11.90	13"14	11.80
1R 1D	*1h 09' 49"*	*62.61*	1'35"	56.60
2R 1D			6'08"	116.08
3R 1D			18'59"	171.32

TABLE II

NARROW BAND RESULTS FOR THE SCENE B : THE COMPUTATION TIMES AND THE MEAN PATH NUMBERS

Inter.	Classical FRT		With oracle	
	Comp. Time	Mean PN	Comp. Time	Mean PN
1R 0D	22"73	3.12	4"83	3.12
2R 0D			7"74	5.13
3R 0D			12"74	6.47
4R 0D			19"82	7.83
0R 1D	1'34"	14.71	15"89	14.71
1R 1D			2'30"	53.10
2R 1D			11'46"	113.17

TABLE III

NARROW BAND RESULTS FOR THE SCENE C : THE COMPUTATION TIMES AND THE MEAN PATH NUMBERS

The results are respectively presented in the table I, II and III for the test scene A, B and C. The number of receiver locations is 15158 for the scene A, 8983 for scene B, and 8024 for the scene C.

Clearly, the computation times are drastically reduced, even with a small number of interactions. The mean number of computed paths is not so reduced. This is verified by comparing the coverage area obtained. Moreover, since we obtain almost all the main paths, the full path loss is not very different. For instance, with the test scene A, the total path losses for the 5 receiver locations depicted in FIG. 1 – where Rx1 and Rx2 are in LOS, and Rx3, Rx4 and Rx5 in NLOS – are 64, 76, 107, 112 and 99 dB with the FRT, and 64, 80, 109, 114 and 99 dB with the oracle method.

The computation with a FRT implementation are often impossible to obtain, due to the very long time needed. Nevertheless, the results presented show that our solution is faster and valid in term of number of paths obtained.

As a general conclusion on the narrow band results, it is shown that the Monte Carlo construction produces an important computation time optimisation, without missing too many paths. Moreover, the missed contributions generaly imply small faces or dihedra, and so are neglictable in the final results.

C. Wide band results

The results presented here are produced only with the scene test A, since the results are roughly the sames with the other scenes. They are computed with the same receiver locations than those presented in the previous section, since they illustrate LOS and NLOS configurations. The FIG. 4 shows the impulse responses at Rx1, and the FIG. 5 at Rx3, for the FRT algorithm and the oracle one. It seems obvious that the two curves are very similar, which is in accordance with the narrow band results.

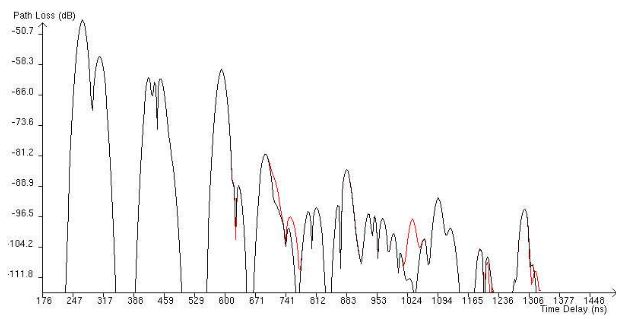

Fig. 4. Power Delay profiles at Rx1: Red line for FRT, and dark line for Oracle construction

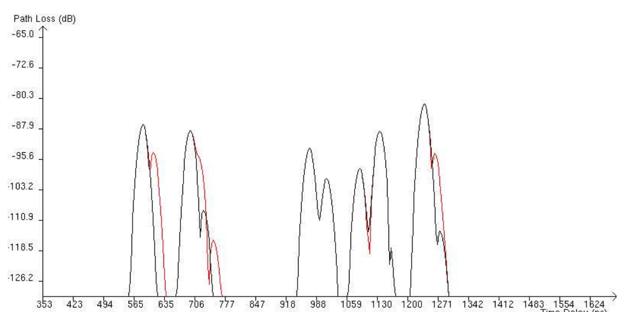

Fig. 5. Power Delay profiles at Rx3: Red line for FRT, and dark line for Oracle construction

The Delay Spread parameter is one of the most popular for characterizing the channel impulse response. The table IV shows such Delay Spread results for two reflexions and one diffraction, with both the FRT and the oracle method. These results seem to be different even if the impulse responses are very closed.

Rx	All paths		Dynamic of 15 dB	
	FRT	Oracle	FRT	Oracle
1	235.89	167.89	226.58	96.12
2	128.06	169.18	48.55	78.02
3	528.65	464.51	561.94	474.17
4	670.96	648.67	687.69	672.79
5	430.36	364.84	445.14	365.28

TABLE IV

DELAY SPREAD RESULTS IN NANO SECONDS

This can be explained by the fact that the delay spread parameter is very sensible to the low variations of the impulse response. Furthermore, one can note that, as it is shown in the litterature, the considered dynamic for the impulse response computation play an important role in the determination of the Delay Spread. Thus, we constate a difference of about 80 ns for the RX2 location between when we use a dynamic of 15 dB, which is the recommanted value by the IUT.

IV. CONCLUSION

We have presented here a propagation prediction tool, with an oracle construction based on Monte Carlo techniques. It allows to control the trade-off between the rapidity and the accuracy of the calculations. We have shown some results and discussed about them, showing the reduction of the computation times and the low prediction error obtained, compared to a full ray tracing software.

Of course, many aspect can be improved: The first one is the computation time, which may be reduced using some visibility precomputation from the receiver locations. The second problem concerns the missed paths, due to the Monte Carlo approach. A solution may be to perform the oracle construction using 3D volume propagation with a discrete approach. Nevertheless, this may imply to find too much contribution, and so to increase the global computation times.

REFERENCES

[1] R. G. Kouyoumjian et P. H. Pathak. *A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface.* Proceedings of the IEEE, 62(11):1449–1461, November 1974.

[2] L. Aveneau, R. Vauzelle, Y. Pousset and M. Mériaux, *Development and Evaluations of Physical and Computer Optimizations for the 3D UTD Model*, Proceedings of Antennas and Propagation 2000, Davos, Switzerland, April 2000.

[3] Fernando Aguado Agelet, Arno Formella, José Maria Hernando Rábanos, Fernando Isasi de Vicente, and Fernando Prez Fontán, *Efficient Ray-Tracing Acceleration Techniques for Radio Propagation Modeling*, IEEE Transactions on Vehicular Technology, vol. 49, no. 6, pp. 2089-2104, November 2000.

[4] Zhengqing Yun, Zhijun Zhang, and Magdy F. Iskander, *A Ray-Tracing Method Based on the Triangular Grid Approach and Application to Propagation Prediction in Urban Environments*, IEEE Transactions on Antennas and Propagation, vol. 50, no. 5, pp 750-758, May 2002.

[5] M. H. Kalos and P. A. Whitlock, *Monte Carlo Methods, Volume I: Basics*, John Wiley & Sons, New York, 1986.

[6] L. Aveneau, E. Andres, and M. Mériaux, *The discrete tube: an efficient acceleration for diffraction computation*, DGCI'1999, Lecture Notes in Computer Science, pp 423-424, Springer-Verlag.

[7] S. Y. Tan and H. S. Tan, *Propagation Model for Microcellular Communications Applied to Path Loss Measurements in Ottawa City Streets*, IEEE Transactions on Vehicular Technology, Vol. 44, No. 2, May 1995.