Dublin Institute of Technology
ARROW@DIT

9th. IT & T Conference

School of Computing

2009-10-01

Parallelised EM Wave Progagation Modelling for Accurate Network Simulation

Catalin David
Dublin City University

Conor Brennan
Dublin City University, brennanc@eeng.dcu.ie

Olga Ormond
Dublin City University

Marie Mullen
Dublin City University

Follow this and additional works at: http://arrow.dit.ie/ittpapnin

Part of the Computer Sciences Commons

Recommended Citation

David, C., Brennan, C., Ormond, O., Mullen, M.: Parallelised EM wave progagation modelling for accurate network simulation. Ninth IT & T Conference, Dublin Institute of Technology, Dublin, Ireland, 22nd. - 23rd. October, 2009.

This Conference Paper is brought to you for free and open access by the School of Computing at ARROW@DIT. It has been accepted for inclusion in 9th. IT & T Conference by an authorized administrator of ARROW@DIT. For more information, please contact yvonne.desmond@dit.ie, arrow.admin@dit.ie.

This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 License

Parallelised EM wave propagation modelling for accurate network simulation

Catalin David, Conor Brennan, Olga Ormond, Marie Mullen
Network Innovation Centre,
Research Institute for Networks and Communications Engineering (RINCE),
Dublin City University,
Ireland.
brennanc@eeng.dcu.ie

Abstract

A description of ongoing work which aims to provide better quality propagation models for use in network simulators is provided in this paper. A 3D ray-tracing model is described which allows for accurate specification of a variety of wave scattering phenomena. Details of its parallelisation are given as well as a discussion of future work including the incorporation of a visibility algorithm. Results illustrate the increased realism obtained by using site-specific propagation models.

Keywords: Ray tracing, wave propagation, parallel computing, ad hoc networks

1 Introduction

The accurate modelling of wireless networks is a complex task as it must take into account what is happening at all layers of the OSI model. Simulation packages such as NS2 allow engineers to form reasonably useful models which attempt to address all such layers. However these are acknowledged to suffer in particular from over-simplistic propagation models at the physical layer [Stepanov, 2008]. The assumption of free-space propagation or simple two-ray models are simply inadequate to model the complexity of propagation effects, especially in urban or indoor scenarios. Consequently the conclusions to be drawn from such network simulations are often questionable, suffering as they do from an overly benign model of the wireless channel. Deployments of networks designed using incorrect channel information can be sub-optimal and inefficient, in terms of energy usage and capacity, as compared to those based on correct treatment of the physical layer[Coinchon, 2002]. An alternative is to incorporate a more realistic propagation model into the simulations. Options include the COST231 Walfisch Ikegami model, which is based on an abstracted model of propagation over a succession of rooftops [Walfisch, 1988]. Input parameters include the average building height and separation and as such it only models propagation in an average sense. A further drawback is its inability to produce information about delay spread and angle of arrival. In contrast a ray-tracing model uses precise building database information and computes a simplified high frequency approximate solution to Maxwell's equations for the problem at hand. It can generate signal strength information as well as angle of arrival and delay spread information, the latter being particularly important for high speed wideband systems where ISI becomes more of an issue. The main drawback of ray-tracing is the onerous computational burden associated with it. It is simply unfeasible to incorporate a ray-tracing engine into a network simulator to be called each time an estimate of propagation information between two points is needed. The approach we describe in this paper is instead one where exhaustive simulations are performed off-line and the resultant database is made available to the network simulator as required. In particular the parallelisation of the ray tracing code is essential, especially when modelling ad-hoc networks. The analysis of ad-hoc networks require ray traces for multiple receiver locations. Thus the use of a parallelised ray tracing code would allow the simulation of each transmitter-receiver channel to be performed on a separate processor. The simulation of such networks

would not otherwise be feasible. This paper documents our ongoing work in this area and is organised as follows. Section 2 reviews the ray-tracing methodology for computing electromagnetic fields and briefly explores the parallelisation of the resultant ray tracing code. Section 3 presents some results illustrating the necessity of proper consideration of site-specific propagation modelling. We close with a discussion of future work and draw some conclusions.

