

レイトレーシング法に特化したGPUによる電波伝搬シミュレーションの高速化

発表者： I専攻 成見研究室 学籍番号:2031008 荒生 太一
指導教員： 成見 哲
副指導教員： 沼尾 雅之

1. はじめに

近年, ITS(高度道路交通システム)での車車間・路車間無線通信技術や5G技術の発展等, 無線通信技術がますます発展している. 実世界での測定には多くのコストが必要であるため, 電波の伝わり方(電波伝搬)をコンピュータで計算する電波伝搬シミュレーションが盛んに行われてきた. 無線通信の大容量・高周波化および解析領域の大規模化に伴い解析にはレイトレーシング法が良く用いられている.

2018年に発売されたNVIDIA社製のGPU(Graphics Processing Unit)であるRTXシリーズではレイトレーシング法の計算を高速化するために従来の汎用プロセッサに加えてRTコアと呼ばれる専用プロセッサを搭載している. レイトレーシング法に近い計算問題であれば同様に高速化することが可能である[1]. 本研究ではNVIDIA RTXを用いて電波伝搬シミュレーションの高速化を行う.

2. 既存技術・既存研究

2.1 レイトレーシング法

レイトレーシング法は送信点から発射された電波が解析領域内で反射や回折を繰り返し受信点に到達するまでの経路を探索する手法であり, 大きく分けてイメージング法とSBR(Shooting and Bouncing Rays)法の2つが存在する(図1). イメージング法は反射面の組み合わせから鏡像点を求めることでレイの軌跡を厳密に求める手法で高精度であるが反射面や反射回数が増えるにつれて計算量が指数的に増加してしまう.

SBR法は送信点からレイを一様に発射しその軌跡を逐次追跡する手法である. 受信点に厳密にレイが届くことはほぼないため, 受信点の周りに一定サイズの受信領域を設置する. イメージング法よりも計算量が少ないが, 追跡するレイの本数や受信領域のサイズが適切でない場合精度が低下する. 特に同一経路のレイを2回カウントしてしまう場合があり, 精度向上のためにはこれを除去する必要がある.

本研究では両者を組み合わせたSBR-image法[2]を用いた. SBR法でレイが通る反射面を計算した後, その情報をもとにイメージング法で補正を行うことでSBR法と同等の計算量で精度の高い計算を行うことができるが, GPUでの実装が複雑になる.

2.2 NVIDIA OptiX [3]

レイトレーシング法で最も計算時間がかかるのはレイと交差するオブジェクトを探索する処理であり, RTXのRTコアはこの交差判定処理に特化している. RTコアを使用できるAPIはいくつか存在しているが, 本研究ではNVIDIA OptiXを用いた. これは, OptiX以外のAPIはゲームのようなリアルタイム処理に重きを置いているためである. OptiXはレイの振る舞いをユーザーが自由に記述できる他, オブジェクトを木構造に格納することで交差判定を行う回数を大幅に削減することができる.

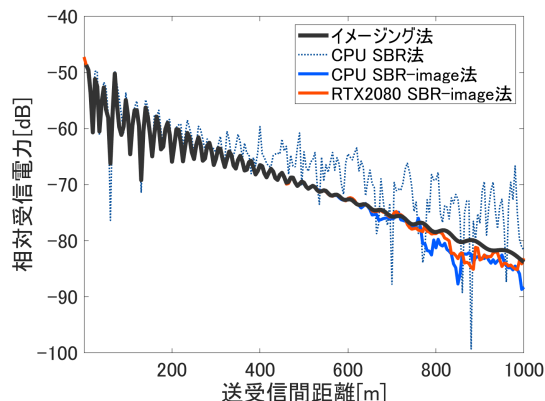


図2: 各手法の精度比較

2.3 既存研究

電波伝搬シミュレーションにOptiXを用いた研究としてミリ波帯を用いた屋内環境[4]や大規模トンネル内を対象としたもの[5], 汎用ゲームエンジンで電界計算を行うためのもの[6]が存在する. これらの研究はシミュレーション精度の議論が中心であり, 高速化に関しての議論があまり行われていないという点で本研究とは異なる.

3. システム概要

OptiXを用いてSBR-image法を計算するプログラムを作成した. 大まかな処理の流れを以下に示す.

1. シミュレーションに必要なパラメータ(レイの最大反射回数, 解析領域情報, 周波数, 送信機と受信機の位置等)を入力.
2. それらの情報をGPUに転送し, OptiXでSBR法の計算を行う. このときレイがどの反射面と衝突したかの経路情報を配列に格納しておく.
3. 得られた経路情報からGPUでイメージング法の処理を行う.
4. 同一経路を辿ったレイを除去し, 総電界を計算してCPUに転送し結果を表示する.

二重カウント除去の手法としてイメージング法で後処理を行う際に計算される受信機に到達する直前の鏡像点(最終鏡像点)を用いた. この座標が同じ場合, 二重カウントとして扱う. こうすることで反射回数が増えても比較する要素数を一定に保つことができる.

