A New Approach to Solve Angular Dispersion of Discrete Ray Launching for Urban Scenarios

Zhihua Lai*†, Nik Bessis*, Guillaume de la Roche*, Pierre Kuonen†, Jie Zhang* and Gordon Clapworthy*

*Institute for Research in Applicable Computing
University of Bedfordshire
Luton LU1 3JU - UK

†GRID and Ubiquitous Computing Group.
University of Applied Sciences of Fribourg
UCH-1705 Fribourg - Switzerland

{zhihua.lai, nik.bessis, guillaume.delaroche, jie.zhang, gordon.clapworthy}@beds.ac.uk, pierre.kuonen@eif.ch

***Abstract*—Ray-based methods such as ray tracing and ray launching have been increasingly used in radio wave propagation modelling. Ray tracing is used for point-to-point multipath prediction (for few receivers) while ray launching, being more adaptable, is more suitable for multi-point prediction. However, ray launching suffers from angular dispersion which causes rays to miss pixels when the distance from the emitter increases. Several solutions such as beam tracing or ray splitting have been proposed to resolve this, but this paper presents a new approach, which is suitable for discrete ray launching, to avoid the problem. Results show that by this approach, discrete ray launching is suitable for radio wave propagation modelling. Significant speedups are observed compared to traditional ray-based models via parallelization techniques such as multi-threading and distributed computing. Complex channel characteristics due to multipaths in the urban environment can be obtained via this method.**

I. Introduction

Ray-based methods, comprising ray launching and ray tracing, are suitable for the calculation of radio wave multipaths in complex environments, which are essential for the analysis of channel characteristics such as delay spread.

Exact multi-paths can be obtained via ray tracing because an image-based methodology is employed, but the complexity of ray tracing increases exponentially with the number of objects in the scenario [1], so the performance is problematical in large urban scenarios in which only a limited number of ray interactions can be used.

In ray launching, the launched rays are normally separated by a small constant angle but this suffers from the inherent problem of angular dispersion, which is particularly the case for distant pixels that cannot be easily reached by nearby rays. In spite of this, ray launching is often chosen as an area prediction method because it offers lower computational complexity than ray tracing.

This paper is organised as follows. In Section II, the problem of angular dispersion with discrete ray launching is described, and in Section III, a new approach to solving this problem is developed and detailed. Results to validate the proposed method are presented in Section IV.

II. Angular Dispersion

As mentioned above, although ray launching methods may be computationally efficient, they suffer from the inherent problem of angular dispersion, as illustrated in Figure 1, where A is the emitter and an angle $\angle BAC$ separates the rays. Even if the angle is small, some pixels will be missed between rays A-B and A-C, especially when the distance between A and B or A and C is large. Besides, as can be seen, the reflected rays B-D and C-E will further disperse, producing a large gap between the two rays. Normally, this can be solved by introducing more rays between A-B and A-C in order to reduce the angle that separates the rays. This approach, called ray splitting, was proposed in [2]. However, it reduces rather than resolves the problem, it leads to a much higher computation complexity and, more importantly, one cannot find a perfect angle that suits all scenarios.

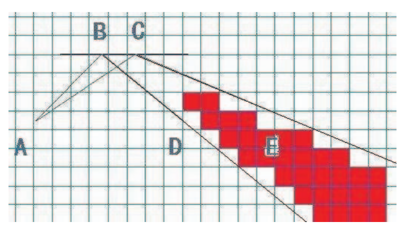

Fig. 1. angular dispersion; red pixels not visited by reflected rays

To illustrate the problem of angular dispersion in ray launching simulations, the scenario of Vigo, represented in Figure 2 is used. When ray launching with a separation angle of approximately one degree is used, the rays undergo three reflections and two diffractions, which can be illustrated in Figure 3. It is obvious that the rays are dispersed and that the received power is not computed at a number of pixels. With discrete ray launching, the performance largely depends on the resolution, i.e. the size of the pixels where the signal is considered as constant. A lower resolution (larger pixels) will produce larger gaps and hence lower accuracy and vice versa.

978-1-4244-2721-5/09/$25.00 ©2009 IEEE

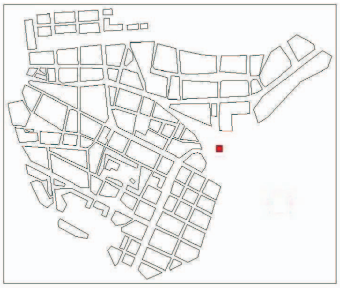

Fig. 2. scenario Vigo; emitter is marked with a red dot

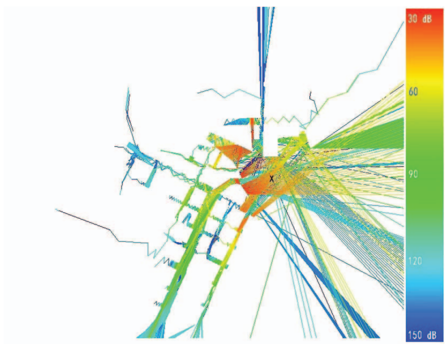

Fig. 3. scenario Vigo; angular dispersion problem; three reflections and two diffractions

