Efficient Implementation and Evaluation of Parallel Radio Wave Propagation

Florian Schröder, Michael Reyer, and Rudolf Mathar
Institute for Theoretical Information Technology
RWTH Aachen University
D-52074 Aachen, Germany
Email: {schroeder, reyer, mathar}@ti.rwth-aachen.de

***Abstract*—For planning, analysis, and optimization of modern, self organizing radio networks field strength prediction plays an essential role. A huge number of predictions is needed for planning purposes while high quality and level of detail are important as well. In this paper, we present a promising ray optical approach for radio wave propagation to meet those demands. As the performance improvement of processors has shifted from higher clock speed to more cores, our redesign of a radio wave propagation algorithm supports parallel hardware architectures. In this work, we will present promising results based on our algorithm supporting the ray optical effects reflection and vertical diffraction at roof tops while allowing for multi path evaluation.**

I. Introduction

Fast field strength prediction plays an essential role in planning, analysis, and optimization of modern radio networks. Such networks will be self organizing networks (SON), need detailed information like multi path propagation (MPP), and should support MIMO channel models. Despite the required high detail level of information the predictions need to be fast for two reasons. On one hand, a huge variety of prediction scenarios have to be evaluated for planning of networks. On the other hand, detailed and precise predictions are needed to support the self organizing management process of future networks.

An overview of radio wave propagation models is given in [1] and [2]. Models proposed in literature can basically be divided into (semi) empirical and ray optical models. Semi empirical models calculate the received power on the basis of frequency, distance, and an empirical part mainly describing the obstacle influence. The strength of such approaches is the speed of prediction. However, the prediction quality is low if the influence of deflection effects like diffraction, reflection, and transmission is high. This leads to ray optical approaches which identify ray paths through the scene to combat the lack of prediction quality at the cost of higher computation time. Those models are primarily used in urban and/or indoor scenarios, because there the required data is available, e. g., building outlines, construction drawings, and so forth.

In ray optical models the environment, e. g., buildings, is usually described by polyhedrons, formed of surface sections, called facets in the following. If the building heights are given, but the roof shapes are missing the data is called 2.5D. Several ray paths between the transmitter and receiver point are searched, regarding deflection effects as reflection on, transmission through, and diffraction at edges of the given facets. Ray optical models are classified as ray tracing and ray launching, depending on the way the ray paths are determined.

In ray tracing models all possible ray paths starting from a receiver point to the transmitter are searched. The strength of such models are precise predictions, but at cost of huge computational effort. Ray launching methods emit a finite set of rays from transmitter in predetermined directions, cf. [3] and [4]. If rays hit a facet, possible deflection effects are performed. For diffraction it is necessary to emit a new ray bundle into the diffraction cone, whereas for reflection the direction has to be changed. As the rays disperse, important deflection points or even receiver points may not be hit. To avoid that effect a receiver point is called to be hit if the ray path crosses its proximity. Consequently, no receiver point is missed, if the ray density is high enough. A method to keep the number of rays low is to adaptively multiply rays at each deflection point, e. g., [4]. In another method the amount of followed rays is reduced by jointly processing similar rays as 3D cones, see [5]. Beyond this work, mixed models have been investigated which follow partly rays and partly use empirical parameters, cf. [6]. Additional work on prediction algorithms, which is based on ray optical approaches, can be found for example in [7]. If a dense prediction is needed the ray launching method is superior to the ray tracing model as in ray launching rays are bundled which are processed individually in ray tracing.

Recent hardware development indicates that the improvement of processors has shifted from clock speed to parallel architectures. Here are to be mentioned multi core processors, graphics cards, and also the CBEA (Cell Broadband Engine Architecture). The high potential of parallel architectures is well known, especially for graphics cards it is discussed in [8], [9]. Using graphics cards for non graphical purposes is called GPGPU (General Purpose computations on Graphics Processing Unit), see [10]. There are numerous applications, e. g., physical simulation in [8], image processing [11], audio processing, and sorting in [9]. With the introduction of the Playstation 3, for short PS3, in March 2007 the cell processor is available for research at low cost. The high potential of this architecture is described in [12].

