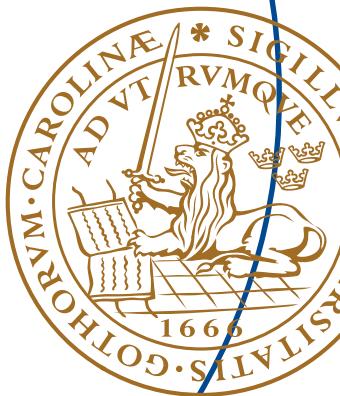


Master's Thesis

# Efficient Ray Tracing Simulation

Xuhong Li

Department of Electrical and Information Technology,  
Faculty of Engineering, LTH, Lund University, May 2014.  
In cooperation with Forschungszentrum Telekommunikation Wien GmbH



LUND UNIVERSITY AND FORSCHUNGSZENTRUM  
TELEKOMMUNIKATION WIEN GMBH

## **Efficient Ray Tracing Simulation**

---

**Xuhong Li**

**Supervisor: Mingming Gan**

**Examiner: Fredrik Tufvesson**

**Lund 2014**



# Abstract

Ray Tracing (RT) is a geometrical optics approach which determines most propagation paths that can propagate from transmitter (Tx) to receiver (Rx). It has been proved to be an accurate and versatile propagation prediction tool for both indoor and outdoor scenarios. As other deterministic channel models, RT requires exact knowledge of the geometrical and electromagnetic description of the relevant environment. Hence, the accuracy of RT comes at the cost of high computational complexity, which directly scales with the number of propagation paths considered.

To date, the developed RT tool includes not only specular reflection, transmission through dielectric blocks and diffraction, but also diffuse scattering mechanisms. The large number of paths involved in RT makes the computation workload very heavy. In order to accelerate the execution of a current three dimensional (3-D) RT model, which is implemented in MATLAB, the code is optimized and time-consuming functions are converted to MEX functions by using MATLAB Coder. The speeding up effort in the thesis focuses on the reflection and diffuse scattering calculations. Compared with the original code, the simulation time of the revised Matlab code is significantly reduced. For example, about ninety percent time consumption is saved relative to the original RT for an 8-block scenario.

The diffuse scattering components constitute a high proportion of all the propagation paths, which is closely related to the search for the diffuse scattering tiles. For reducing the simulation time of the original RT further, an efficient approach to generate diffuse scattering tiles is explored. In the original RT model, each surface is recursively divided until all segments fulfill the far-field condition. A substantial number of low-energy scattering paths might be produced due to the over segmentation problem. Moreover, the other approaches mentioned by relevant papers are usually specific to certain cases. We proposed an effective approach, which can be used as a general procedure to subdivide rough surfaces. Scattering tiles are generated on each surface with full consideration of the random characteristic of diffuse components. Besides, the proper tile size can be determined based on the system bandwidth, which makes this approach site independent and widely applicable in different scenarios. Another approach to calculate the tile size based on the far-field condition is also provided and proved to be efficient as well. Simulation results of the proposed algorithm are compared with the original RT model for the purpose of validation. Parameters such as power delay profile(PDP), delay spread(DS), angle spread(AS) are analyzed as the criteria for the performance assessments. With the proposed algorithm, lower computational complexity and less simulation time can be achieved without degradation of the prediction accuracy.



# Acknowledgments

First and foremost I would like to thank Forschungszentrum Telekommunikation Wien GmbH (FTW) for providing me the great opportunity to do my master thesis. It has been very educational and a perfect summary for my master study of wireless communication.

My most sincere appreciation goes to my supervisor MSc. Mingming Gan. During the whole process, we worked closely. She showed me not only handy skills of conducting research but also good working ethics and personalities leading to professional success. It was great fun and pleasure to work under her mentoring.

I am extremely grateful to Dr. Thomas Zemen, for his helpful discussions and great support in this project.

Special thanks for Prof. Dr. Fredrik Tufvesson, my examiner from the department of Electronics and Information Technology, Lund University for his support and interest in this project.

I would like to thank my great working colleagues, MSc. Markus Hofer, MSc. Sanda Drakuli and MSc. Zhinan Xu, for your kindness and the wonderful time we spent together.

Finally, I specially thank my dear families and friends, for their encouragement and tremendous support.



# List of Abbreviations

<b>2-D</b>	Two Dimensional
<b>3-D</b>	Three Dimensional
<b>AS</b>	Angle Spread
<b>CIR</b>	Channel Impulse Response
<b>CPU</b>	Central Processing Unit
<b>DS</b>	Delay Spread
<b>ER</b>	Effective Roughness
<b>GO</b>	Geometrical Optics
<b>IBS</b>	Incident Shadow Boundary
<b>IOs</b>	Interacting Objects
<b>LOS</b>	Line-of-Sight
<b>MS</b>	Mobile Station
<b>MRE</b>	Mean Relative Error
<b>MSE</b>	Mean Square Error
<b>NLOS</b>	Non Line-of-Sight
<b>PDP</b>	Power Delay Profile
<b>RT</b>	Ray Tracing
<b>Rx</b>	Receiver
<b>Tx</b>	Transmitter
<b>UWB</b>	Ultra Wideband



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Brief Background . . . . .	1
1.2	Related Previous Work . . . . .	2
1.3	Contributions of This Thesis . . . . .	3
1.4	Thesis Outline . . . . .	4
<b>2</b>	<b>Ray Tracing Channel Model</b>	<b>5</b>
2.1	Ray Tracing . . . . .	5
2.2	Propagation Mechanisms . . . . .	6
2.2.1	Line of Sight . . . . .	6
2.2.2	Reflection . . . . .	6
2.2.3	Diffraction . . . . .	8
2.2.4	Transmission . . . . .	9
2.2.5	Diffuse Scattering . . . . .	9
2.3	Reflection Path Calculation . . . . .	11
2.4	Test Environment . . . . .	13
<b>3</b>	<b>Speedup of 3-D RT</b>	<b>15</b>
3.1	Introduction of MATLAB Coder . . . . .	15
3.2	Speedup of 3-D RT Model . . . . .	17
3.2.1	Acceleration of Reflection Components Calculation . . . . .	17
3.2.2	Effective Construction Function . . . . .	21
3.2.3	Acceleration of Subdivision Algorithm . . . . .	24
3.3	Simulation Results Comparison . . . . .	25
<b>4</b>	<b>New Algorithm to Generate Diffuse Scattering Tiles</b>	<b>27</b>
4.1	Subdivision Based on Far-Field Condition . . . . .	27
4.2	New Algorithm to Generate Scattering Tiles . . . . .	29

## Contents

---

4.2.1	Scattering Tiles Generation . . . . .	29
4.2.2	Proper Tile Size . . . . .	31
<b>5</b>	<b>Simulation Results and Comparison</b>	<b>33</b>
5.1	Simulation Configuration . . . . .	33
5.2	Numerical Results and Analysis . . . . .	33
5.2.1	Time-Variant Transfer Function Comparison . . . . .	34
5.2.2	PDP comparison . . . . .	36
5.2.3	Delay Spread Comparison . . . . .	38
5.2.4	Angle Spread Comparison . . . . .	39
5.2.5	Computational Complexity Comparison . . . . .	42
<b>6</b>	<b>Conclusions</b>	<b>43</b>
<b>7</b>	<b>Future Work</b>	<b>45</b>

# Chapter 1

## Introduction

---

*C*hapter 1 focuses on the motivation of this thesis. Section 1.1 gives the brief introduction of background knowledge about channel models. Related previous work is introduced in Section 1.2. The contributions and outlines of this thesis are finally presented in Section 1.3 and 1.4.

### 1.1 Brief Background

In wireless communications, system performance is strongly affected by the radio channel properties, which can be described mathematically and converted into generic simulation models. These channel models are widely needed for the design, simulation and planning of wireless systems [5]. Generally speaking, there are three fundamental approaches to channel modeling: deterministic, stochastic and geometry-based stochastic [3].

In many circumstances, adopting statistical models is quite straightforward due to their low computational complexity assuming that it is too complicated to describe all reflection, diffraction, and diffuse scattering processes. Moreover, it is often preferable to describe the statistics of the channel impulse response with a certain power delay profile, Doppler power spectral density, angle-of-arrival profile etc [4]. However, due to the lack of physical background, statistical models only apply with good results in the environments, which are very close to the one where the relevant parameters have been inferred from.

For geometry-based stochastic channel models (GSCM), scatterers are located according to certain statistical distributions or at physically realistic positions. Then their contributions are determined using very simplified ray-tracing methods and summed up at the receiver side [6]. GSCMs have been proven to be well suited for non-stationary environments.

Unlike statistical channel models, deterministic physical channel models are location-specific approach which use geographical and electromagnetic information for a deterministic solution of Maxwell's equation or some approximation thereof [5]. Hence, such models could provide highly accurate prediction results. Unfortunately, they require a precise description of the environment which results

in massive computational complexity. They are particularly suited for short-term analysis and dynamic modeling in typical reduced configurations.

In general, statistical channel models are more applicable for the design and comparison of different systems, while deterministic channel models are preferable for network planning and system deployment [5]. Furthermore, these two approaches can be combined to improve the efficiency of a model.

In recent years, ray-tracing(RT) based channel models have received a lot of attention due to their accurate and versatile prediction capability both in indoor and outdoor scenarios. Conventional RT only accounts for specular reflection and diffraction. To date, advanced RT tools also involve transmission and diffuse scattering, which has been proved to be important contributions in determining time and angle dispersion of the propagation channel [11]. Most of the propagation paths connecting Tx and Rx are determined. The key to any robust RT model is an efficient algorithm for finding the propagation paths. Compared with two-dimensional (2-D) models, RT is more preferable to built 3-D models involving more significant interactions. Finally, the parameters such as field strength and transfer functions can be calculated [14] using RT techniques.

A RT model requires exact knowledge of the relevant environment and the number of propagation paths involved are tremendous. Consequently, the accuracy of RT comes at the cost of a high computational complexity. To sum up, the widespread adoption of RT models is still limited due to its high complexity and massive calculation time. Accelerating the simulation of RT is quite essential.

## 1.2 Related Previous Work

RT techniques were primitively used in broadcast television transmission [18]. After many years of development, RT has been greatly enhanced and widely used in predicting and analyzing wireless channel characteristics. The early approaches [7] and some simplified RT embedded in geometry based channel models [4] were mainly 2-D (horizontal plane) and only specular reflection and diffraction were involved. However, several significant interactions are missed in such models. For example, contributions from the ceiling or ground reflections are ignored in indoor scenarios. More recently, some new features are further involved in advanced RT models, such as 3-D consideration of the buildings and significant objects in relevant scenarios, transmission and diffuse scattering mechanisms [2]. Hence, the prediction capability of channel parameters in power delay profile, delay spread, especially the angular spread are greatly improved.

Advanced 3-D RT models can be used to characterize narrowband, wideband and even ultra-wideband (UWB) wireless systems [13]. Moreover, different antenna patterns can be easily integrated.

High computational complexity is the main restrictive aspect of 3-D RT. Re-

cently, some techniques aimed at solving this problem have been proposed. Due to most of the RT models are based on image theory, some acceleration techniques generally performed by preprocessing of the vector database [12]. Another approach to reduce simulation time is to combine ray launching with RT and propagation paths are separately considered in vertical and horizontal planes. Our speedup effort of the presented 3-D RT focuses on both implementation and algorithm aspects.

The calculation of diffuse scattering components is directly related to the scattering model and subdivision algorithm used. Models for scattering of electromagnetic waves from rough surfaces have been proposed in previous work. Models mentioned in [15] are based on “Kirchhoff approximation”, which only takes into account surface roughness deviations with Gaussian statistics and a limited curvature. Such theoretical models are not very suitable for real building walls [9]. A simple “effective roughness” (ER) model for modeling diffuse scattering from building walls is proposed in [10]. A sort of effective roughness is associated with each wall, which not only takes into account real surface roughness but also the wall irregularity effect in a mean, statistical way. A 3-D RT model including this ER scattering model [9] shows a sensible improvement in prediction accuracy compared with the conventional RT. However, the ER model simply assumes that the scattering radiation lobe has its maximum in the direction perpendicular to the wall according to the “Lambertian Model” without considering the direction of incidence.