2 Ray Tracing

The ray tracing method generates an asymptotic solution to the problem of determining electromagnetic wave propagation. It identifies dominant direct, reflected and diffracted rays composed of straight line segments from the transmitter to the receiver and uses geometrical optics and the uniform theory of diffraction [Kouyoumjian, 1974] to compute the fields associated with each ray. The total field at a point is given by the superposition of fields from each ray. To illustrate the principles of ray tracing we consider the case of a plane wave of unity amplitude incident upon a half-plane at an incidence angle $\phi^{'} = 30^{o}$. We plot the magnitude of the various fields at a distance of one wavelength from the edge of the half plane. This result as illustrated in figure (1). This result is validated in [Balanis, 1989]. The

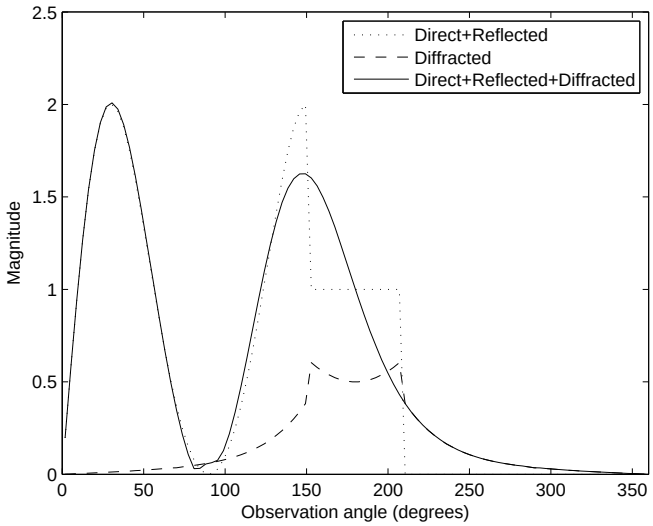

Figure 1: Field distribution of various components of a plane wave incident normally on a conducting half-plane.

main computational burden associated with the method is the specification of the rays. This is greatly facilitated by the use of image theory (valid in the case of planar walls and ground - see figure (2) where reflected ray-segments are thought of as emanating from image or virtual sources which allows easy identification of the reflection point. However the specification of reflected rays still represents a significant burden as the number of images grows exponentially with the number of buildings. A similar observation holds for diffracted rays which require the identification of a diffracting point lying on a building edge. Images of diffracting edges allow the identification of reflected-diffracted rays and so on for higher order combinations. Visibility algorithms [Agelet, 1997, Schettino, 2007] reduce the burden significantly by using information about what faces and edges are visible to the transmitter in order to prune the number of active images and traverse the image tree more efficiently. For this paper we instead address the computational burden issue by running the code in parallel on PC cluster managed by the DCU Centre for Scientific Computing and Complex Systems modelling. Future work will involve the incorporation of a suitable visibility algorithm, which in conjunction with the parallel computing resource, will allow us to accurately tackle problems on a realistic scale. For completeness we describe