Adaptations of algorithms are needed with respect to those hardware developments. Such adaptations are necessary in the first place to be able to run the algorithms and additionally to exploit the full potential of the parallel architectures. Radio wave propagation has and is been applied to parallel architectures. Recent results for graphics cards may be found in [13], [14] and for the cell in [15], [16].

In the present paper we extend the results of [16] by vertical diffraction. Additionally, the algorithm for radio wave propagation has been generalized to support different parallel architectures. This approach enables for multi path propagation while providing high resolutions in urban scenarios.

This paper is organized as follows. In Section II the general structure of the ray launching algorithm is described. Various acceleration techniques are explained in Section III. In Section IV the evaluation setup and the promising results are presented and discussed. Finally, Section V concludes this work.

II. The Ray Launching Algorithm PIROPA

The ray launching approach **P**arallel **I**mplemented **R**ay **O**ptical **P**rediction **A**lgorithm (PIROPA) is introduced in this section. Therefore, we first present the basic components and their interactions. After that, we derive an abstract scheme from this design and describe the opportunities it bears with respect to parallelism.

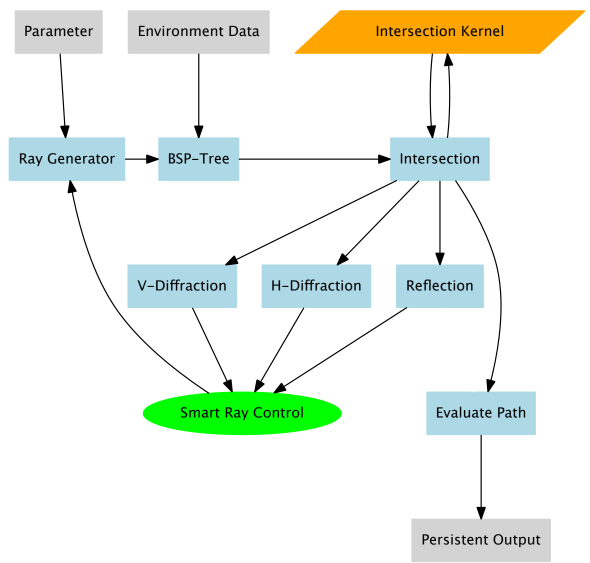

Fig. 1. Logical data flow in PIROPA

A. Design of PIROPA

The algorithm PIROPA is explained following Figure 1. Parameters and the environment data, e. g., building data, and receiver points, represent the input of the algorithm. Following the ray launching approach rays consisting of a starting point and a direction are initially generated with the transmitter as starting point heading into every direction. Note that in the course of the algorithm the starting points of rays will also be deflection points which will be explained later. The rays are connected to the relevant part of environment data, which is efficiently determined by a BSP tree in our case, see Section III. Using this information the intersection test is performed. It marks all interaction points on a ray. As the intersection is a core component we exemplary show how OpenCL is used. All sub algorithms are black boxes such that internally this specific algorithm may be outsourced o special hardware (e. g. GPU). This outsourced kernel is depicted as orange block.

In the following steps interaction points are analyzed with regard to their relevance for physical effects. Our model for field strength prediction considers the ray optical effects 1. line-of-sight, 2. reflection, 3. horizontal diffraction, and 4. vertical diffraction which are depicted in Figure 2. Though, the algorithm is not limited to these

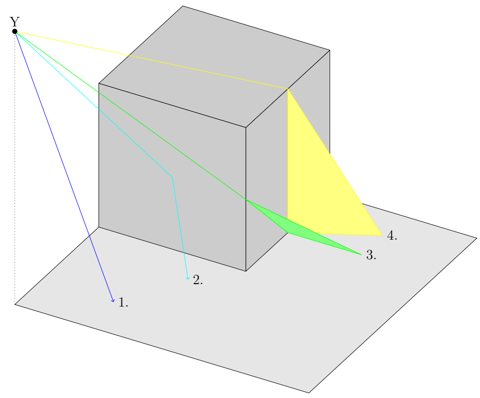

Fig. 2. Overview of ray optical effects

effects and may be easily enhanced. If an interaction point is a deflection point or lies in the vicinity of a receiver point the starting point of the corresponding ray and that interaction point build a path segment. The path is complete if a receiver point is met. Hence, the whole path describes the track from the transmitter to a receiver point with all its deflection points in between. If the path is not complete some information has to be fed back to the ray generator to continue the calculations of path segments.