Some approaches to generate scattering tiles are also mentioned in relevant papers. They are usually specific to certain cases and the universality in different scenarios is not studied. The simplest approach is to divide the target surface into segments of area  $\Delta S$ , which should be small enough to represent the geometry of the room with reasonable accuracy [16]. The center frequency is also considered in some cases to decide the dimension of segments. In [17], there are 400 scattering tiles within the same size placed around each specular reflection interaction point. In the presented 3-D RT [2], each rough surface is recursively divided until all segments fulfill the far-field condition.

### 1.3 Contributions of This Thesis

For accelerating the original 3-D RT model which is implemented in MATLAB, we use MATLAB Coder to convert time-consuming functions into MEX functions. Besides, several techniques are proposed and integrated into RT.

The main contributions of this part are as following:

- We propose several approaches such as “last layer check” and “surfaces on the same plane”, to remove useless image points before path calculation and avoid repeated calculation for the reflection case.

- We develop an efficient image-tree construction function “Effconstruct” and solve the “Out of Memory” problem which exists in many RT models for the reflection case.
- We decompose the original subdivision algorithm into three sub functions and the recursion structure is no long used for the diffuse scattering case.
- We convert time-consuming functions into MEX functions with MATLAB Coder, which significantly reduces the simulation time.

In the second part of this thesis, we propose a general and simplified approach for diffuse scattering tile generation, which can be used in different indoor scenarios. The main contributions of this part are summarized as follows:

- The random characteristic of diffuse components is fully considered when generating scattering tiles on building surfaces. Proper tile size can be easily calculated using the system bandwidth or the far-field condition.
- The simulation results, such as PDP, DS and AS are compared with the original RT model and analyzed as the criteria for the performance assessments.
- The computational complexity and simulation time needed for the diffuse scattering calculation are significantly reduced without degradation of the prediction accuracy.

## 1.4 Thesis Outline

This thesis report is organized as follows. In Chapter 2, basic principle of the original RT is outlined and the involved propagation mechanisms are introduced. The 8-block test environment, based on which the simulation results are obtained in this thesis is also presented in this chapter. In Chapter 3, MATLAB Coder is briefly introduced firstly. Then, the details about the speedup work of the original RT is described. Chapter 4 gives detailed description of the original and proposed subdivision algorithms for the diffuse scattering contributions in RT. Finally, the conclusion and future work are presented in Chapter 5 and 6.

# Chapter 2

## Ray Tracing Channel Model

---

**C**hapter 2 presents the 3-D RT model used in the presented work. Propagation mechanisms are described in detail in Section 2.1. Theories and structure of RT model are introduced in Section 2.2. Finally, the test environment, based on which the simulation results are obtained in this thesis is presented in Section 2.4.

### 2.1 Ray Tracing

All our work in this project is based on an existing 3-D RT implementation, which has been proved to be accurate by comparing the simulation results with measurement data [1]. Relevant objects in the scenarios are modeled as rectangular solid blocks which are made of different materials. Dielectric properties of these materials influence the field strength of propagation paths. The complex effective relative permittivity of a certain material can be defined by [5]:

$$\varepsilon_{r,\text{eff}} = \varepsilon_r - j \frac{\sigma}{2\pi f_c \varepsilon_0}, \quad (2.1)$$

where  $\varepsilon_0 = 8.854 \cdot 10^{-12}$  Farad/m is the vacuum dielectric constant,  $\sigma$  is the conductivity of the material and  $f_c$  is the center carrier frequency. Although this definition is strictly valid for a single frequency, it can actually be used for a narrowband system, where the bandwidth is smaller than 500 MHz.

In 3-D RT, not only reflection and diffraction, but also transmission and diffuse scattering mechanisms are involved. Different propagation paths are denoted with different colors as shown in Fig. 2.1. It should be noted that reflections from floors and ceilings represent very important role among the propagation paths [2]. Hence, the 3-D RT model is capable to provide more accurate prediction results compared with the conventional 2-D simplified model.

By using knowledge of the geometrical and electromagnetic description of the scenarios, RT can find most multipath components from Tx to Rx, the majority of which are reflection and diffuse scattering contributions. Finally, field strength and transfer functions can be calculated.

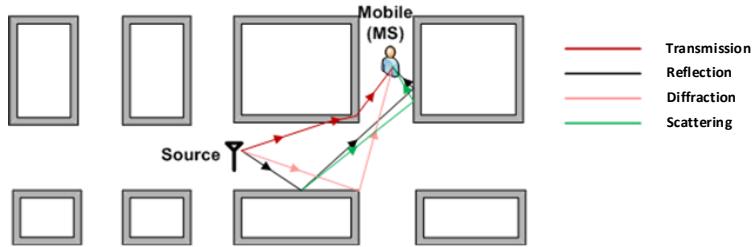


Figure 2.1: RT model.

## 2.2 Propagation Mechanisms

We briefly introduce the propagation mechanisms involved in RT and how to find corresponding paths in this section.

### 2.2.1 Line of Sight

For wireless communication, free space propagation is the simplest case in which signal propagates directly from Tx antenna to Rx antenna. The received line-of-sight (LOS) electric field  $E_{\text{LOS}}$  can be easily obtained as [8], [14]:

$$E_{\text{LOS}} = [\bar{g}_{\text{LOS}}^{\text{R}}]^* \cdot E_0 \cdot \bar{g}_{\text{LOS}}^{\text{E}} \cdot \frac{e^{-jkd}}{d}, \quad (2.2)$$

where  $j$  is the imaginary unit,  $\bar{g}_{[\cdot]}^{\text{E}}$  and  $\bar{g}_{[\cdot]}^{\text{R}}$  are the complex vectors accounting for the Tx/Rx antenna polarisation and antenna gain in the direction of the transmitted/received wave,  $[\cdot]^*$  designates the complex conjugate,  $E_0$  is the emitted field,  $k = \frac{2\pi}{\lambda}$  is the wave number where  $\lambda$  is the wavelength and  $d$  represents the distance between Tx and Rx.

### 2.2.2 Reflection

Dielectric and conducting objects exist in most propagation environments. If surfaces of these objects are smooth, electromagnetic waves are specularly reflected (Fig. 2.2) and part of the energy is lost due to the transmission into the object. Furthermore, if the standard deviation of the surface roughness is significantly smaller than the wavelength, incident waves can still be seen as specularly reflected [14].

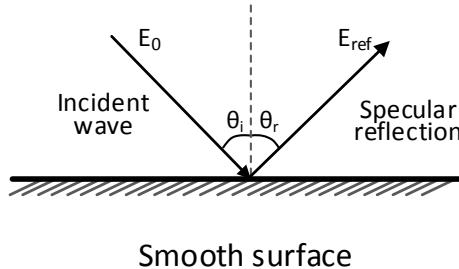


Figure 2.2: Specular reflection from a smooth surface.

According to Snell's law, the angle of incidence  $\theta_i$  is the same as the reflection angle  $\theta_r$  for the specular reflection case [5]. If the reflection point belongs to the dimension of the surface, the first-order reflected field at the receiving point  $P$  can be expressed as [14]

$$E_{\text{ref}}(P) = [\bar{g}_{\text{ref}}^{\text{R}}]^* \cdot \bar{\mathbf{R}} \cdot E_0 \cdot \bar{g}_{\text{ref}}^{\text{E}} \cdot \frac{s}{s'} e^{-jks'}, \quad (2.3)$$

where  $s$  is the path length from the Tx to the reflection point,  $s'$  is the path length from the reflection point to the point  $P$ ,  $\bar{\mathbf{R}}$  is the Fresnel dyadic coefficient, which is a  $3 \times 3$  matrix featured by the following expressions [14]:

$$\begin{aligned} \bar{\mathbf{R}} &= R_h \bar{e}_{\text{inc},\parallel} \bar{e}_{\text{ref},\parallel} + R_s \bar{e}_{\perp} \bar{e}_{\perp}, \\ R_h &= \frac{\varepsilon_{r,\text{eff}} \cos \theta_i - \sqrt{\varepsilon_{r,\text{eff}} - \sin^2 \theta_i}}{\varepsilon_{r,\text{eff}} \cos \theta_i + \sqrt{\varepsilon_{r,\text{eff}} - \sin^2 \theta_i}}, \\ R_s &= \frac{\cos \theta_i - \sqrt{\varepsilon_{r,\text{eff}} - \sin^2 \theta_i}}{\cos \theta_i + \sqrt{\varepsilon_{r,\text{eff}} - \sin^2 \theta_i}}, \end{aligned} \quad (2.4)$$

where  $\bar{e}_{\text{inc},\parallel}$  and  $\bar{e}_{\text{ref},\parallel}$  are the unit vectors parallel to the planes of incident and reflection. Because the planes of incidence and reflection coincide,  $\bar{e}_{\text{inc},\perp} = \bar{e}_{\text{ref},\perp} \triangleq \bar{e}_{\perp}$  where  $\bar{e}_{\text{inc},\perp}$  and  $\bar{e}_{\text{ref},\perp}$  are the unit vectors perpendicular to the planes of incidence and reflection, respectively.

### 2.2.3 Diffraction

Real objects like walls and furniture do not have infinite sized surfaces, diffraction could happen on the edges of objects due to the nature of electromagnetic radiation. Huygen's principle explains diffraction in a very straightforward way that each point of a wave front can be considered as the source of a spherical wave and the superposition of the spherical waves results in another homogeneous plane wave [5]. The total incident field can be expressed by Fresnel integral. In real environments, it is more properly to present objects as wedge structure. The geometry for wedge diffraction is shown in Fig. 2.3 [14].

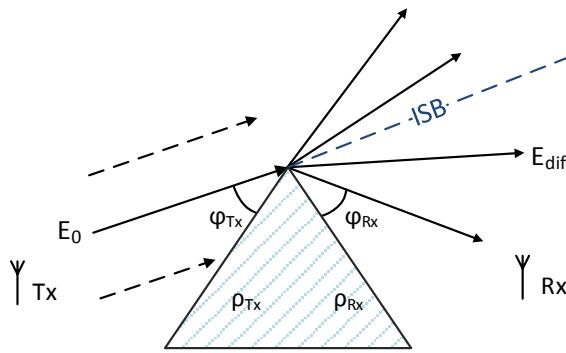


Figure 2.3: Geometry for wedge diffraction.

As can be seen from Fig. 2.3, below the incident shadow boundary (ISB), no incident and reflected waves exist, the received field strength totally depends on diffraction. The diffracted field can be expressed as [5], [14]:

$$E_{\text{diff}} = [\bar{g}_{\text{diff}}^{\text{R}}]^* \cdot E_0 \cdot \bar{g}_{\text{diff}}^{\text{E}} \cdot D(\varphi_{\text{Tx}}, \varphi_{\text{Rx}}) \cdot A(d_{\text{Tx}}, d_{\text{Rx}}) e^{-jkd_{\text{Rx}}}, \quad (2.5)$$

where  $D(\varphi_{\text{Tx}}, \varphi_{\text{Rx}})$  is the diffraction coefficient which depends not only on the diffraction angles  $\varphi_{\text{Tx}}$  and  $\varphi_{\text{Rx}}$ , but also the boundary conditions, i.e. the reflection coefficients  $\rho_{\text{Tx}}$  and  $\rho_{\text{Rx}}$ ,  $A(d_{\text{Tx}}, d_{\text{Rx}})$  is the geometry factor determined only by the distances from the wedge to Tx and Rx, which are denoted as  $d_{\text{Tx}}$  and  $d_{\text{Rx}}$ ,  $e^{-jkd_{\text{Rx}}}$  is the phase factor.

The law of diffraction is formulated by Keller as: "A diffracted ray and the corresponding incident ray make equal angles with the edge at the point of diffraction; they lie on opposite sides of the plane normal to the edge at the point of diffraction" [14]. If the propagation of the incident and diffracted waves are known, the

location of the diffraction point can be easily calculated by simple vector calculation.

### 2.2.4 Transmission

Transmission is very important for wave propagation, especially in indoor scenarios. The calculation of transmission through blocks is merged into the calculation of other propagation mechanisms. For simplicity, the propagation direction of the rays which experienced transmission in 3-D RT are assumed to be unchanged. The electric field after the transmission can be evaluated as [2]:

$$E_{\text{tr}} = [\bar{g}_{\text{tr}}^{\text{R}}]^* \cdot E_0 \cdot \bar{g}_{\text{tr}}^{\text{E}} \cdot T_{\text{in}} \cdot T_{\text{out}} \cdot \frac{e^{-jk_0(d_1+d_3)} e^{-jk_{\text{tr}}d_2}}{d_1 + d_2 + d_3}, \quad (2.6)$$

where  $T_{\text{in}}$  and  $T_{\text{out}}$  are the dyadic transmission coefficients,  $d_1$  is the distance from the transmitting point to the incoming transmission point,  $d_3$  is the distance between outgoing penetration point and the receiver point,  $d_2$  is the distance between the two transmission points,  $k_0$  and  $k_{\text{tr}}$  represent the propagation constants in free space and inside blocks, where the transmission occurs.