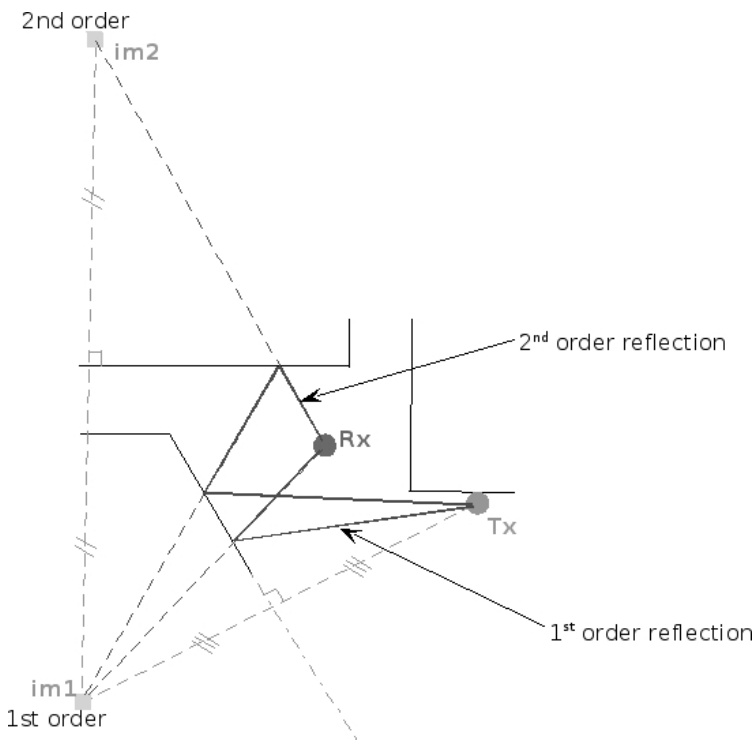

Figure 2: Image Theory used to identify reflection points

the computation of the reflected and diffracted fields. The field reflected from a planar boundary to the point $\mathbf{P}$ is obtained by decomposing the incident field into components parallel and perpendicular to the plane of incidence [Balanis, 1989] and applying appropriate reflection coefficients at the point of reflection $\mathbf{Q}$ to each. The components are attenuated by a spreading factor and multiplied by a phase term which accounts for propagation from the reflection point a distance s to the field point. This can be expressed compactly using dyadic notation as follows

$$\mathbf{E}^r(\mathbf{P}) = \mathbf{E}^i(\mathbf{Q}) \cdot \mathbf{R} A_r(s) e^{-jks} \tag{1}$$

where the dyadic reflection coefficient is written as

$$\mathbf{R} = R_s \hat{\mathbf{e}}^i_\perp \hat{\mathbf{e}}^r_\perp + R_h \mathbf{e}^i_\parallel \mathbf{e}^r_\parallel \tag{2}$$

The hard and soft reflection coefficients are given by

$$R_s = \frac{\cos\theta - \sqrt{\epsilon' - \sin^2\theta}}{\cos\theta + \sqrt{\epsilon' - \sin^2}} \tag{3}$$

$$R_h = \frac{\epsilon' \cos\theta - \sqrt{\epsilon' - \sin^2\theta}}{\epsilon' \cos\theta + \sqrt{\epsilon' - \sin^2}} \tag{4}$$

where the complex permittivity is given by

$$\epsilon' = \epsilon_r - \jmath \frac{\sigma}{\omega \epsilon_0} \tag{5}$$

The fields diffracted from a vertical or horizontal building edge can be expressed in a similar fashion. Assume that the building edge represents the edge of a wedge with interior angle α. We define n as

$$n = \frac{2\pi - \alpha}{\pi} \tag{6}$$

The diffracted fields from the point $\mathbf{Q}$ to the point $\mathbf{P}$ is then given by

$$\mathbf{E}^d(\mathbf{P}) = \mathbf{E}^i(\mathbf{Q}) \cdot \mathbf{D} A_d(s) e^{-jks} \tag{7}$$

where the diffraction dyad is given by

$$\mathbf{D} = -D_s \hat{\beta}_0' \hat{\beta}_0 - D_h \hat{\phi}' \hat{\phi} \tag{8}$$

The soft and hard diffraction coefficients are in turn given by

$$D_{s,h} = D_1 + D_2 + R_{s,h} \left(D_3 + D_4 \right) \tag{9}$$

where

$$D_1 = -\frac{e^{-\jmath\pi/4}}{2n\sqrt{2\pi k}\sin\beta_0} \cot\left(\frac{\pi + (\phi - \phi')}{2n}\right) F\left(kLa^+\left(\phi - \phi'\right)\right) \tag{10}$$

$$D_2 = -\frac{e^{-\jmath\pi/4}}{2n\sqrt{2\pi k}\sin\beta_0} \cot\left(\frac{\pi - (\phi - \phi')}{2n}\right) F\left(kLa^-\left(\phi - \phi'\right)\right) \tag{11}$$

$$D_3 = -\frac{e^{-\jmath\pi/4}}{2n\sqrt{2\pi k}\sin\beta_0} \cot\left(\frac{\pi + (\phi + \phi')}{2n}\right) F\left(kLa^+\left(\phi - \phi'\right)\right) \tag{12}$$