The smart ray control which is no algorithm of its own but a control structure within the sub algorithms will check if and how many new rays need to be generated. This control enables for ray multiplication such that the density of rays is kept high enough to ensure to hit all relevant interaction points in the vicinity. Obviously, those smart decisions of dropping or multiplying rays is challenging and has severe impact on the overall runtime. We will comment on that topic in Section III in more detail. Completed paths are evaluated and the evaluation is made persistent[1]. The block scheme has several advantages which are discussed in the next part.

B. Abstract Scheme

By omitting the semantics of the blocks from the algorithm depicted in Figure 1 we derive an abstract scheme. It is closely related to the algorithm but reveals the opportunities of parallelism including the parallel design of work flows.

The scheme consists of three abstract objects: Jobs, Workers and Manager. Jobs are data container, Workers are basic templates for algorithms and the Manager is in control of the work flow. Some, at least one, of the Workers have to be sources and generate Jobs and some have to be sinks making the processed Jobs persistent.

In PIROPA rays or path segments are Jobs, the sub algorithms (ray generator, intersection test, and so on) are Workers and the Manager controls the Job flow, i. e., connections, between the Workers. These blocks depend on each other on data as well as on algorithmic level. On data level a Job only depends on its predecessors which means each generation[2] can be handled completely in parallel. The dependencies of the Workers are defined by the underlying algorithms and are marked by their connections.

Such an abstract scheme displays many opportunities for parallelism. Each Worker may run in its own thread. Different Jobs at the same Worker may be handled in parallel. The Manager controls who is active at which time and assigns the Jobs to the Workers. It controls which Worker will run in parallel and/or at which order. Additionally, it selects – if available – which version of a sub algorithm should be used, e. g., a CPU or GPU optimized version. New Jobs will only be generated if the parent Job is completely processed and a Worker will only get a Job if the necessary preprocessing is done. On this level, several concepts of parallelism can be analyzed and compared without reprogramming the whole algorithm. Compared to previous monolithic designs this is a major improvement for testing of different solutions and opens plenty of opportunities.

C. Current Implementation

Our current implementation supports reflection and vertical diffraction and the effect line-of-sight if a receiver point is hit. The intersection test is currently a single thread CPU implementation for debugging and usage on single core machines. Another version using OpenCL which will make use of graphics or accelerator cards to maximize performance is available soon. The smart ray control is based on some straightforward rules to be mentioned in Section III. For comparison purposes we have implemented the same evaluation model as in [17]. Additionally, we support an output including the angles of departure and arrival, the path length, and the attenuation of multiple paths. This output will be used by our industry partner Qosmotec Software Solutions GmbH[3] for their channel simulator QPER which supports MIMO channel models. The prototypical development has been partly funded by the BMWi \ AiF within ZIM-KOOP[4].

[1]Stored in a file, e. g., as picture or list of data.

[2]A generation is a set of Jobs created from the same source description.

III. Acceleration Techniques

In this section several acceleration techniques are described. This includes the usage of Binary Space Partitioning (BSP) with Ropes for the partition of the environment data. Secondly, a hardware independent paradigm for accelerating the intersection test of rays and obstacles is introduced. This acceleration is applicable to environments where all facets lie in parallel or orthogonal to the xy-plane. This is the case for 2.5D building data as given in the COST231 project of Munich, see [1]. And finally, some easy, but effective stop criteria for the smart ray control of the algorithm are explained.