### 2.2.5 Diffuse Scattering

Due to rough surfaces in real scenarios, propagation paths are often scattered in a wide range of directions but not following the geometrical optics (GO) rules, as shown in Fig. 2.4. The scattering contribution has a significant impact on the “tail” of the power delay profiles, where multipath components arrive at the Rx with large delays [9].

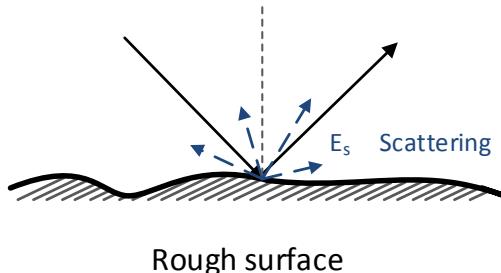


Figure 2.4: Diffuse scattering on rough surface.

The scattering model used in RT has a direct influence on the field strength of diffuse components. The directive scattering model used in the original RT is based on the assumption that the scattered rays originate from the centers of diffuse scattering tiles and steered towards the direction of the specular reflection [11], as shown in Fig. 2.5.

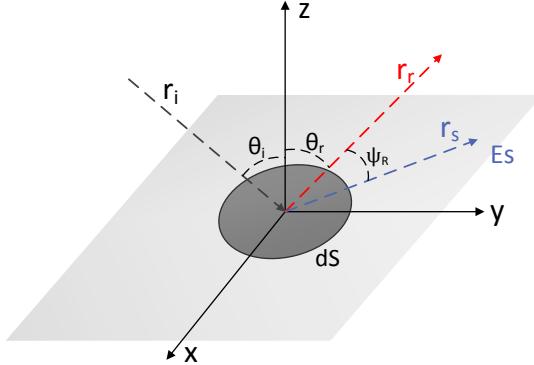


Figure 2.5: A generic surface element producing reflection and diffuse scattering.

The complex scattered field can be expressed as:

$$E_S = |\bar{E}_S| \exp(j\varphi_s), \quad (2.7)$$

where  $\varphi_s$  is the random phase between  $[0, 2\pi]$ . The power density carried by the scattered wave is proportional to  $\bar{E}_S$ , which is the intensity of the scattered field.  $\bar{E}_S$  can be expressed as [11]:

$$\bar{E}_S = E_{S0} \left( \frac{1 + \cos \varphi_R}{2} \right)^{\frac{\alpha_R}{2}}, \quad \alpha_R = 1, 2, \dots, N, \quad (2.8)$$

where  $\varphi_R$  is the angle between the directions of scattered wave and specular reflection. The exponent  $\alpha_R$  is related to the width of the scattering lobe. The greater  $\alpha_R$ , the narrower the scattering lobe. The maximum amplitude  $E_{S0}$  can be computed by the following expression [11]:

$$E_{S0} = [\bar{g}_S^R]^* \cdot E_0 \cdot \bar{g}_S^E \cdot \frac{SK}{r_i r_s} \cdot U \cdot \left( \frac{dS \cos \theta_i}{F_{\alpha_R}} \right)^{\frac{1}{2}}, \quad (2.9)$$

where  $S$  is the scattering coefficient,  $U = |E_r|/|E_i|$  where  $|E_r|$  and  $|E_i|$  are the norms of the reflected and incident fields on the surface element  $dS$ ,  $F_{\alpha_R}$  is a

function related to  $\varphi_R$ ,  $K$  is a constant depending on the amplitude of the impinging wave,  $r_i$  and  $r_s$  represent the distances between the scattering point and the source/receiving point. From (2.9), it can be seen that the overall scattered power is directly proportional to the surface area  $dS$ .

The subdivision algorithm involved in diffuse scattering calculation has an important impact on the size and the number of the generated diffuse scattering tiles. Approaches for subdivision of rough surfaces are introduced in more detail in Chapter 4.

## 2.3 Reflection Path Calculation

An efficient algorithm to find the propagation paths is the key point for successful implementation of RT. In this section, we focus on the algorithm to calculate reflection paths, which constitute a high proportion of all propagation paths. The implementation of the path calculation algorithm consists of two main steps: the visibility algorithm and the backtrack procedure which helps to determine the reflection paths. The leading task of the visibility algorithm is to create an image tree which has a layered structure. The image tree consists of nodes and branches, which indicate image points and potential reflection paths respectively. Then, the backtrack procedure is performed from the Rx to Tx.

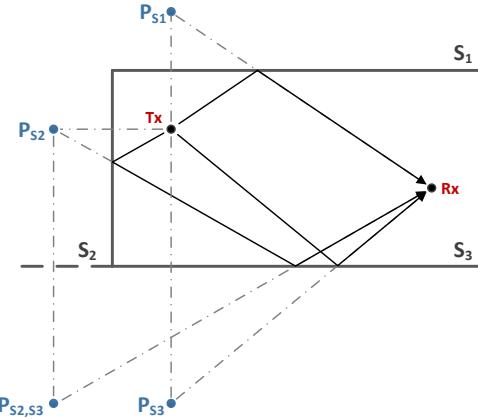


Figure 2.6: The image principle

The image principle is the foundation of the visibility algorithm. A simple case is demonstrated in Fig. 2.6. If the coordinates of the Tx are known, the 3-D coordinates of image points  $P_{S1}$ ,  $P_{S2}$  and  $P_{S3}$  corresponding to surfaces  $S_1$ ,

$S_2$  and  $S_3$  are to be found easily by using the image principle. The produced image points are further treated as new sources for the next interaction, such as  $P_{S2,S3}$ . Then backtrack from Rx, single-order and second-order reflection paths are determined. Theoretically, reflection paths with any interaction order can be traced by this principle.

Starting from the Tx, which is the root of the tree, the image tree is recursively built by using the image principle. The structure is clearly shown in Fig. 2.7.

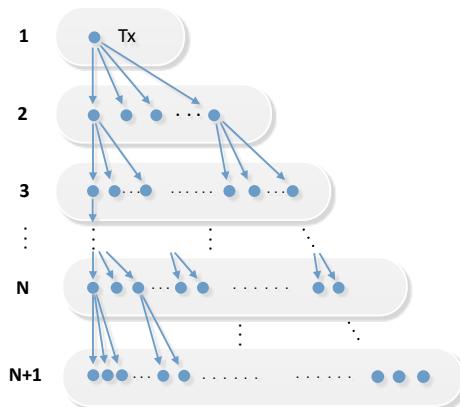


Figure 2.7: Structure of image tree

Considering all the visible surfaces for the Tx, image points are calculated and stored in the second layer. The lower layer nodes contain the image points that are corresponding to upper layer nodes. This procedure is repeated until the maximum reflection order is reached. Using the backtrack procedure from the Rx, if the Tx is reachable by the node on the last layer, this branch is saved as a valid propagation path.

In real cases, the visibility relations among different nodes strongly depends on the parameter settings for each block. The details will be introduced in Chapter 3. The construction of the image tree may require a great amount of central processing unit (CPU) time. It also can be seen from the image tree structure, that the node number in each layer increases exponentially with the reflection order. Hence, the “Out of memory” problem arises for RT especially when higher reflection order and large test environments are involved. Effort regarding these two problems are also presented in Chapter 3.

## 2.4 Test Environment

Unless otherwise specified, the simulation results in this thesis are obtained based on the scenario described in this section. The size of this indoor scenario is  $13.25\text{ m} \times 29\text{ m} \times 4.6\text{ m}$ . The 8 blocks shown in Fig. 2.8, together with the ceiling and the floor are involved. The blocks are made of different materials, which are denoted by different colors. The materials sketched with grey, red and yellow represent concrete, brick and wood walls, respectively. Some propagation paths are represented by the visualized lines with different colors. Transmission is merged into the calculation of other propagation mechanisms.

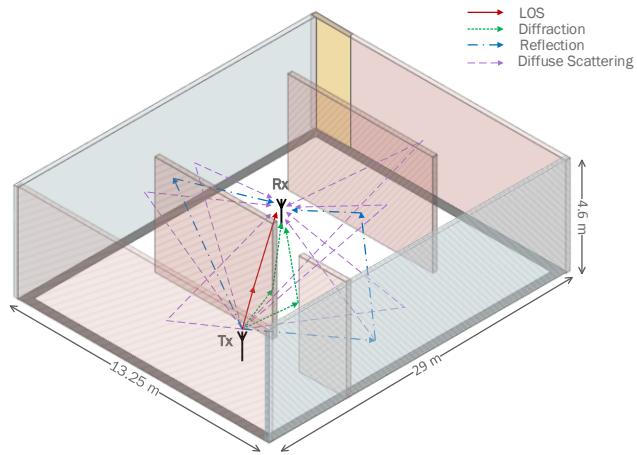


Figure 2.8: Environment



# Chapter 3

## Speedup of 3-D RT

**T**

*This chapter shows how to accelerate 3-D RT simulation using MEX functions. A basic introduction for MATLAB Coder is given in Section 3.1. Then the procedure to generate MEX functions of the original MATLAB code is described in Section 3.3. The speedup work focuses on the two most time-consuming parts: reflection and diffuse scattering calculations. The final Section 3.3 is about the simulation results and efficiency comparison with the original RT model.*

### 3.1 Introduction of MATLAB Coder

MATLAB Coder enables engineers to develop algorithms in MATLAB, then the MATLAB code can be automatically converted to standalone C and C++ code which is portable and readable. The generated code can be used for accelerating computationally intensive simulations by replacing the MATLAB code with automatically generated MATLAB executables (MEX function) that call the compiled C/C++ code [23]. Also, MATLAB algorithms can be integrated into other software as a compiled library component and verified on an embedded processor. The general workflow of MATLAB Coder is shown in Fig. 3.1 [23].

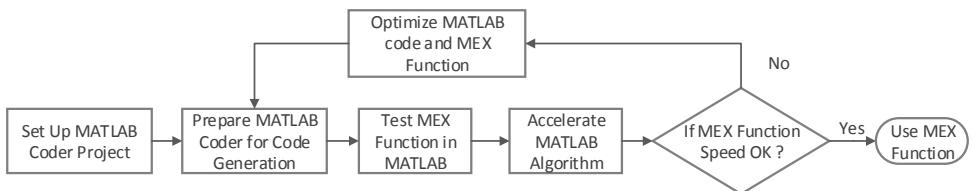


Figure 3.1: Workflow for MATLAB Coder.

To generate C/C++ code or MEX functions, the first step is to install the MATLAB Coder product and a C/C++ compiler in the computer and to set up the compiler by using the ‘mex -setup’ command. The second step is to create a MATLAB Coder project for the MATLAB function. Using the project user interface, we add types for input parameters of the function and entry-point files

which are called directly from MATLAB. If global variables are used in the MATLAB code, global type definition and initial values for each parameter should be stated before building the project. Selecting the “Redirect entry-point calls to MEX function” option enables the MATLAB Coder software to call the generated MEX functions instead of calling the original MATLAB function.

Clicking the “Build” button, a MEX function can be generated. In the process, MATLAB Coder produces a report that identifies errors which must be fixed to make the MATLAB algorithm compliant for code generation. An iterative process between fixing errors and regenerating a MEX function is needed until the MATLAB algorithm is suitable for code generation. A complicated application may be composed of more than one MATLAB files. In this case, the ‘codegen’ command is needed for checking if the MATLAB code is suitable for code generation and MEX function generation [23]. Until now, the generated MEX function can not be used directly. The speed should be tested, if it is faster compared with original MATLAB code, the MATLAB code can be replaced with MEX function.

In order to generate efficient and standalone C/C++ code, some key points are listed below for consideration:

- Language features

MATLAB Coder supports a subset of core MATLAB language features, including program control constructs, functions, and matrix operations. The detailed list can be found in MATLAB Documentation [23]. Language features which are not included in the documentation will lead to errors during compilation.

- Variables and globales

A complete assignment to each variable, including type, dimension and initial values should be declared in the project file. Do not forget those used in the child functions. Dimensions of the input variables used in functions should be strictly the same as the dimensions declared in the project file.

- Memory

Before converting to a MEX function, enough memory space should be allocated to globals and variables. Dynamic memory allocation uses less memory but requires extra time to manage the memory. Better speed can be achieved by using static memory allocation, but the problem is higher memory usage [23].