$$D_4 = -\frac{e^{-\jmath\pi/4}}{2n\sqrt{2\pi k}\sin\beta_0} \cot\left(\frac{\pi - (\phi + \phi')}{2n}\right) F\left(kLa^-\left(\phi - \phi'\right)\right) \tag{13}$$

$$\tag{14}$$

The Fresnel function used in the diffraction coefficients is defined as

$$F(x) = 2\jmath\sqrt{x} \int_{\sqrt{x}}^{\infty} e^{-\jmath u^2} du \tag{15}$$

and the constants $a^{\pm}$ are given by

$$a^{\pm} = 1 + \cos\left(\phi \mp \phi' - 2\pi n N^{\pm}\right) \tag{16}$$

where $N^{\pm}$ is the integer that most nearly satisfies

$$2\pi n N^+ - \left(\phi \mp \phi'\right) = \pi$$
$$2\pi n N^- - \left(\phi \mp \phi'\right) = -\pi$$

2.1 Parallelisation

The ray-tracing algorithm as described is readily parallelised as it essentially consists of a sequence of ray-traces to independent field points. It is a straightforward matter to split the workload evenly by grouping field points together and assigning the ray-traces for each group to one processor. The incorporation of a dynamic visibility algorithm, where the visibility list is sequentially built as the ray traces to the field points are performed, may render this process more complicated as it is important to equally balance the load given to each processor. However we intend to apply a pre-computed visibility algorithm that will mean that our code retains the simple easily parallelised structure described above. Open MPI was used to parallelise the code on the DCU Sci-Sym Ampato cluster. MPI works by spawning multiple copies of the same program, assigning to each one a rank and a pool in which it would reside. The rank is useful

because it is available to the programmer through the interface, allowing the manipulation of the code in different ways for different processes. The pool is necessary and useful for the case in which one would use multiple pools and would do collective communication (to the processes in the same pool). The pools are designed in the same manner in which tags are (one process can be part of multiple pools). In the design of the parallel code, only one pool and a master-slave architecture was used. The principle behind this is that there is one process reading, directing and storing the data, while the other processes use the data they receive from the main process in order to do the necessary computations and then pass on the data to the main process.

3 Results

Our code was applied to a mocked-up $1km^2$ city centre environment composed of roughly 100 buildings of various sizes situated on flat ground. The buildings were assumed to be concrete with electrical parameters chosen accordingly. For each possible transmitter location accurate propagation data is computed at 2GHz for points on a regular grid ($1m$ by $1m$ although a finer or coarser resolution is possible). Figures (3-7) show the total signal power levels (in decibels) throughout the grid for a variety of transmitter locations located along a north-south trajectory. These images are taken from a larger set of images which together constitute a movie illustrating how the power level distribution varies as the transmitter moves. Fields were not computed inside buildings and these show up as dark rectangles on the plots. The transmitter height for each simulation was $3m$ and the field points were all assumed to be $2m$ above the ground. Up to a maximum of 12 reflections and a single diffraction were included in each ray path although these settings can be altered within the code. In practice a maximum reflection image order of 4 to 5 as well as 3 to 4 diffractions per path usually suffices, with higher order effects being significantly attenuated. The site-specific nature of the wave propagation is immediately evident from the results as the the power level distributions seen are very different from the concentric circle pattern one would expect from a free-space or two ray model. In addition each transmitter location produces a radically different field distribution and it is clear that simple, separation-distance based propagation models cannot capture the complexity of signal structure seen, arising as it does from a variety of propagation effects such as shadowing, wave canyoning and multiple diffraction. The propagation data computed throughout the grid is stored in an individual file for each transmitter location and made available to be read into the NS2 simulator, replacing the inbuilt propagation models as required. Using 64 processors of the DCU Sci-Sym cluster (8 processors on each of 8 machines) each computation took 48 minutes to run, which represented a near-linear speed up compared to using a single processor. It should be noted that in addition to the signal strength information displayed here the code can compute time of arrival information as well as angle of arrival data.