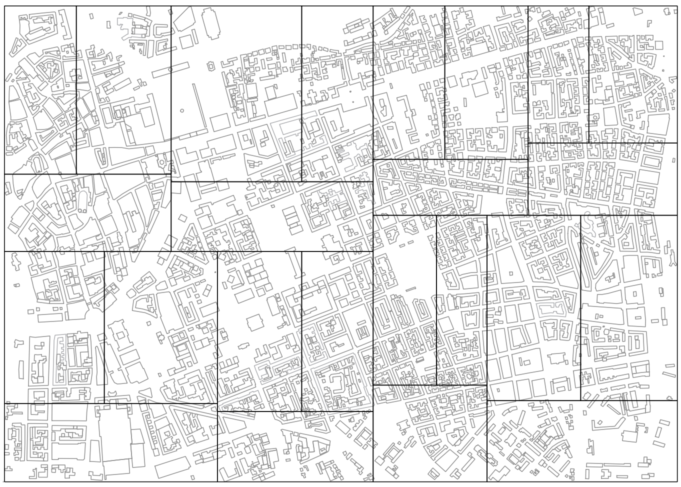

Fig. 3. Graphical representation of a BSP tree in Munich, [1]

A. Binary Space Partitioning (BSP) Trees with Ropes

With BSP a set of data will be repeatedly divided into two subsets distinguished by a binary decision and, hence, saved into a binary tree. In our case the division is assessed by axis parallel planes such that each node of the tree represents the partitioning plane and each leaf denotes a subset of the final partition. In the following leaf is a synonym for subset.

Figure 3 shows a 2D representation of a tree, where each rectangle denotes a leaf and each line – excluding the outer

[3]http://www.qosmotec.com

[4]http://www.bmwi.de and http://www.aif.de

frame lines – depicts a node. For example the vertical line roughly in the middle, splitting the image into two, is the root node followed by two horizontal lines splitting these halves again and so forth. Since the tree has no information about the neighborhood of leafs, an additional structure is set on top of the tree, so called ropes. Ropes add to each leaf the information about their neighbors such that quick navigation among leafs is possible. This allows for a highly efficient selection of the relevant leafs used in the intersection test, see Section II.

B. Smart Intersection

In our case regarding 2.5D building data it is most likely that rays just differing in the vertical angle either will hit a facet or pass over it. Meaning, in a full 3D calculation a bunch of rays will intersect the facet at same (similar) x and y coordinates but differing in the z coordinate, where z represents the height. By splitting up this 3D calculation into two 2D calculations, as suggested in [18], much of the redundancy is taken away. In Figure 4 the first phase is depicted where vertical overlaying rays are put together. The green and blue half circles mark the position

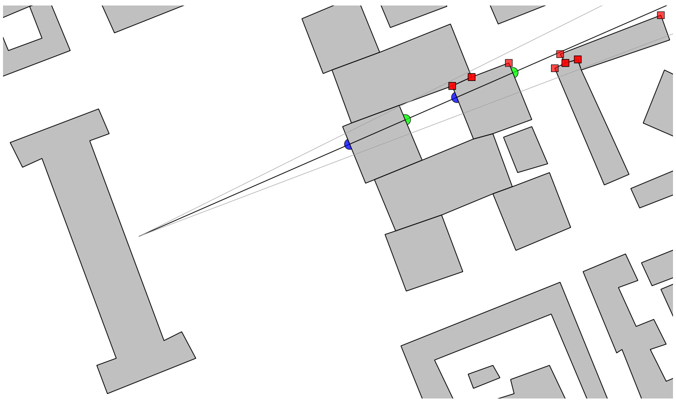

Fig. 4. First phase: determining intersections in xy-plane

where a ray hits or drops out of a building. The red squares mark edges in a close cone around the ray, indicating candidates for horizontal diffraction. In the second phase the vertical overlaying rays are separated again to process the different effects. Figure 5 depicts the vertical diffraction in the second phase, where the numbers of vertical diffractions a path undergoes on its way to the receiving plane is exemplary shown. So naturally a recursive process is defined in which nodes can be discarded whenever all children are processed. The children are independent and therefore allow for easy parallelization. The sheer number of calculation operations and memory copies is reduced by this paradigm. This reduction comes at the price of much higher logical complexity, but already improved the overall performance in a prototypic test scenario.