III. Solution

In [3], a method (IRLA Intelligent Ray Launching Algorithm) based on discrete ray launching [4][5] is proposed to predict field strengths for an urban environment. The 3D vector scenario data is discretised into a cubic database. Each cubic entry represents different information: roofs, ground, corners etc. Based on the discrete information, the ray launching algorithms described in [3] offer fast computation of rays compared to alternative approaches. Moreover, the approach to solving ray angular dispersion can be easily adopted in IRLA to improve the accuracy.

A. Overview

Discrete ray segments in IRLA are identified by the positions of two cubes, which marks the start and end positions. Rays are considered double marked if they represent the same cubic space, for example, if two cubes originate from the same last reflection point and they have the same ID (rays are given a unique ID when they are launched from the emitter), these two discrete rays are double marked and only one of them should be considered. Elimination of double marked rays is important for both speed and accuracy. i.e. Calculations of duplicated rays cost time and lead to inaccurate prediction.

The LOS (Light-Of-Sight) component of IRLA, launches rays from the emitter, marks the pixels that are visible to the emitter and recursively launches secondary rays (reflection, diffraction). To avoid double marking and missing rays, the approach is to launch rays based on scenario boundary cubes, the coordinates of which satisfy: $C_{x,y,z} = 0$ or $C_{x,y,z} = N_{x,y,z}$ where N (N_x, N_y, N_z) are the maximum dimensions of the scenario. The total number of rays can thus be obtained via $N = 2N_z(N_x + N_y - 1)$. The approach guarantees that there are no missing pixels in the scenario but still requires the removal of duplicate rays, which can be easily performed by configuring a bit in the data associated with cubes when they are crossed by rays. Since N is already known from the size of the scenario, the rays can thus be easily parallelised. In [6], the algorithms have been parallelised via POP-C++ [7], which is a high-level, object-oriented, parallel programming language.

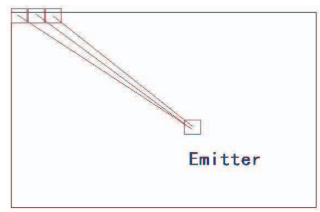

Fig. 4. launching LOS rays mechanism

The LOS component produces a number of secondary pixels where discrete rays are launched from and handled by HDR (Horizontal Diffraction and Reflection, see Figure 5). Since rays are discrete and cubes are positioned by a pixel, there will be gaps that rays cannot visit in between secondary rays (as depicted in Figure 3).

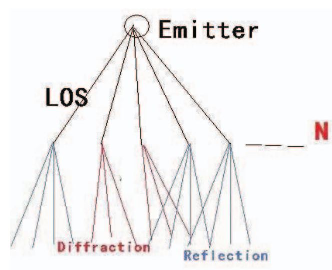

Fig. 5. launching rays mechanism

Hence, to compensate for the pixels that might be missed, the following strategy is proposed.

B. Algorithm

Vector ray launching suffers from an inherent problem: pixels are missed when the rays propagate and disperse. A typical solution is to use more rays but this greatly increases the computational complexity. IRLA uses discrete cubic coordinates instead of actual vector points - between two cubes, it is assumed that only one discrete ray exists. Unavoidably, if a discrete ray is formed from cube C to A, and A and B are neighbouring cubes, the reflection rays from C-A and C-B will disperse. Depending on the resolution, the rays will miss different numbers of pixels due to the angles between the two reflection rays. The situation is represented in Figure 3.

The problem can be resolved by using the technique proposed below (Algorithm 1).

Algorithm 1 avoid pixels missing (reflection)

$v_1 \leftarrow$ Reflection Ray 1 (C-A)
$v_2 \leftarrow$ Reflection Ray 2 (C-B)
$c_1 \leftarrow$ Boundary Cube that v_1 hits
$c_2 \leftarrow$ Boundary Cube that v_2 hits
for all $C \in$ Cubes between c_1 and c_2 **do**
  Launch Reflection Rays from A(or B) to C
end for

Algorithm 1 traces the rays from neighbouring pixels. Normally, a pixel (for example, at a wall) has two neighbour pixels. A virtual discrete ray has to be generated next to the actual deflection point (i.e. a point at which a reflection, refraction or diffraction occurs). The intersections of these two reflection rays and the boundary cubes can be obtained. The shadow area between the deflection point and two intersections should be considered as a valid reflection area, in which virtual rays should be launched. If the two intersections ($C1$ and $C2$) (Figure 6) are located in different bordering segments (i.e. $C1_x \neq C2_x$ or $C1_y \neq C2_y$ or $C1_z \neq C2_z$), the shadow area should be extended to $C1$-$C2$-D, from where the rays launched from B or C to between $C1$ and $C2$ disperse. The deflection point can be either chosen as the original deflection point, the neighbouring pixel or the middle point between these two. Since reflections occur between neighbouring cubes, one must remove the duplication of rays that are due to overlaps of the reflection shadow area. An easy way of doing this is to adopt an open and closed region strategy. i.e. the shadow area starts from C-$C1$ (contained) but leaves out C-$C2$.