4 Conclusions and Future work

The code has successfully been ported to run on a parallel computing resource which has reduced the run times significantly. However the computational burden still grows sharply with the number of transmitter locations and buildings. Consequently we are developing a visibility algorithm which will pre-compute the visibility between building faces and edges in order to speed up the identification and processing of rays. This will greatly reduce the run-time for each transmitter location and enable us to realistically create a pre-computed database of propagation information throughout the environment for all possible transmitter locations on a regular grid. Such exhaustively tabulated data will be used in the future in order to model the performance of ad-hoc networks.

Acknowledgements: The authors would like to acknowledge Enterprise Ireland and Science Foundation Ireland (through the ODCSSS UREKA scheme) for their support of this work.

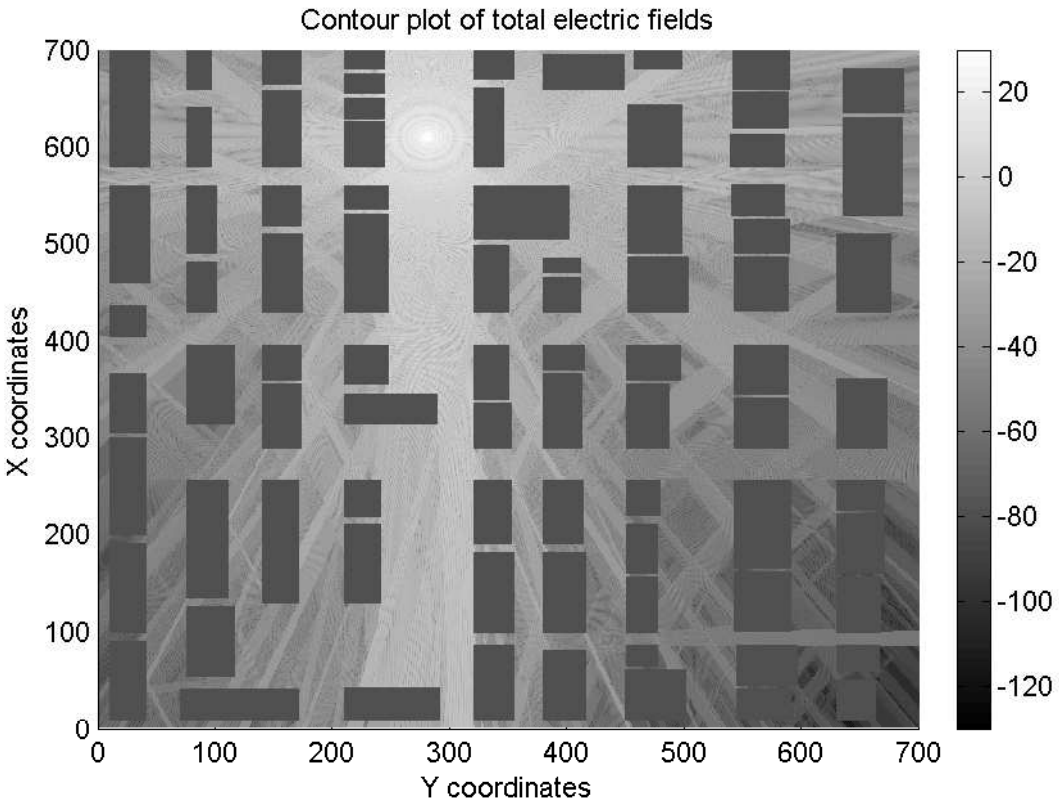

Figure 3: Field strength throughout urban region at 2GHz. Transmitter at $(270, 600, 3)$