4. 評価

計算速度の比較を行う前に, 実装したプログラムが正しく計算を行っているか確認した. 結果を図2に示す. 解析領域として空洞の直線トンネルを用いた. SBR-image法はSBR法に比べて高精度であり, かつGPU版とCPU版の誤差も少ないことが確認できた.

次に実装したGPU版のSBR-image法の計算時間をCPU版のSBR-image法の計算時間と比較した. 使用したGPUの詳細を表1, 結果を図3に示す. 解析領域として $50 \times 4 \times 50$ [m]の部屋を用いた. レイの本数を2,621,442本で固定とし, 反射回数を0から10まで増やしながら計測を行った. イメージング法は反射回数が増えるにつれて計算時間が指数的に増加しており, そのままでは反射回数が多い場合の計算を行うのは現実的ではない. また, 各GPU版はCPU版SBR-image法よりも高速に計算できており, 最も性能が高いRTX A6000の場合はCPUに比べ約30倍高速であった.

測定に用いた部屋はポリゴン数が12でありレイトレーシング部分の計算負荷が非常に小さい. 従って, より大規模な領域でもシミュレーションを行った. TxおよびRxの位置を追加し, 各エリアでの受信レベルを色分けして表示した解析領域を図4に示す. また, 結果を図5に示す. 図5より, CPUの計算時間は図3に比べて大きく増加しているが, GPUの計算時間はあまり増加していない. これはCPU版

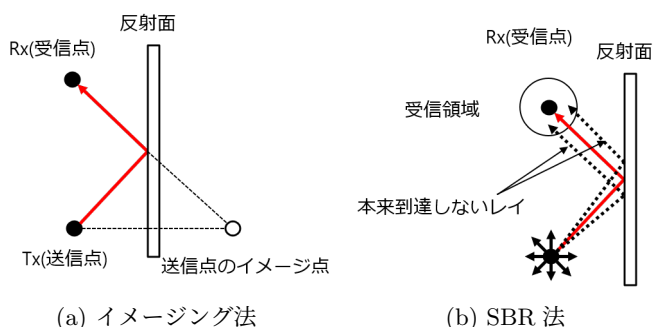


図1: 各手法の概要図

表 1: 使用した GPU の詳細

GPU 名	RTX A6000	RTX 3070	RTX2080	GTX1660Ti
CUDA コア数	10,752	5,888	2,944	1,536
RT コア数	84	46	46	0
ベースクロック [MHz]	1,800	1,730	1,710	1,770

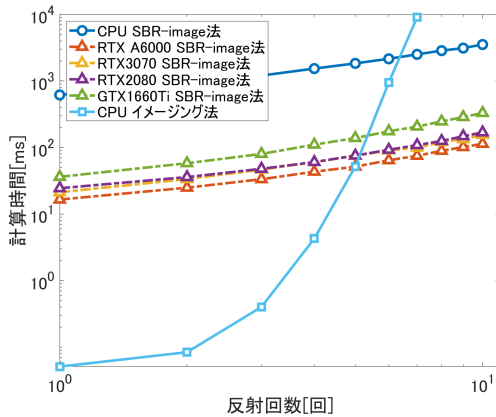


図 3: CPU と GPU の計算時間の比較

が毎回全てのポリゴンに対して交差判定を行うのに対し、GPU 版はポリゴンを木構造に格納し交差判定回数を削減しているためである。特に今回用いた解析領域は屋外であるためレイがどのポリゴンとも交差しにくい可能性が高く、例えば真上方向に出射されたレイも CPU では全てのポリゴンとの交差判定を行うため、多くの計算時間がかかっていると考えられる。比較の結果、最も性能が高い RTX A6000 の場合は CPU に比べ約 5000 倍高速であった。

各 GPU によってどの程度計算時間に差があるかを確認するために、図 5 において GTX1660Ti による計算速度を 1 としたときの他 GPU による計算速度の比をプロットしたものを図 6 に示す。表 1 より、RTX2080 は GTX1660Ti よりも CUDA コアの数約 1.9 倍多いが、RTX2080 は GTX1660Ti よりも 2 倍以上高速に計算を行っている。これは GTX1660Ti は RT コアを持たないが RTX2080 は RT コアを持つため、より高速に計算を行うことができていると考える。

一方で、RTX A6000 は GTX1660Ti よりも CUDA コア数が 7 倍、RTX3070 は約 3.83 倍多いが、計算時間は RTX A6000 で約 4 倍、RTX3070 で約 3 倍程度となっている。比較に用いた屋外領域は、レイがどのオブジェクトとも交差しにくい場合が多いため、トレースするレイの数が少なくなっている。従って、これらの GPU が持つ計算リソースを使い切れていないことが考えられる。屋外ではなく部屋のようなレイとオブジェクトとの交差が起きやすく、かつポリゴン数が多い領域を用意する、もしくはレイの出射本数や反射回数をより多くして測定を行うことが必要であると考える。