C. Stop Criteria

In addition to the above mentioned techniques it is beneficial to identify promising rays as early as possible.

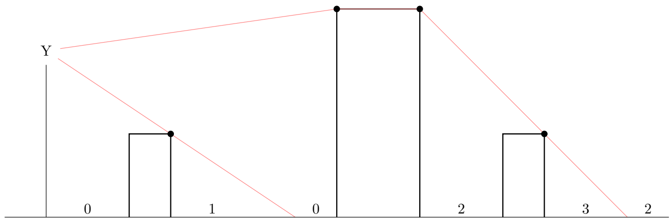

Fig. 5. Second phase: vertical diffraction

Therefore, rays are dropped in the smart ray control if their power drops under a configurable threshold. Additionally, the maximum number of effects along a path may be set. It is possible to set bounds for each effect or for the total number of effects so far. Another stop criteria which is not supported yet will drop a bunch of rays if a maximum number of multi paths is reached. More complex criteria are easy to insert and will be considered in a later stage of the project.

IV. Numerical Evaluation

In this section we will first shortly introduce the basic setup for our evaluation before presenting and discussing the results.

A. Setup

The evaluation is performed on basis of data from the well known COST 231 project in Munich, see [1]. It contains 2.5D building data of a 2.5×3.5 square km area of Munich and three measurement tracks. We took the 1031 receiver points from the first measurement track. The current version of PIROPA implements the effects roof diffraction and reflection. In the smart ray control paths with attenuations above 200 dB are cut off. The restrictions to the maximum numbers of reflections and vertical diffraction are chosen such big that the maximum attenuation always is reached in advance.

B. Evaluation

We evaluate PIROPA on its operational use in a multi path environment. For that purpose we determine the number of different paths leading to a receiver point. Therefore, we replace a set of similar paths with a representative path. Similar paths are characterized by similar angle of departure and distance, and by same effects. Since we collect paths hitting the vicinity of the receiver points. We make the characterization parameter with respect to the vicinity size. In the MIMO channel simulation model WIM2, see [19], there are, depending on the scenario, up to 20 clusters. Those clusters correspond to the different paths in our evaluation. We call PIROPA appropriate to multi path simulation, if it is able to identify at least the same amount of clusters as in WIM2.

In the WIM2 model there are 20 simulated rays per cluster. Those are constructed in a regular pattern by a deterministic algorithm. For PIROPA we suggest to calculate only one path representing the cluster. By means of this path we may construct more rays in this cluster following the ray construction in the WIM2 model. Consequently, the ray density can be significantly reduced by the smart ray control.

C. Results

The analysis of our algorithm shows that we cover all receiver points. Comparing the prediction quality by using the same evaluation model as in [17] reveals that we are similar to CORLA when it is run without horizontal diffraction. Naturally, we overestimate the attenuation where horizontal diffraction contributes significantly to the reception in comparison to the test run measurement of the COST 231 project.

In this scenario we get an average number of clusters of about 31. However, in the furthermost part of the test run, where the attenuation is about 80-90 dB higher than in line of sight case, only few clusters are identified. We expect the number of clusters to increase considerably if horizontal diffraction is introduced to the algorithm. Consequently, our approach should be appropriate to multi path simulation.

An evaluation of runtime is not meaningful at the current stage. For that purpose we need to include horizontal diffraction and configure the smart ray control in such a way that we get the desired number of clusters for multi path simulation. However, our current investigations clearly indicate that complex computations of multi path information is practicable within a reasonable amount of time.

V. Conclusion

We have introduced our radio wave propagation algorithm PIROPA. Its strength lies in its conceptual design which inherently allows for parallelization and supports flexibility, e. g., different evaluation models may be easily applied. Our implementation has been extended by vertical diffraction. Results including the effects reflection and vertical diffraction show great promise for multi path propagation. Even the identification of 20 clusters – this is the maximum value in the WIM2 model – should come into reach when implementing horizontal diffraction and some rules for the smart ray control. Consequently, a support for MIMO channel model will be applicable.