- Speed

Loading large-size globals is very time-consuming, which may result in no efficiency improvement or even lower efficiency. Choosing a suitable C/C++ compiler instead of using the default compiler is helpful for improving the speed of the generated code. Disabling run-time checks also results in faster simulations [23].

## 3.2 Speedup of 3-D RT Model

The speedup work of the original 3-D RT model is done by utilizing MATLAB Coder to generate MEX functions of the time-consuming functions. The whole application is very complicated and composed of over thirty MATLAB files. Before code generation, we noticed that language features and function structures like cell array and recursion structure are not supported by MATLAB Coder. Proper revisement of original MATLAB code is the primary task.

The proposed techniques in this section aim at not only making the MATLAB algorithm compliant for code generation, but also solving the “Out of Memory” problem mentioned in Chapter 2. From the previous analysis, reflection and scattering mechanisms produce the majority of propagation paths, which means calculation of these two parts consumes most of the simulation time. From this point of view, the rest of this chapter is naturally divided into two parts.

### 3.2.1 Acceleration of Reflection Components Calculation

The key point of reflection calculation is to build up the image tree and to use the backtrack procedure to find all propagation paths. The techniques introduced in this section are based on the idea of reducing the number of image points and avoiding repeated calculation.

Before introducing techniques to accelerate the reflection calculation, relative background knowledge is given firstly. In order to perform accurate calculation, a detailed description of the test environment is needed for the RT model. All relevant objects in the environment, i.e., walls, ceilings and furniture are models as rectangular solid blocks and each block has 6 surfaces, each surface is determined by 4 points. 3-D coordinates and dielectric properties of these blocks are saved in an input database in a suitable format. Several important electromagnetic boundary conditions of one block are listed as below:

- 3-D coordinates of unit normal vectors of 6 surface
- 3-D coordinates of 8 block vertexes
- Diffuse scattering flag (if the block has scattering)
- Transmission flag (if the block allow transmission)
- Flag indicates if the block is a perfect electric conductor
- Permittivity and conductivity of each block

How the propagation rays interact on a certain block is determined by these conditions. For example, if diffuse scattering flag is set to 0, which means the sur-

faces are smooth and no scattering happens, only specular reflection, transmission and diffraction are considered when a propagation ray reaches this block.

The image tree is recursively built. When image points are calculated level by level, the corresponding image point number is directly related to geometrical description and electromagnetic properties of the environment. An example is given in Table 3.1

Layer	Node Number
1	1
2	12
3	178
4	2688
5	40964

Table 3.1: Number of nodes on each layer for an 8-block environment.

The data in Tab. 3.1 is from the simulation result of the test environment and the maximum reflection order is set to 4. Including the “root” Tx, the image tree totally have 5 layers. There are over forty thousand nodes on the last layer. However, only 142 reflection paths are found finally. Therefore, it can be seen that most of the nodes are useless for the final results, so that substantial CPU time is wasted. For more complicated scenarios where the number of blocks may increase up to dozens or hundreds and dimensions are much larger, the potential number of nodes is tremendous, and “out of memory” may happen during the simulation.

According to the analysis above, a few algorithms are proposed to remove useless nodes in advance to save computation time and reduce database size.

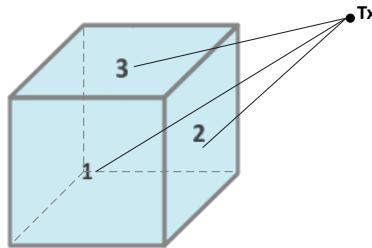


Figure 3.2: Geometry of visible surface algorithm.

The first algorithm is based on the electromagnetic property of a certain block and the geometric spatial relationship between the block and the source point, which means Tx or the image point on a previous layer. Wherever a source is

located in the 3-D space, 6 surfaces are all visible for the source point if transmission is allowed by this block, the propagation wave can penetrate the block and interact on the backside surfaces. However, if the block is a perfect conductor or transmission flag is set to 0, at most 3 surfaces can be “seen” by the source as shown in Fig. 3.2. In this case, the propagation wave is blocked and no interaction happens on the invisible surfaces.

When building up the image tree, image points corresponding to 6 surfaces are all calculated without considering the “visibility” in the original MATLAB code. The validity of each point is checked when tracking back from the Rx, and then useless image points are discarded. But during this process, unnecessary CPU time has already been spent.

Based on the spatial relationship, visible surfaces for source point can be calculated in advanced. The algorithm is named “Effsurfaces”. Six surfaces are numbered and there are totally 26 different spatial relationships to decide which surfaces are visible. The efficiency of the “Effsurfaces” algorithm highly depends on the electromagnetic properties of the test environment. If transmission is not allowed for most blocks in the environment, this algorithm is capable to tremendously remove useless nodes and reduce the database size. One drawback that can not be ignored is that this algorithm should be executed for each node, time consumption may even increase in the end. Therefore, efficiency should be checked before the algorithm is used in the RT model.

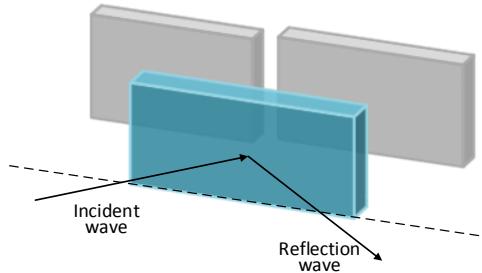


Figure 3.3: Remove invalid blocks in advance.

The second algorithm is based on the geometric spatial relationship between blocks, we name it as “Effblocks”. Unlike “Effsurfaces” which removes useless surfaces, “Effblocks” shown in Fig. 3.3. is capable to remove blocks which produce useless image points for the next interaction. According to the propagation direction of reflected wave, the next interaction could only occurs on the blocks in front of the blue block. The gray blocks behind should not be considered. Once the database of the environment is given, the “Effblocks” algorithm is only executed once and the result is stored in a matrix for reference.

It is easy to find that most of the nodes are contained in the last layer of the

image tree. The idea of the third method is to reduce the image points number in the last layer by checking if there is an actual crossover point on the corresponding surface. The principle of this “Last level check” algorithm is shown in Fig. 3.4.  $P_s$  is the image points on the last level. Draw a line between Rx and  $P_s$ , if there is no crossover point on the surface S, no propagation path will be found when tracking back from Rx. Hence,  $P_s$  should be deleted.

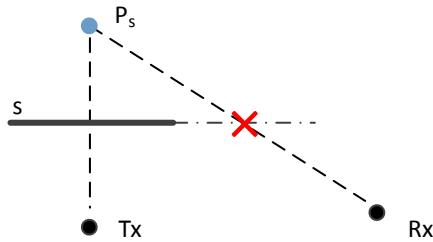


Figure 3.4: Principle of last layer check.

The previous three algorithms are all based on the consideration of reducing the size of the image tree by removing useless image points before path calculation. The idea of the fourth algorithm “Surfaces on the same plane” is avoiding repeated calculation according to the distribution regularities of image points. For a certain source point, the image points corresponding to surfaces which are on the same plane have the same 3-D coordinates. As shown in Fig. 3.5,  $P_{S1}$  and  $P_{S2}$  are image points corresponding to the same source point Tx. Due to surfaces  $S_1$  and  $S_2$  are on the same plane, coordinates of the two points are identical. Once one image point is calculated, the relevant information can be shared. Hence, repeated calculation is avoided.

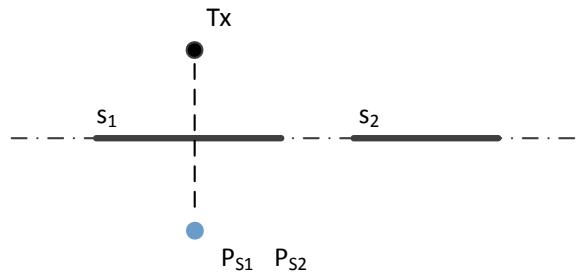


Figure 3.5: Principle of surfaces on the same plane.

The four algorithms introduced above are implemented and integrated into the original RT model separately. We found that due to algorithm complexity, the simulation runtime reduction results are not as ideally as originally thought. Only the “Last layer check” algorithm can remove considerable amount of image points with achieving obvious time consumption reduction. When using these algorithms, we need to compromise between execution efficiency and computational complexity. Further more, these algorithms did not change the recursion structure of the RT model, which is still not compliant for fast code generation.

### 3.2.2 Effective Construction Function

The “Effective Construction Function” is the most important change for the original RT model. It can be considered as the combination of image tree construction and path calculation.

Similar as the original construction algorithm, the new algorithm “Effconstruct” also starts from the root of the tree, which contains the Tx, but the image tree is no longer recursively built.

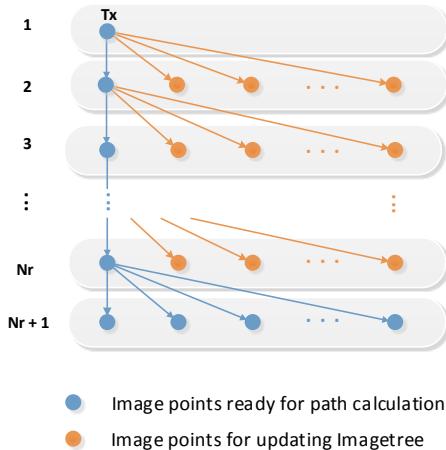


Figure 3.6: Principle of effective image tree construction.

Image points of the first node in each layer are calculated and stored in the next layer, until the maximum reflection order is achieved. The structure of the new image tree is clearly shown in Fig. 3.6. After the initial image tree is built, the validity of the nodes on the last layer is checked by utilizing the “Last layer check” algorithm mentioned above. During this process, some useless nodes are discarded, the remaining nodes in the last layer along with the first nodes in upper layers (blue solid circles) are prepared for the path calculation. Only the effective

propagation paths are saved. The processed nodes in layer  $N_r + 1$  and the first node in layer  $N_r$  are deleted. Then the image points of the second node in layer  $N_r$  are calculated and stored in the last layer. The image tree is continuously updated during the whole process until the last node on the second layer is used.

With those algorithms, the “recursion structure” problem is solved, but the MATLAB code is still not compliant for MEX function generation because of the improper data format. In the original RT model, information of nodes is saved in a cell array, named “trun”. An example is given in Fig. 3.7, where “face” shows the indexes of block and surface. 3-D coordinates of the image point are given in value of “point”. Totally 13 derived image points are stored in the next layer. “Previouslevel” indicates that this image point originates from the first point in the first layer.

trun{2, 1} <1x1 struct>	
Field	Value
face	[1;1]
point	[-2;19.2000;1.5500]
Nbfaces	13
PreviousLevel	[1,1]

Figure 3.7: An example of information in cell array “trun”.

A cell array is an indexed list of data with symbolic names, which makes it more readable and much easier for user to understand the structure of image tree. However, cell array is not supported by MATLAB Coder. In order to generate MEX functions to speedup the simulation, a more proper data format is needed.

A 3-D matrix is proved to be an ideal option for such a situation. Information of all nodes are stored in a 3-D matrix named “Mpath”. The dimension of “Mpath” is no longer unpredictable. According to maximum reflection order and block number, it is easy to predefine the dimension of “Mpath” as  $N_r \times N_f \times 5$ . Also, the new algorithm structure decides that less information is needed for each node, which is listed as:

- 1~2: indexes of block and surface.
- 3~5: coordinates of image point.

The first node in each layer is the default “Previouslevel” for the nodes in the next layer.

The “Effconstruct” algorithm is implemented in MATLAB and the code is fully supported for MEX function generation by MATLAB coder. “Out of memory” problem is completely solved as well. It should be noticed that the algorithm mentioned above aims to improve the RT model simulation efficiency, which does not change the accuracy of the final prediction results.