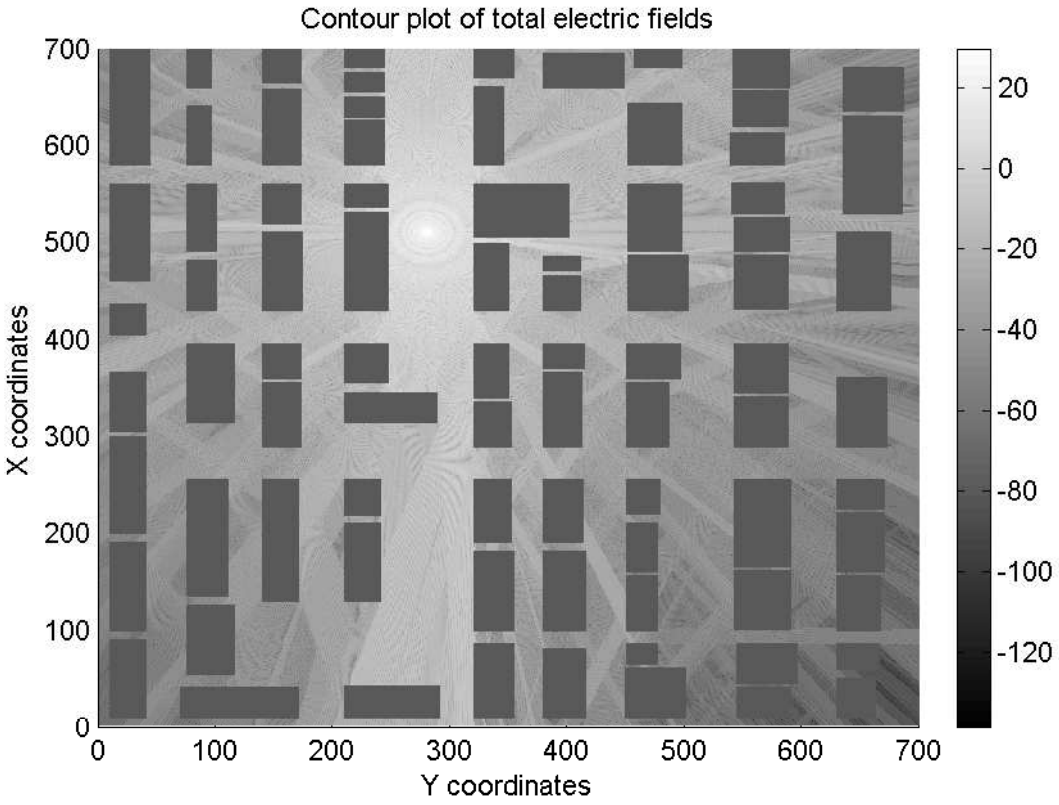

Figure 4: Field strength throughout urban region at 2GHz. Transmitter at $(270, 500, 3)$