5. おわりに

本研究では高速な電波伝搬シミュレーションを行うために、レイトレーシング法に特化した GPU である RTX で計算を行うプログラムを作成した。作成したシステムを評価した結果、システムは CPU だ

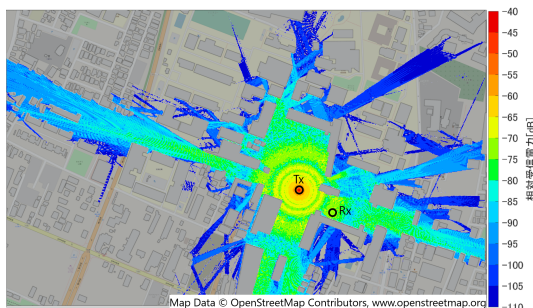


図 4: 解析領域 ([7] より得られた地形データを使用)

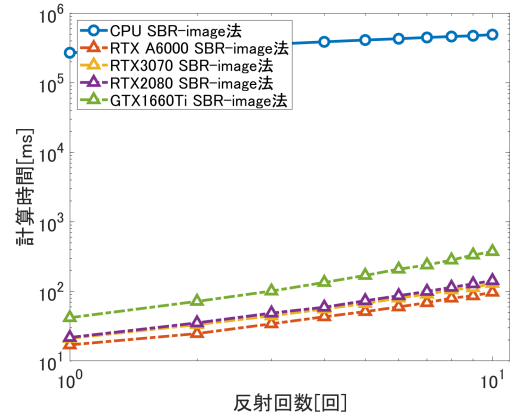


図 5: CPU と GPU の計算時間の比較 (大規模領域)

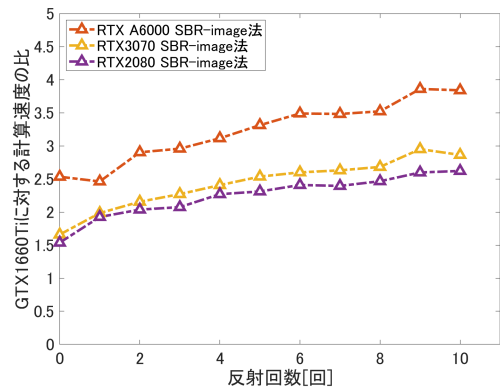


図 6: GTX1660Ti に対する計算速度の比

けで処理を行った場合と同等の精度で計算を行うことが可能であり、かつ低負荷時でもおよそ 30 倍、高負荷時ではおよそ 5000 倍高速に計算することができていた。また、RTX ではない従来 GPU との比較により、RTX が搭載している RT コアによってレイトレーシング法の計算が高速に行われていることが確かめられた。

今後の課題としてさらなる高速化や省メモリ化、回折等の物理現象の実装が挙げられる。

参考文献

- [1] Ingo Wald, Will Usher, Nathan Morrical, Laura Lediaev, and Valerio Pascucci. RTX Beyond Ray Tracing: Exploring the Use of Hardware Ray Tracing Cores for Tet-Mesh Point Location. In Markus Steinberger and Tim Foley, editors, *High-Performance Graphics - Short Papers*. The Eurographics Association, 2019.
- [2] Shin-Hon Chen and Shyh-Kang Jeng. SBR image approach for radio wave propagation in tunnels with and without traffic. *IEEE Transactions on Vehicular Technology*, Vol. 45, No. 3, pp. 570–578, 1996.
- [3] Steven G. Parker, James Bigler, Andreas Dietrich, Heiko Friedrich, Jared Hoberock, David Luebke, David McAllister, Morgan McGuire, Keith Morley, Austin Robison, and Martin Stich. OptiX: A General Purpose Ray Tracing Engine. *ACM Transactions on Graphics*, Vol. 29, No. 4, pp. 1–13, jul 2010.
- [4] Robert Felbecker, Leszek Raschkowski, Wilhelm Keusgen, and Michael Peter. Electromagnetic wave propagation in the millimeter wave band using the NVIDIA OptiX GPU ray tracing engine. In *2012 6th European Conference on Antennas and Propagation (EuCAP)*, pp. 488–492, 2012.
- [5] Blake A Troksa. GPU accelerated cone based shooting bouncing ray tracing. Master's thesis, Colorado State University, Electrical and Computer Engineering, 2019.
- [6] Esteban Egea-Lopez, Jose Maria Molina-Garcia-Pardo, Martine Lienard, and Pierre Degauque. Opal: An open source ray-tracing propagation simulator for electromagnetic characterization. *PLOS ONE*, Vol. 16, No. 11, pp. 1–19, 11 2021.
- [7] OpenStreetMap contributors. OpenStreetMap. <https://www.openstreetmap.org> (アクセス日 2022-01-21).