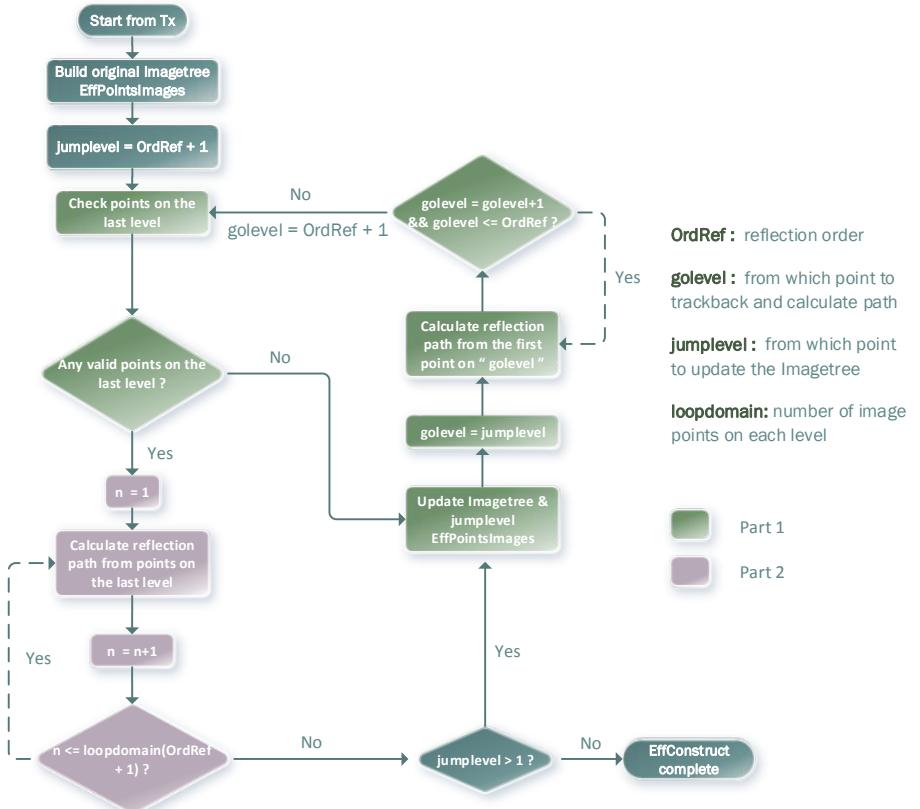


Figure 3.8: Flow chart of effective image tree.

The workflow about how the effective image tree algorithm works is shown in Fig. 3.8. The whole application is mainly composed of two parts. Part one is about the last layer check and the calculation of reflection paths with lower reflection orders. The aim of the second part is to calculate propagation paths with the maximum reflection order. Several flags, i.e., “golevel” and “jumplevel” are created for denoting where to update the image tree and where to calculate

paths, respectively.

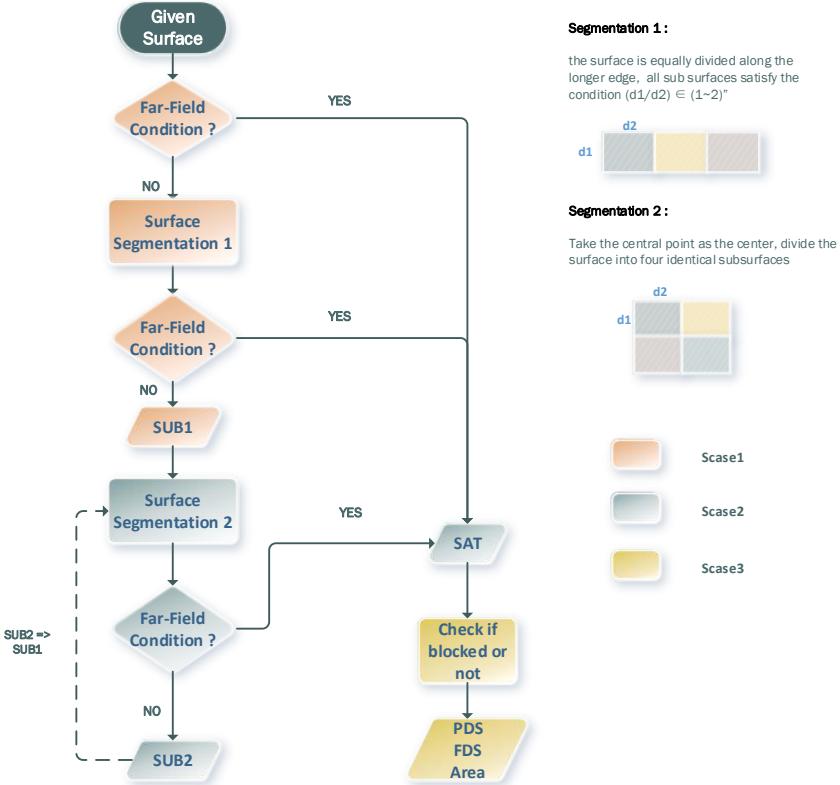


Figure 3.9: Flow chart of updated subdivision algorithm.

### 3.2.3 Acceleration of Subdivision Algorithm

The proposed algorithms in previous sections are about how to accelerate the calculation for reflection components. The following section is about the speedup work of the diffuse scattering part.

When calculating diffuse scattering, maximum two interactions are considered in the simulations. The three mechanisms involved are:

- Single scattering.

- Scattering before reflection.
- Scattering after reflection.

The whole calculation process for each mechanism falls into two main steps: tracing scattering paths and scattered field strength calculation by using a proper scattering model. Details are given in Chapter 2.

The scattering tile generation is the key point for tracing scattering paths. In the original RT model, subdivision based on the far-field condition algorithm is used to search for scattering tiles on block surfaces and this algorithm is the most time-consuming part during the whole simulation besides the reflection component calculation.

The original MATLAB code of the subdivision algorithm adopts a recursion structure, which is not supported by MATLAB coder. We decompose it into three subfunctions, which are named as “Scase1”, “Scase2” and “Scase3”. The scattering surface is firstly processed in “Scase1”. If the far-field condition is fulfilled, information of this surface is stored into the matrix “SAT” for further processing in “Scase3”. If not, the surfaces is equally divided along the longer edge until all segments satisfy the condition that the ratio between two edges  $d_1$  and  $d_2$  is not larger than two. The segments are further checked in “Scase2”. If the far-field condition is not fulfilled, the segments are divided into four identical sub surfaces. The qualified segments are sent to “Scase3”. The workflow is shown in Fig. 3.9.

The revised subdivision function is implemented with MATLAB code and the code is compliant for MEX function generation.

### 3.3 Simulation Results Comparison

In order to evaluate the efficiency improvement of the updated RT model in different scenarios, besides the 8-block environment, we also test the performance of the updated RT model in a more complex scenario within 34 blocks. To show the efficiency improvement intuitively, execution time of some functions are compared in Fig. 3.10 and Fig. 3.11, the blue bar represents the time needed for the original MATLAB code and the red bar indicates the time consumption of the revised code.

As seen from the figure, about 90% total time consumption is saved from the original RT for the 8-block scenario and 99% for the 34-block scenario. The most significant contributions come from reflection and diffuse scattering parts. The efficiency of the revised subdivision code is not improved as much as the revised reflection path calculation function.

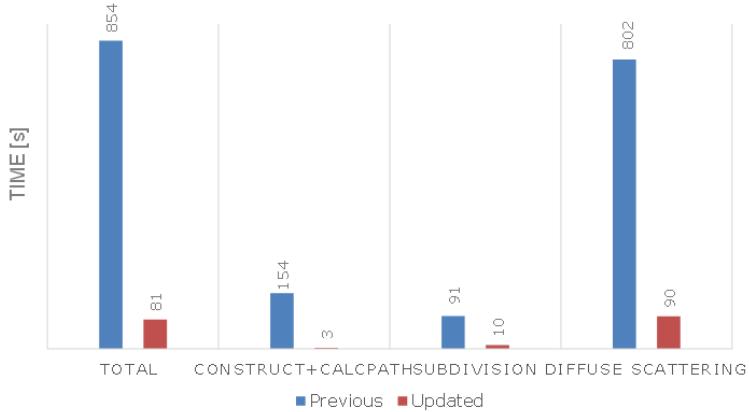


Figure 3.10: Simulation time comparation for 8-block environment.

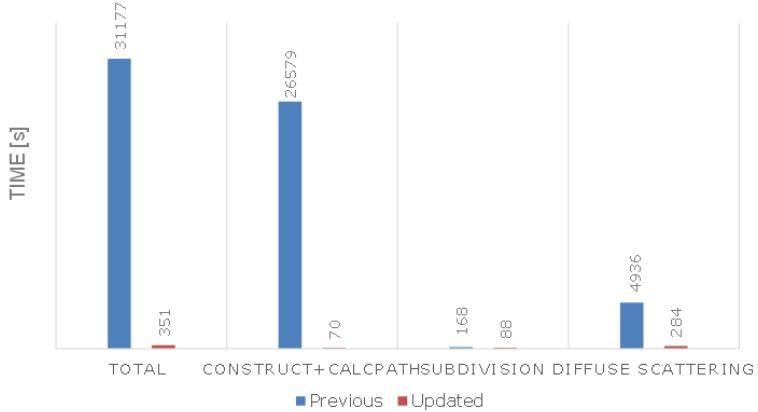


Figure 3.11: Simulation time comparation for 34-block environment.

To sum up, remarkable efficiency improvement can be achieved for most indoor scenarios with the updated 3-D RT model. However, a more efficient model is still needed for more complicated environments.

# Chapter 4

## New Algorithm to Generate Diffuse Scattering Tiles

---

**I**n this chapter, we propose a novel and efficient algorithm to generate diffuse scattering tiles, which can be applied in different scenarios. After the introduction of the “subdivision based on far-field condition” algorithm used in the original RT model, the detailed description of the proposed algorithm is given in section 4.3. Taking the simulation results of the original RT model as reference, we evaluate the performance of the new algorithm by comparing propagation parameters such as PDP, DS and AS.

### 4.1 Subdivision Based on Far-Field Condition

The subdivision algorithm involved in the diffuse scattering calculation has an important impact on the size and the number of the generated diffuse scattering tiles. The approach used in the original RT model is the “subdivision based on far-field condition” algorithm, in which the size of each scattering tile is decided by the well known far-field condition [2]

$$d_{\max} \leq \sqrt{\frac{d_s \lambda}{2}}, \quad d_{\max} = \max(d_1, d_2), \quad (4.1)$$

where  $d_{\max}$  is the larger dimension of the surface element and  $d_s$  is the distance between the center of the element and the terminal.

The principle of the subdivision algorithm is shown in Fig. 4.1. In the beginning, the far-field condition is checked for the entire surface (Fig. 4.1-(a)). If the condition is satisfied, further processing is no longer needed and the incident wave is only scattered by the center point. Otherwise, the surface is equally divided into four pieces with identical size (Fig. 4.1-(b)) and each segment is further evaluated, unqualified segments are recursively divided until they meet the far-field condition.

With this algorithm, the distribution of scattering tiles is related to the source location. One possible situation is that the segments closer to the source do not

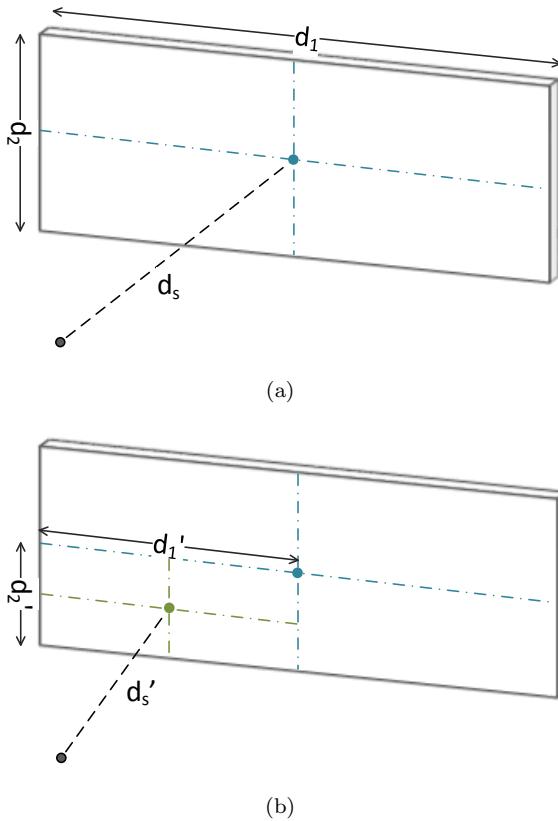


Figure 4.1: Principle of the subdivision algorithm based on far-field condition.

meet the far-field condition but the further ones do, the surface is unevenly divided into elements of different sizes [2].