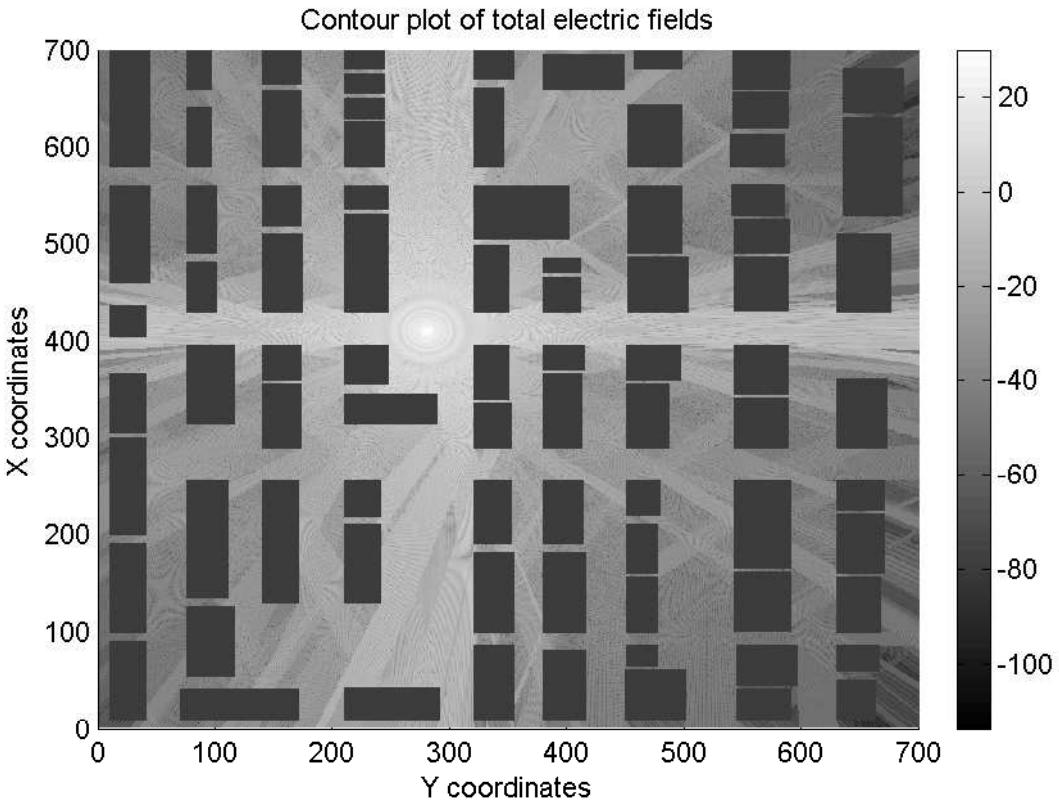

Figure 5: Field strength throughout urban region at 2GHz. Transmitter at $(270, 400, 3)$

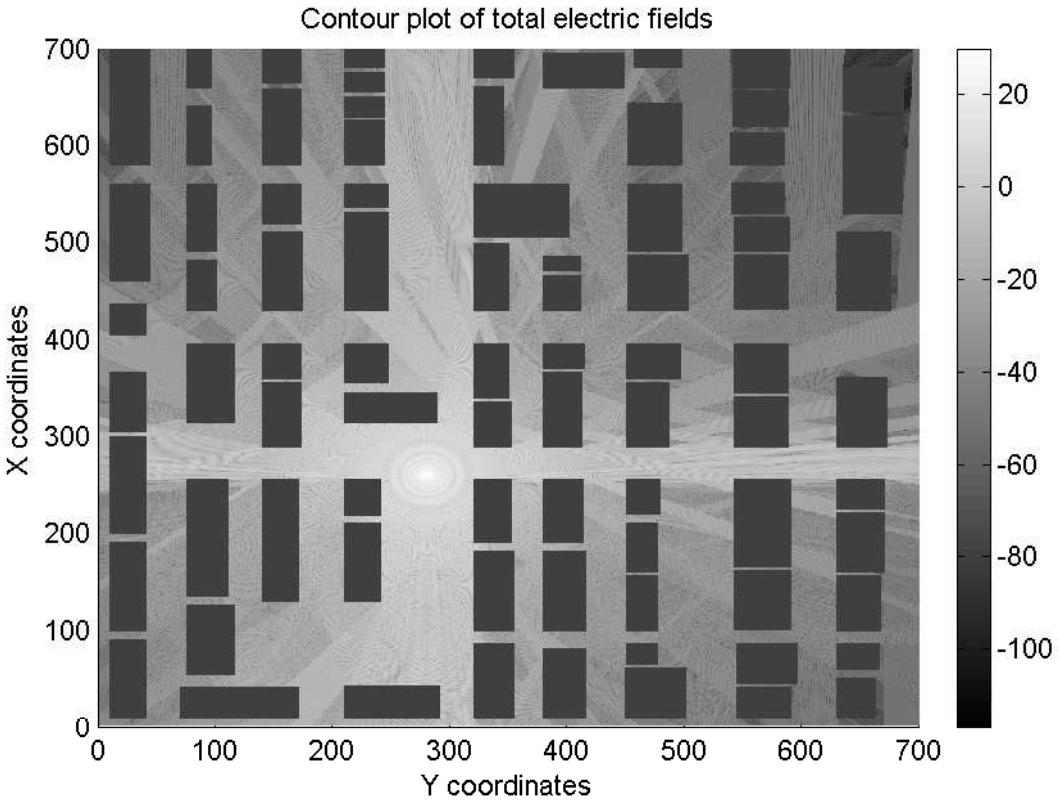

Figure 6: Field strength throughout urban region at 2GHz. Transmitter at $(270, 250, 3)$