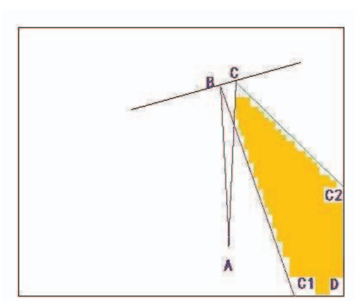

Fig. 6. reflection scenario

Diffractions are handled in a similar way: the shadow area is determined by Geometry Theory of Diffraction (GTD)[8] or Unified Geometry Theory of Diffraction (UTD)[9]. Secondary diffracted rays are launched from the deflection pixel (Figure 7).

To handle the removal of double marking for thousands of rays (both reflected or diffracted) in an efficient manner, the following pseudo code (Algorithm 2) is proposed.

Algorithm 2 allocates two matrices for handling reflection and diffraction. At each deflection, the corresponding counter is incremented and pixels within the shadow area are marked with this counter if they have not been updated. Secondary cubes are collected for the next iterations. This approach is efficient since there is no need to manually clean the shadow area because this is automatically accomplished via the use of incrementing counters.

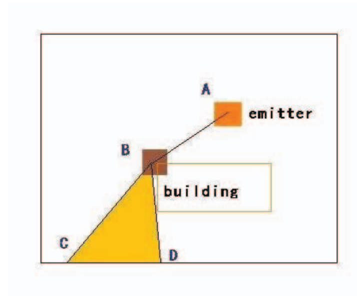

Fig. 7. diffraction scenario

Algorithm 2 avoid double marking

$M_r \leftarrow$ 3D counter matrix for reflection
$M_d \leftarrow$ 3D counter matrix for diffraction
$C_r \leftarrow$ Initial reflection counter
$C_d \leftarrow$ Initial diffraction counter
Inc(C_r) if reflection shadow area is being handled.
Inc(C_d) if diffraction shadow area is being handled.
if M_r[current cube] $\neq C_r$ **then**
  handle_reflection(current cube)
  M_r[current cube] $\leftarrow C_r$
  collect current cube for next reflection
end if
if M_d[current cube] $\neq C_d$ **then**
  handle_diffraction(current cube)
  M_d[current cube] $\leftarrow C_d$
  collect current cube for next diffraction
end if

IV. RESULTS

This section presents some results via the use of this fast approach in IRLA. The proposed method for handling reflection and diffraction was applied to the scenario of Vigo. The coverage for three reflections and two diffractions (excluding vertical diffraction) is displayed in Figure 8. It is verified that more pixels are filled and the rays are able to propagate for a greater distance.

COST-231 Munich [10] is considered as a reasonably large scenario with 2,088 buildings and more than 8,000 vertices and is often used as a benchmark scenario. To evaluate the performance of the model, two simulations (with/without the approach presented in this paper) were performed and the outcomes compared.

From Table II, it is clear that, with the proposed method, more rays are traced and hence more reflections and diffractions are computed. This leads to a higher accuracy but at the

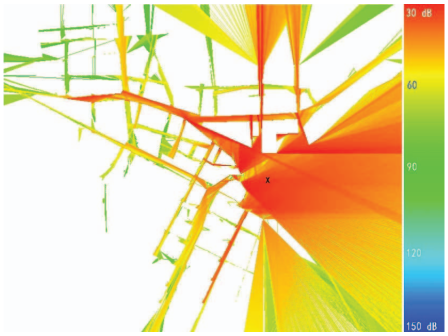

Fig. 8. scenario Vigo; angular dispersion problem solved; rays propagate further

TABLE I
NETWORK CONFIGURATIONS FOR BOTH RUNS

Area	8.1 km^2 (2.4km X 3.4km X 100m)
Buildings	2088
Antenna Type	Omni
Emitter Frequency	GSM 947MHz
Resolution	5 X 5 X 5
Maximum Reflection	3
Maximum Horizontal Diffraction	7
Maximum Vertical Diffraction	**Unlimited** *
Maximum Transmission	**Unlimited** *

* until signal strength is under threshold

cost of a longer running time. It is noticeable that the accuracy for this scenario does not improve to a great extent, due to the fact that the three measurement routes provided are mostly distantly located in a dense urban environment, the signal contribution of which are mainly from vertical diffractions (not affected by angular dispersion in IRLA since it does not actually launch and trace rays). Using this model, multipath information can be easily obtained and one example (receiver located at LOS to emitter) can be found in Figure 9.