An example of the possible subdivision results is demonstrated in Fig. 4.2. Because the virtual source is close to the bottom left corner of the surface, the scattering tiles in this area tend to be more intensive and much smaller after subdivision. Consequently, a substantial portion of generated scattered rays would contain very low energy due to the very small tile size. The large number of diffuse paths due to the over segmentation problem would result in very high time consumption

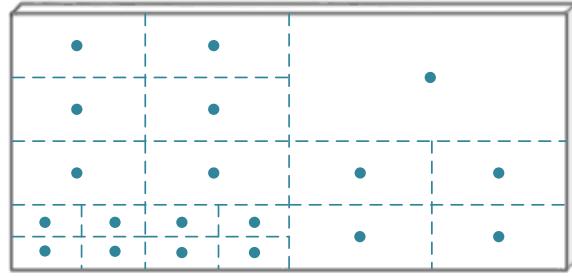


Figure 4.2: An example of the subdivision result.

## 4.2 New Algorithm to Generate Scattering Tiles

In this section, an efficient algorithm based on concentric circles for diffuse scattering tile generation in indoor scenarios is proposed. Besides implementation complexity, the random characteristic of diffuse components is also considered. The algorithm consists of two main parts: “scattering tiles generation” and “proper tile size calculation”, which are introduced in the following subsections.

### 4.2.1 Scattering Tiles Generation

The principle of the generation algorithm is shown in Fig. 4.3. Scattering tiles are assumed as equal-sized circular segments (blue dashed circles) on a certain surface and the incident waves are scattered by the center points (blue solid dots).

A randomly located point  $\mathbf{c}$  is firstly generated on the surface as the center of concentric circles (black dashed lines). The distance  $r_{\max}$  from center  $\mathbf{c}$  to the furthest vertex determines the number of concentric circles  $N$ . The radius of the concentric circles are given as:

$$r_n = \begin{cases} \Delta d & \text{if } n = 0, \\ 2n \cdot \Delta d & \text{if } n = 1, \dots, N, \end{cases} \quad (4.2)$$

$$N = \lfloor \frac{r_{\max}}{2\Delta d} \rfloor, \quad (4.3)$$

where  $\Delta d$  is the radius of the scattering tile and  $\lfloor \cdot \rfloor$  takes the nearest integer toward minus infinity.

Starting from the inner most circle, scattering tiles are generated on each circle separately. The first tile is located by a random angle  $\theta_{n,0}$  within  $[0, 2\pi]$  and the tile center is denoted as  $\mathbf{s}$  in Fig. 4.3. Then using the corresponding angle resolution

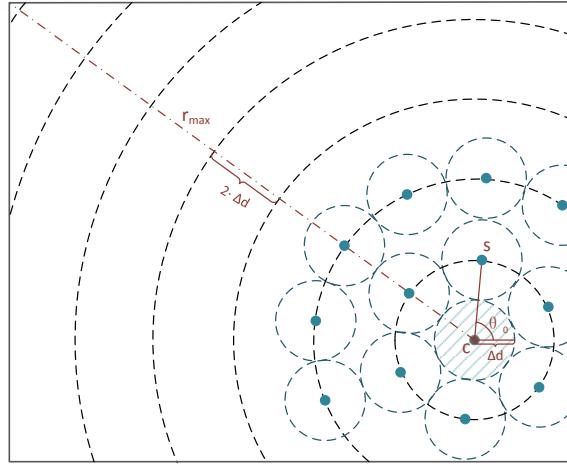


Figure 4.3: Principle of the proposed generation algorithm based on concentric circles.

$\theta_n$ , which is the angle difference between two adjacent tile centers on the same circle, tiles are generated consecutively. Tiles with centers outside of the surface are deleted. The angle resolution is expressed as:

$$\theta_n = 2 \arcsin \left( \frac{1}{2 \cdot n} \right), \quad n = 1 \cdots N, \quad (4.4)$$

The 3-D coordinates of a generated tile center  $(x, y, z)$  can be obtained. For example, the surface is horizontal, the coordinates can be expressed as:

$$s = (x, y, z) = \begin{cases} x = x_c + r_n \cdot \cos \theta \\ y = y_c + r_n \cdot \sin \theta \\ z = z_c \end{cases}, \quad (4.5)$$

where  $(x_c, y_c, z_c)$  represents the location of center  $\mathbf{c}$ ,  $\theta = \theta_{n,0} + N_{c,i} \cdot \theta_n$  where  $N_{c,i}$  is an index number of the tiles on one concentric circle, which should be no larger than the total number of the tiles on this concentric circle  $N_c$ , which can be expressed as:

$$N_c = \lfloor \frac{2\pi}{\theta_n} \rfloor, \quad (4.6)$$

The procedure described above is repeated until reaching the last circle. The random angle  $\theta_{n,0}$  should be regenerated for each circle. It should be noted that some tiles are too close to the edge and the portions outside the surface are cut

off. However, we still assume that the size of those incomplete tiles is  $dS$ . The unused area between tiles can make compensation to guarantee that the surface is not over occupied.

An example of the proposed generation algorithm is shown in Fig. 4.4. Due to the random center  $\mathbf{c}$  and  $\theta_{n,0}$ , the generated tiles can be seen as randomly located on the whole surface without overlapping, which coincides with the nature of diffuse scattering. With constant tile size, the over-segmentation problem is also avoided and the scattering points are evenly allocated.

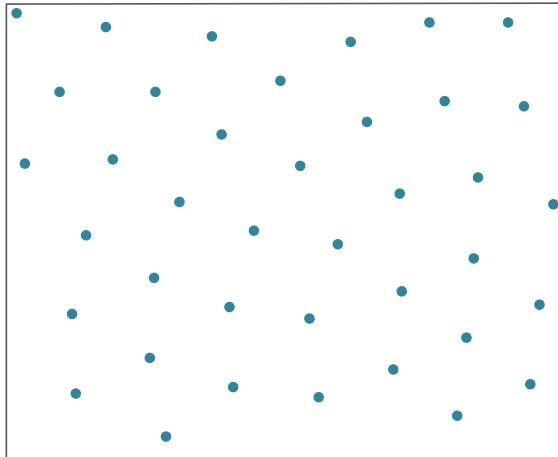


Figure 4.4: An example of the proposed generation algorithm.

This algorithm can be easily implemented and integrated into the 3-D RT model. The generated MATLAB code is fully compliant for MEX function generation, which further accelerate the computation of diffuse scattering.

### 4.2.2 Proper Tile Size

Knowing the procedure to generate scattering tiles on a given surface, the remaining problem is how to determine a proper tile size based on the parameter settings and scenario description. We provide two approaches, which are named as “proposed algorithm based on bandwidth” and “proposed algorithm based on far-field condition”, respectively.

The first approach is based on the system bandwidth  $B$ . Tile size can be expressed as:

$$dS = \pi \cdot \left[ \frac{c}{2 \cdot B} \right]^2, \quad (4.7)$$

where  $c$  represents the speed of light. The distance between center points of two adjacent scattering tiles is approximately assumed as the length difference between corresponding scattering waves. Considering the interaction from the current surface, this approach ensures that no more than one scattering component reaches the next interaction point within a same delay bin. The tile size is unchanged for all surfaces.

The other approach follows the idea of utilizing the far-field condition, but the difference is that the condition is only used once at the beginning and the extremum of the (4.1) determines the largest allowed dimension of one diffuse tile.  $dS$  is calculated as:

$$dS_n = \frac{\pi \cdot d_n \cdot \lambda}{8}, \quad n = 1 \cdots M, \quad (4.8)$$

where  $d_n$  is the distance from the virtual source to the relevant surface center,  $M$  is the total surface number in the scenario.  $dS$  varies from surface to surface.

# Chapter 5

## Simulation Results and Comparison

---

**T**he proposed generation algorithm of diffuse scattering tiles is implemented and integrated into the original 3-D RT model. Simulation configuration and results are given in this chapter. The comparison with the original RT model and a detailed analysis are presented.

### 5.1 Simulation Configuration

Simulation parameters are given in Table 5.1. The Tx remains static and the Rx moves along a direction shown in Fig. 5.1 with speed 3 km/h. The diffuse scattering paths are assumed to be incoherent. Each diffuse process is associated with a uniformly distributed random phase. Each channel simulation at one Rx location is averaged over five independent realizations to overcome the randomness. That is, totally 50 channel realizations are performed for one complete RT simulation. Scattering coefficient  $S$  are set to 0.4. Both Tx and Rx antennas are omni-directional.

Besides the tile size  $0.3068 \text{ m}^2$  which is calculated based on the bandwidth, nine other different tile sizes varies from minimum  $0.2 \text{ m}^2$  to maximum  $8 \text{ m}^2$  are also simulated for investigating the impact of the relevant active scattering region on the simulation results.

### 5.2 Numerical Results and Analysis

For evaluating the functionality of the proposed algorithms, only diffuse scattering components are considered in the simulation. Major propagation parameters such as transfer function, PDP, DS and AS are analyzed as the criteria for the performance assessments.

Parameters	Values
Carrier frequency $f_c$	2.45 GHz
Bandwidth $B$	480 MHz
Number of frequency samples $N_f$	480
Delay domain	$0 \sim 200$ ns
Delay resolution $\Delta t$	1 ns
Number of Rx positions	10
$\Delta d$	$\lambda/3$
Rx velocity	3 km/h
Tile size domain	$0.2 \text{ m}^2 \sim 8 \text{ m}^2$

Table 5.1: Parameters setting for RT model simulation.

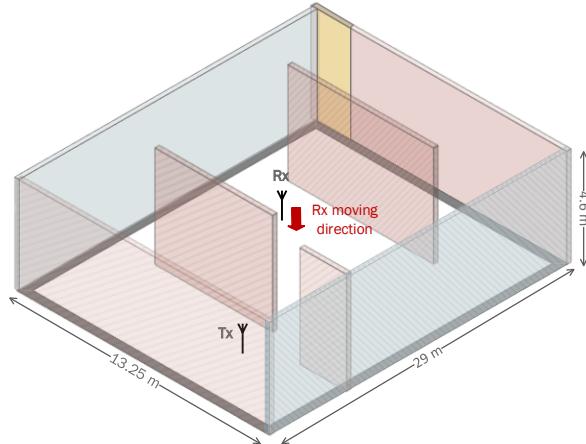


Figure 5.1: Rx moving direction.

### 5.2.1 Time-Variant Transfer Function Comparison

The channel impulse response (CIR)  $h(t, \tau)$  is evaluated first and the time-variant transfer function  $H(t, f)$  is further derived by using the Fourier transform of  $h(t, \tau)$  with respect to the variable  $\tau$  [5],

$$H(t, f) = \int_{-\infty}^{\infty} h(t, \tau) \exp(-j2\pi f\tau) d\tau, \quad (5.1)$$

As shown in Fig. 5.2, CIRs of the proposed algorithms based on bandwidth/far-field condition are comparable to the result of the original algorithm based on far-field condition. The maximum received power is around -50 dB and delays of the significant diffuse components changes with the movement of the Rx, which can be better observed in Fig. 5.3.

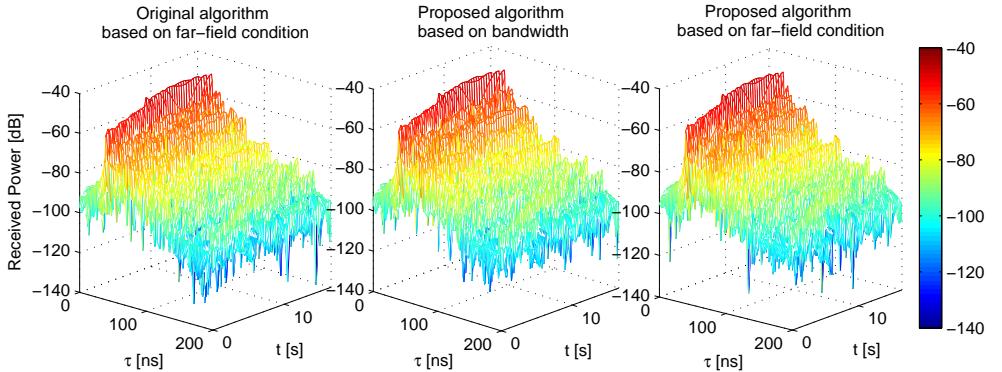


Figure 5.2: CIRs of the original subdivision algorithm and the proposed algorithms based on bandwidth/far-field condition.