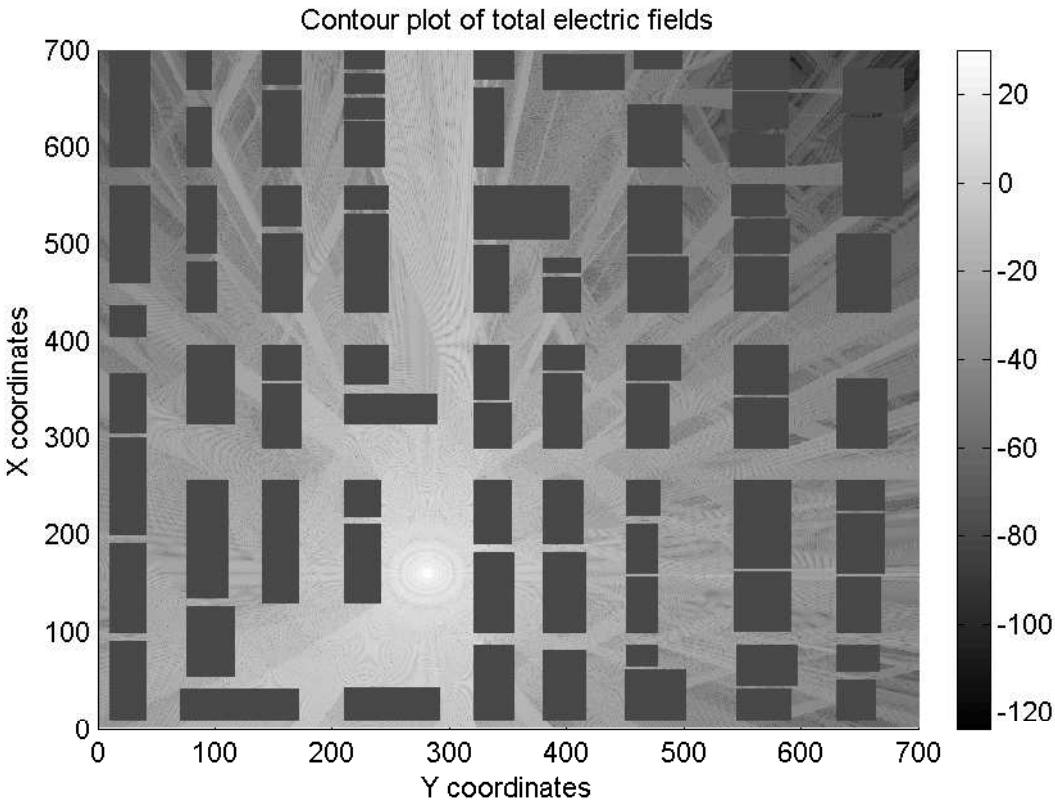

Figure 7: Field strength throughout urban region at 2GHz. Transmitter at $(270, 150, 3)$

References

[Agelet, 1997] Agelet (1997). Fast ray tracing for microcellular and indoor environments. *IEEE Transactions on Magnetics*, 33:1484–1487.

[Balanis, 1989] Balanis, C. (1989). *Advanced Engineering Electromagnetics, pp. 743-850.* Wiley.

[Coinchon, 2002] Coinchon (2002). The impact of radio propagation predictions on urban umts planning. In *Proceedings of the 2002 International Zurich Seminar on Broadband Communications.*

[Kouyoumjian, 1974] Kouyoumjian (1974). A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface. *Proc. IEEE*, 62:1448–1461.

[Schettino, 2007] Schettino (2007). Efficient ray tracing for radio channel characterization of urban scenarios. *IEEE Transactions on Magnetics*, 43:1305–1308.

[Stepanov, 2008] Stepanov (2008). On the impact of a more realistic physical layer on manet simulation results. *Ad hoc networks*, 6:61–78.

[Walfisch, 1988] Walfisch (1988). A theoretical model of uhf propagation in urban environments. *IEEE Transactions on Antennas and Propagation*, 36:1788–1796.