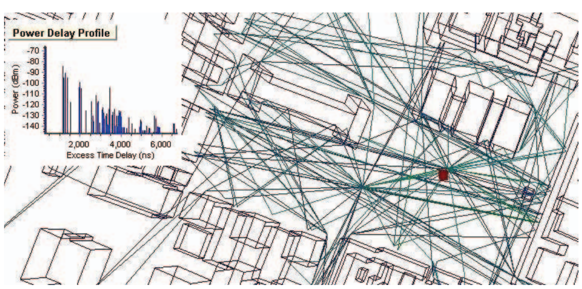

Fig. 9. Munich, multi-paths at LOS receiver

TABLE II
COMPARISONS FOR BOTH RUNS

	With	Without
Total Reflections	121, 170	89, 902
Total Diffractions	157, 201	101, 356
Total Multi-paths	122, 722	93, 521
Avg STD for 3 routes	7.11	7.81
Avg RMSE for 3 routes	7.02	7.69
Avg running time (s)	49	22

V. CONCLUSION

In this paper, a new approach to solving the angular dispersion problem for ray launching has been presented. This method is especially suited to the discrete ray launching model previously presented in [3]. Detailed algorithms and acceleration techniques have also been given in the context. Results shown in Section IV have validated the use of this method which increase the accuracy of the ray launching. However, such a technique has a computational cost, and the computation time is increased. That is why future work includes the investigation of grid computing, which can speed up the ray launching model [11]. It is believed the ray launching model can be further improved via distributed computing techniques.

ACKNOWLEDGMENTS

This work is supported by the EU-FP6 "RANPLAN-HEC" project under grant number MEST-CT-2005-020958, "GAWIND" under grant number MTKD-CT-2006-042783 and "Marie Curie Fellowship for Transfer of Knowledge". Jean-Francois Roche and Laurent Winkler deserve acknowledgments for their help on POP-C++ programming. The acknowledgments have to be extended to Dr Michael Reyer from RWTH Aachen University for his generous help and colleagues Alvaro Valcarce and Hui Song for their useful suggestions.

REFERENCES

[1] F. Aguado, F.P. Fontan, and A. Formela. Indoor and outdoor channel simulator based on ray tracing. *Department of Communications Technologies; university of Vigo, Spain*, 1998.

[2] Steven Fortune. Algorithms for prediction of indoor radio propagation. *ATT Bell Laboratories report*, January 1998".

[3] Z. Lai, N. Bessis, G. De La Roche, H. Song, J. Zhang, and G. Clapworthy. An intelligent ray launching for urban propagation prediction. In *The Third European Conference On Antennas and Propagation*, Berlin, Germany, March 2009.

[4] R. Mathar, M. Reyer, and M. Schmeink. A cube oriented ray launching algorithm for 3D urban field strength prediction. In *IEEE Communications Society*, June 2007.

[5] M. Klepal. *Novel Approach To Indoor Electromagnetic Wave Propagation Modeling*. PhD thesis, Czech Technical University In Prague, 2003.

[6] Z. Lai, N. Bessis, P. Kuonen, G. De La Roche, J. Zhang, and G. Clapworthy. A performance evaluation of a grid-enabled object-oriented parallel outdoor ray launching for wireless network coverage prediction. In *The Fifth International Conference On Wireless and Mobile Communications*, Cannes/La Bocca, French Riviera, France, August 2009.

[7] T.A. Nguyen and P. Kuonen. Programming the grid with POP-C++. In *Future Generation Computer Systems*, Hcmc University Of Technology, Faculty Of Computer Science and Engineering, Ho Chi Minh City, Viet Nam, January 2007.

[8] V.A. Borovikov and B. Ye. Kinber. *Geometrical Theory Of Diffraction*. Institution Of Electrical Engineers, December 1994.

[9] D.A. Mcnamara, C.W.I. Pistorius, and J.A.G. Malherbe. *Introduction to the Uniform Geometrical Theory of Diffraction*. Artech House Publishers, January 1990.

[10] COST231 urban micro cell measurements and building data. http://www2.ihe.uni-karlsruhe.de/forschung/cost231/cost231.en.html.

[11] Z. Lai, N. Bessis, J. Zhang, and G. Clapworthy. Some thoughts on adaptive grid-enabled optimisation algorithms for wireless network simulation and planning. In *UK e-Science, All Hands Meeting*, Nottingham, UK, September 2007.