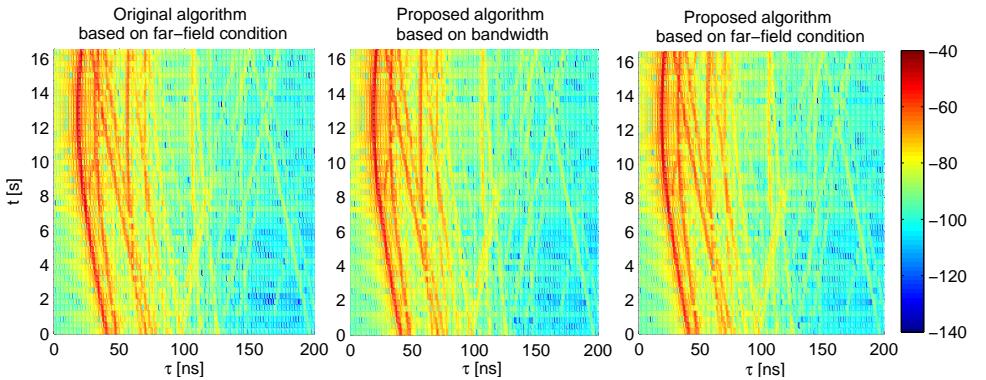


Figure 5.3: CIRs in 2-D.

### 5.2.2 PDP comparison

Based on  $h(t, \tau)$ , the PDP can be obtained as [5]:

$$P(\tau) = \int_{-\infty}^{\infty} |h(t, \tau)|^2 dt, \quad (5.2)$$

PDPs of RT models with the proposed algorithms based on bandwidth/far-field condition and the original algorithm based on far-field condition are shown in Fig. 5.4. Despite some slight difference can be observed, the results are quite similar both in delay and amplitude aspects.

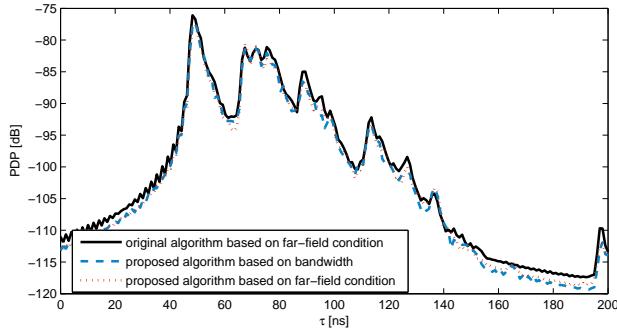


Figure 5.4: PDPs comparison between the original algorithm based on far-field condition and the proposed algorithms based on bandwidth/far-field condition.

The similarity between the PDPs of RT models with different subdivision algorithms is further investigated by calculating the correlation coefficients, which are obtained between the proposed algorithms based on bandwidth/far-field condition and the original algorithm. The result is shown in Fig. 5.5.

As can be seen, the proposed algorithms provide very similar PDPs compared with the original subdivision algorithm. Moreover, the PDP of diffuse components is stable when  $dS$  changes from  $0.2 \text{ m}^2$  to  $8 \text{ m}^2$ . The correlation coefficients between the proposed algorithms based on bandwidth/far-field condition and the original algorithm are 0.986 and 0.982 respectively. The values for other tile sizes are all above 0.96.

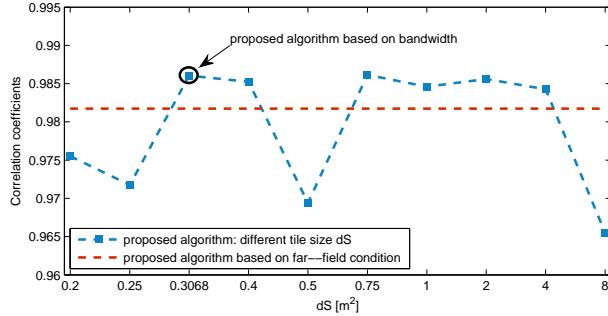


Figure 5.5: Correlation coefficients of the different PDPs.

The insensitivity to tile size variance is also reflected in the values of mean square error (MSE), which is obtained as [21]:

$$\text{err}(t_n) = \frac{\int (h_{\text{sub}}(t_n, \tau) - h_{\text{pro}}(t_n, \tau))^2 d\tau}{\int (h_{\text{sub}}(t_n, \tau))^2 d\tau}, \quad (5.3)$$

where  $h_{\text{sub}}(t_n, \tau)$  and  $h_{\text{pro}}(t_n, \tau)$  are the CIRs averaged over 10 Rx positions corresponding to the original subdivision algorithm and the proposed algorithm based on bandwidth/far-field condition, respectively.

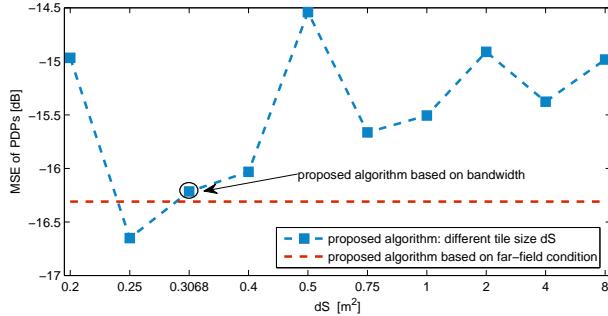


Figure 5.6: MSE of PDPs.

The MSE for the proposed algorithms with  $dS \in [0.2, 8] \text{ m}^2$  relative to the original subdivision algorithm based on far-field condition is shown in Fig. 5.6, from which it can be seen that all the MSE values fluctuate around  $-15.5 \text{ dB}$  and the maximum deviation is about  $2 \text{ dB}$ . The MSE of the proposed algorithms based on bandwidth/far-field condition relative to the original subdivision algorithm are

-16.2 dB and -16.3 dB respectively, which are slightly lower compared with the values achieved with most of the other tile sizes.

### 5.2.3 Delay Spread Comparison

Diffuse components arrive at the Rx with different delays. The delay spread is a measure of the time dispersion of the channel. The expression for calculating the delay spread  $\sigma_\tau$  based on the central moment is given as [20]:

$$\sigma_\tau = \sqrt{\frac{\sum_{i=1}^L P_i \tau_i^2}{\sum_{i=1}^L P_i} - \left( \frac{\sum_{i=1}^L P_i \tau_i}{\sum_{i=1}^L P_i} \right)^2}, \quad (5.4)$$

where  $P_i$  and  $\tau_i$  are the power and delay of the  $i$ -th diffuse scattering path and  $L$  is the number of the diffuse scattering paths. For each Rx position, the delay spread is the average of the values for five realizations. In order to show the difference between the proposed algorithms and the original subdivision algorithm intuitively, the mean relative error (MRE) is calculated as follow:

$$\text{err}_r = \frac{1}{N_{\text{Rx}}} \sum_{i=1}^{N_{\text{Rx}}} \frac{|\sigma_{\tau_{\text{sub}}}(i) - \sigma_{\tau_{\text{pro}}}(i)|}{\sigma_{\tau_{\text{sub}}}}, \quad (5.5)$$

where  $i$  is the index of the Rx position,  $\sigma_{\tau_{\text{sub}}}$  is the delay spread of the original subdivision algorithm based on far-field condition and  $\sigma_{\tau_{\text{pro}}}$  is the delay spread of the proposed algorithm based on bandwidth/far-field condition.

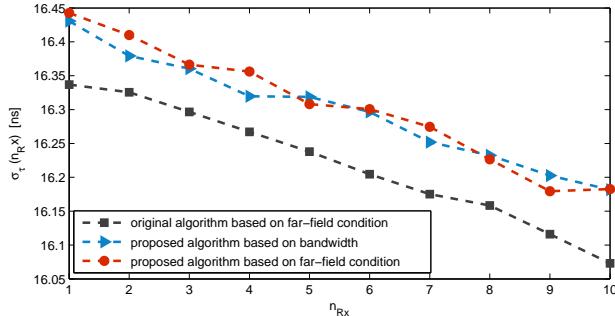


Figure 5.7: DSs of the original subdivision algorithm and the proposed algorithms based on bandwidth/far-field condition corresponding to 10 Rx positions.

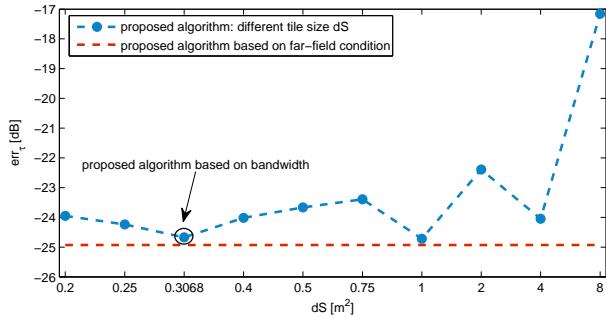


Figure 5.8: MRE of DSs regarding different tile sizes.

The delay spreads  $\sigma_\tau$  of the original algorithm and the proposed algorithms based on bandwidth/far-field condition are shown in Fig. 5.7. It can be seen that the delay spreads decrease with Rx moving and the values based on the proposed algorithms are about 0.1 ns larger compared to the values based on the original algorithm over all Rx positions. MREs regarding different tile sizes are provided in Fig. 5.8. If  $dS$  is determined with the bandwidth, the MRE is about -25 dB. A similar value is achieved with the proposed algorithm based on the far-field condition. The deviation of MRE is not obvious when the  $dS$  is smaller than  $4 \text{ m}^2$ . However, the MRE for  $dS = 8 \text{ m}^2$  sharply increases to -17 dB.

#### 5.2.4 Angle Spread Comparison

Diffuse components arrive at the Rx from different directions. The angle spread  $\sigma_\phi$  is calculated to evaluate the angular dispersion of the channel. A measure of  $\sigma_\phi$  based on the central moment is defined as [20]:

$$\sigma_\phi = \sqrt{\frac{\sum_{i=1}^L P_i \phi_i^2}{\sum_{i=1}^L P_i} - \left( \frac{\sum_{i=1}^L P_i \phi_i}{\sum_{i=1}^L P_i} \right)^2}, \quad (5.6)$$

where  $\phi_i$  is the angle of arrival of the  $i$ -th diffuse scattering path either in azimuth (AoA) or the elevation (EoA) direction. The MRE of the proposed algorithm based on bandwidth/far-field condition compared to the original subdivision algorithm is defined as:

$$err_\phi = \frac{1}{N_{Rx}} \sum_{i=1}^{N_{Rx}} \frac{|\sigma_{\phi_{sub}}(i) - \sigma_{\phi_{pro}}(i)|}{\sigma_{\phi_{sub}}}, \quad (5.7)$$

where  $\sigma_{\phi_{\text{sub}}}$  is the angle spread of original subdivision algorithm based on far-field condition and  $\sigma_{\phi_{\text{pro}}}$  is the angle spread of the proposed algorithm based on bandwidth/far-field condition.

AoA and EoA spreads corresponding to 10 Rx positions of the original algorithm and two proposed algorithms are shown in Fig. 5.9 and Fig. 5.11. The maximum AoA spread variance is about  $0.5^\circ$  and the values obtained when  $dS$  is determined with the bandwidth are relatively more similar to the spreads of the original subdivision algorithm based on far-field condition. The Fig. 5.10 illustrates the relation between MRE of AoA and different tile sizes. The MRE value reaches the minimum value when  $dS = 0.3068 \text{ m}^2$  and changes in the region (-35, -17) dB.

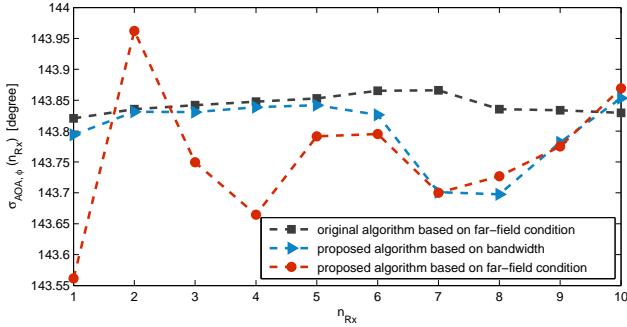


Figure 5.9: AoAs of the original subdivision algorithm and the proposed algorithms based on bandwidth/far-field condition corresponding to 10 Rx positions.

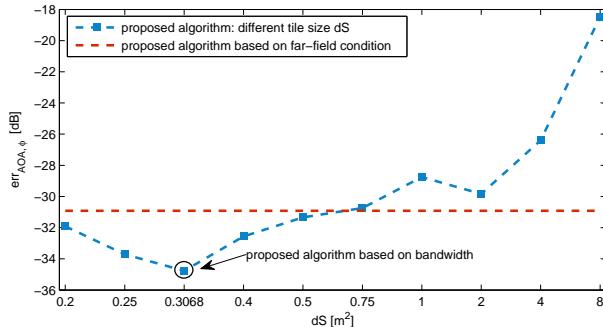


Figure 5.10: MRE of AOA spreads.