Acknowledgment

This research was partly supported by the UMIC excellence cluster of RWTH Aachen University and partly funded by the BMWi \ AiF within ZIM-KOOP initiative.

References

[1] E. Damosso, Ed., *COST Action 231: Digital mobile radio towards future generation systems, Final Report.* Luxembourg: Office for Official Publications of the European Communities, 1999.

[2] N. Geng and W. Wiesbeck, *Planungsmethoden für die Mobilkommunikation.* Springer-Verlag, 1998.

[3] G. Durgin, N. Patwari, and T. Rappaport, "An advanced 3D ray launching method for wireless propagation prediction," in *Proc. IEEE Vehicular Technology Conference*, Phoenix, Arizona, May 1997, pp. 785–789.

[4] M. Schmeink and R. Mathar, "Preprocessed indirect 3D-ray launching for urban microcell field strength prediction," in *Proceedings IEEE AP*, Davos, Switzerland, 2000.

[5] T. Frach, "Adaptives hierarchisches Ray Tracing Verfahren zur parallelen Berechnung der Wellenausbreitung in Funknetzen," Ph.D. dissertation, RWTH Aachen University, 2003.

[6] J. Beyer, "Ausbreitungsmodelle und rechenzeiteffiziente Methoden für die Feldstärkeprognose in städtischen Mikrozellen," Ph.D. dissertation, Universität-Gesamthochschule Siegen, 1997.

[7] R. Wahl, G. Wölfle, P. Wertz, P. Wildbolz, and F. Landstorfer, "Dominant path prediction model for urban scenarios," in *Proc. IST Mobile and Wireless Communications Summit*, Dresden, June 2005.

[8] M. Harris, *GPU Gems.* Addison-Wesley, 2004, ch. Fast fluid dynamics simulation on the GPU, pp. 637–665.

[9] P. Kipfer and R. Westermann, *GPU Gems 2.* Addison-Wesley, 2005, ch. Improved GPU sorting, pp. 733–746.

[10] "GPGPU." [Online]. Available: http://www.gpgpu.org

[11] R. Strzodka, M. Droske, and M. Rumpf, "Fast image registration in DX9 graphics hardware," *Journal of Medical Informatics and Technologies*, vol. 6, pp. 43–49, November 2003.

[12] S. Williams, J. Shalf, L. Oliker, S. Kamil, P. Husbands, and K. Yelick, "The potential of the cell processor for scientific computing," in *CF '06: Proceedings of the 3rd conference on Computing frontiers.* New York, NY, USA: ACM, 2006, pp. 9–20.

[13] D. Catrein, M. Reyer, and T. Rick, "Accelerating radio wave propagation predictions by implementation on graphics hardware," Dublin, Ireland, April 2007, pp. 510–514.

[14] A. Schmitz, T. Rick, T. Karolski, L. Kobbelt, and T. Kuhlen, "Simulation of radio wave propagation by beam tracing," in *Eurographics Symposium on Parallel Graphics and Visualization*, 2009.

[15] F. Schröder, "Konzepte zur effizienten Umsetzung von Algorithmen für die IBM Cell-Architektur - Anwendung auf Verfahren der Feldstärkeprädiktion im Mobilfunk," Master's thesis, RWTH Aachen, 2008.

[16] F. Schröder, M. Reyer, and R. Mathar, "Fast radio wave propagation prediction by heterogeneous parallel architectures with incoherent memory," in *Proceedings of Wave Propagation and Scattering in Communication, Microwave Systems and Navigation*, TU Chemnitz, Dec. 2010, pp. 89–93.

[17] R. Mathar, M. Reyer, and M. Schmeink, "A cube oriented ray launching algorithm for 3D urban field strength prediction," Glasgow, Scotland, UK, June 2007, pp. 5034–5039.

[18] J.-P. Rossi and Y. Gabillet, "A mixed ray launching/tracing method for full 3-d uhf propagation modeling and comparison with wide-band measurements," *Antennas and Propagation, IEEE Transactions on*, vol. 50, no. 4, pp. 517–523, Apr 2002.

[19] "WINNER II D1.1.2 V1.2 WINNER II channel model IST-4-027756."