The maximum variance of EoA spreads corresponding to 10 Rx positions is about  $0.3^\circ$ , which is smaller than the AoA spreads. Moreover, the MRE value of EoA spreads rise with the increasing of  $dS$  from the minimum -30 dB to the maximum -16 dB. The MRE obtained when  $dS$  is determined with the bandwidth is about -27.5 dB, which is smaller than the value obtained with the proposed algorithm based on the far-field condition.

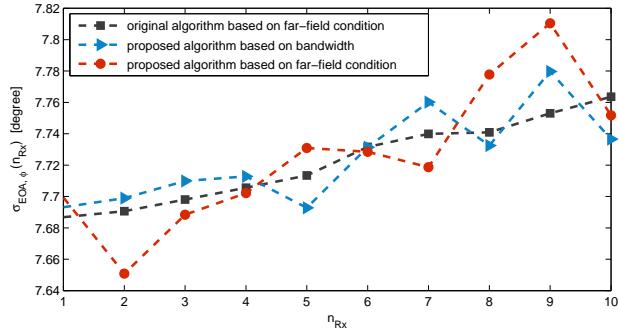


Figure 5.11: EoAs of the original subdivision algorithm and the proposed algorithms based on bandwidth/far-field condition corresponding to 10 Rx positions.

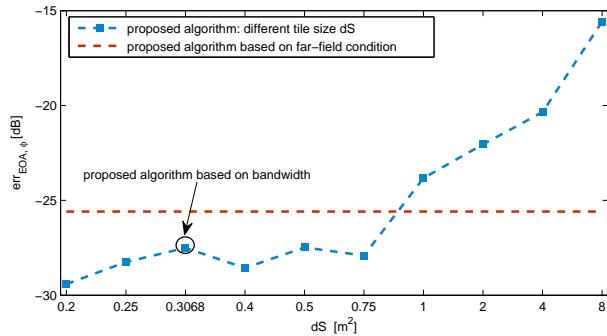


Figure 5.12: MRE of EoA spreads.

To conclude, compared with the PDP, the diffuse scattering tile generation algorithm has more obvious impact on the prediction capability of the time- and angle- dispersion characteristics. The proposed algorithm based on bandwidth can provide relatively more accurate angle information both in the azimuth and elevation directions compared with the proposed algorithm based on the far-field condition.

### 5.2.5 Computational Complexity Comparison

With the proposed algorithms, the computational complexity of RT is significantly reduced. Seen from the Fig. 5.13, the number of generated diffuse paths for each RT realization decreased by about 45%, from 37889 obtained according to the original subdivision algorithm to 21026 which is acquired with the proposed algorithm when  $dS$  is calculated based on bandwidth. Less path number can be further achieved with the proposed algorithm based on far-field condition.

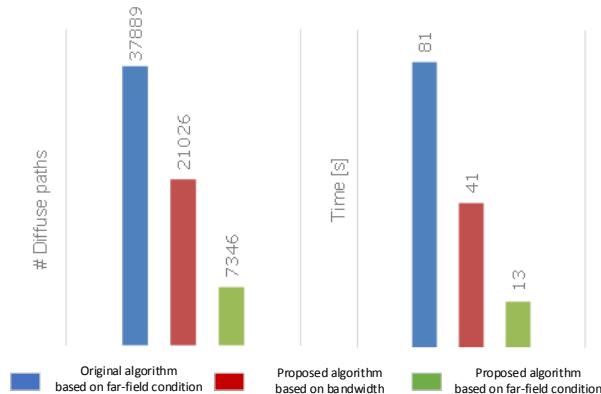


Figure 5.13: Time consumption and number of diffuse scattering paths of the original subdivision algorithm and the proposed algorithms based on bandwidth/far-field condition.

The time consumption for each RT realization decreased from the original 81s to 41s and 13s, (2.4GHz Intel Core i5 CPU with 4GB RAM) which are corresponding to the proposed algorithms based on bandwidth/far-field condition, respectively.

# Chapter 6

## Conclusions

---

This thesis focused on the ray tracing techniques for indoor propagation scenarios. The entire work is composed of two parts: (1) speeding up the original 3-D RT model and (2) a new algorithm for diffuse scattering tile generation.

The speeding up effort in this thesis included the reflection and diffuse scattering calculation parts. We optimized the original MATLAB code and replaced time consuming functions with MEX functions which were generated with MATLAB Coder. By introducing some new features such as “last layer check” and “surfaces on the same plane”, the computational load during the creation of image tree was significantly reduced. In order to solve the “out of memory” problem of RT when applied in complex and large dimensional scenarios, we proposed an efficient image tree construction algorithm “Effconstruct”, in which the image tree has a constant size and is continually updated. Compared with the original execution time, the execution time of the revised RT is significantly reduced. For example, about ninety percent time consumption is saved from the original RT for an 8-block scenario.

We analyzed the existing diffuse scattering models and realized that the method used to generate scattering tiles on building surfaces directly influenced the number and energy of scattering components involved in the final calculation. For further reducing the execution time of the original RT, a general and efficient subdivision algorithm for diffuse scattering was proposed. Scattering tiles were generated on each surface fully considering the random characteristic of diffuse components. Besides, the proper tile size could be determined based on the system bandwidth, which makes this approach site independent and widely applicable in different scenarios. Another approach to calculate the tile size based on the far-field condition was also provided and proved to be efficient as well. Parameters such as the PDP, DS and AS (Azimuth and Elevation) of diffuse scattering components were analyzed as the metrics for the performance assessments. Compared with the original subdivision algorithm based on the far-field condition, the proposed algorithms could achieve much lower computational complexity and less simulation time without causing a degradation of prediction accuracy.



# Chapter 7

## Future Work

---

The simulation time of the original RT has been greatly reduced with the presented work in this thesis. However, the time consumption of RT in large scale environments with high interaction order is still problematic. Further speed up effort is still needed.

We noticed that the energy contained in each propagation path is not only determined by its experienced interaction number. Tracking propagation paths simply based on default interaction order may cause the problem of excluding energy portion from higher order propagation paths. Hence, how to eliminate propagation paths based on estimated strength during the compilation of the image tree is worth further investigation.

Involving higher order reverberation paths could improve the prediction accuracy of RT. Relevant approaches with low computation complexity, e.g., graph-based stochastic model [22], shall be studied and their feasibility should be verified.



# Bibliography

- [1] M. Gan, F. Mani, F. Kaltenberger, C. Oestges, and T. Zemen, “A ray tracing algorithm using the discrete prolate spheroidal subspace,” in *IEEE International Conference on Communications (ICC)*, June 2013.
- [2] F. Mani, F. Quitin, and C. Oestges, “Accuracy of depolarization and delay spread predictions using advanced ray-based modeling in indoor scenarios,” *EURASIP Jounal in Wireless Communications and Networking*, vol. 2011, p. 11, 2011.
- [3] P. Almers, E. Bonek, A. Burr, N. Czink, M. Debbah, V. Degli-Esposti, H. Hofstetter, P. Kyoesti, D. Laurenson, G. Matz, A. Molisch, C. Oestges and H. Oezcelik “Survey of channel and radio propagation models for wireless MIMO systems”, *EURASIP J. Wireless Commun. Networking*, 2007.
- [4] J. Karedal, F. Tufvesson, N. Czink, A. Paier, C. Dumard, T. Zemen, C. F. Mecklenbräuker, and A. F. Molisch, “A geometry-based stochastic MIMO model for vehicle-to-vehicle communications”, *IEEE Trans. Wireless Commun.*, vol. 8, no. 7, pp. 3646-3657, Jul. 2009.
- [5] Andreas F. Molisch, *Wireless Communications* 1st edition, Wiley Press, 2006.
- [6] O. Renaudin, V. M. Kolmonen, P. Vainikainen, and C. Oestges, “Wideband measurement-based modeling of inter-vehicle channels in the 5-GHz band”, *IEEE Transactions on Vehicular Technology*, vol. 62, no. 8, Oct. 2013.
- [7] K. Rizk, J. F. Wagen and F. Gardiol, “Ray tracing based path loss prediction in two microcellular environments,” in *Proc. IEEE PIMRC*, Hague, The Netherlands, Sep. 1994, pp. 384–388.
- [8] C. Oestges, B. Clerckx, L. Raynaud, and D. Vanhoenacker-Janvier, “Deterministic channel modeling and performance simulation of microcellular wideband communication systems,” *IEEE Trans. Veh. Technol.*, vol. 51, no. 6, pp. 1422-1430, Nov. 2002.
- [9] V. Degli-Esposti, D. Guiducci, A. deMarsi, P. Azzi, and F. Fuschini, “An advanced field prediction model including diffuse scattering,” *IEEE Trans. Antennas Propag.*, vol. 52, no. 7, pp. 1717-1728, Jul. 2004.
- [10] V. Degli-Esposti, “A diffuse scattering model for urban propagation prediction,” *IEEE Trans. Antennas Propag.*, vol. 49, no. 7, pp. 1111-1113, Jul. 2001.
- [11] V. Degli-Esposti, F. Fuschini, E. Vitucci, and G. Falciasecca, “Measurement and modelling of scattering from buildings,” *IEEE Trans. Antennas Propag.*, vol. 52, no. 1, pp. 143-153, Jan. 2007.

- [12] J. Maurer, O. Drumm, D. L. Didascalou, and W. Wiesbeck, “Ray-density normalisation in the determination of visible surfaces in 3D vector geometries for ray-optical wave propagation modelling,” in *Proc. 51st IEEE Veh. Technol. Conf.*, Tokyo, Japan, pp. 1651–1655, May 2000.
- [13] M. Gan, P. Meissner, F. Mani, E. Leitinger, M. Froehle, C. Oestges, K. Witrisal, and T. Zemen, “Low-complexity sub-band divided ray tracing for UWB indoor channel,” in *IEEE Wireless Communications and Networking Conference (WCNC)*, April 2014, to be presented.
- [14] C. Oestges, “Propagation modelling of low earth-orbit satellite personal communication systems”, Ph.D dissertation, Universit catholique de Louvain, Dec. 2000.
- [15] T. Kurner, D. J. Cichon, and W. Wiesbeck, “Concepts and results for 3D digital terrain-based wave propagation models: An overview,” *IEEE J. Select. Areas Commun.*, vol. 11, pp. 1002-1012, 1993.
- [16] O. Franek, J. B. Andersen, and G. F. Pedersen, “Diffuse scattering model of indoor wideband propagation,” *IEEE Trans. Antennas Propag.*, vol. 59, no. 8, pp. 3006-3012, Aug. 2011.
- [17] C. Jansen, S. Priebe, C. Moller, M. Jacob, H. Dierke, M. Koch, and T. Kurner, “Diffuse scattering from rough surfaces in THz communication channels,” *Terahertz Science and Technology, IEEE Transactions on*, vol. 1, no. 2, pp. 462-472, Nov. 2011.
- [18] H. R. Anderson, “A second generation 3-D ray-tracing model using rough surface scattering,” *Vehicular Technology Conference*, 1996, IEEE 46th, vol. 1, pp. 46–50, Apr. 1996.
- [19] F. Mani, F. Quitin, and C. Oestges, “Directional spreads of dense multipath components in indoor environments: Experimental validation of a ray-tracing approach,” *Antennas and Propagation, IEEE Transactions on*, vol. 60, no. 7, pp. 3389-3396, Jul. 2012.
- [20] P. Petrus, J. H. Reed, T. S. Rappaport, “Geometrical-based statistical macro-cell channel model for mobile environments,” *Communications, IEEE Transactions on*, Vol. 50, No. 3, pp. 495–502, Mar. 2002.
- [21] G. Tiberi, S. Bertini, W. Malik, A. Monorchio, D. Edwards, and G. Manara, “Analysis of realistic ultrawideband indoor communication channels by using an efficient ray-tracing based method,” *IEEE Trans. Antennas Propag.*, vol. 57, no. 3, pp. 777-785, Mar. 2009.
- [22] Pedersen. T, Steinbeck. G, Fleury. B. H “Modeling of Reverberant Radio Channels Using Propagation Graphs”, *IEEE Trans. Antennas Propag.*, vol. 60, no. 12, pp. 5978–5988, 2012.

## Bibliography

---

- [23] MATLAB version 8.0.0.783 Natick, Massachusetts: The MathWorks Inc., 2012.



LUND  
UNIVERSITY