

Um Tutorial para Geração de Traçado de Raios Utilizando Softwares OpenSource

Alex Ferreira, e Diego Gomes

Resumo— Este trabalho apresenta um tutorial de como utilizar múltiplos sistemas de software para o estudo de propagação de ondas no contexto de 5G. O processo envolveu a exportação de cenas do Blender utilizando dados do OpenStreetMap, com orquestração de Código em Python rodando da plataforma Google Colab. Para a geração do traçado de raios foi utilizada a biblioteca Sionna RT para Python. Como resultado foi calculada a resposta ao impulso do canal (CIR).

Palavras-Chave— Traçado de raios, Sionna RT, 5G.

Abstract— This work presents a tutorial on how to use multiple software systems to study wave propagation in the context of 5G. The process involved exporting Blender scenes using OpenStreetMap data, with code orchestration in Python running on the Google Colab platform. To generate the ray tracing, the Sionna RT library for Python was used. As a result, the channel impulse response (CIR) was calculated.

Keywords— Ray tracing, Sionna RT, 5G.

I. INTRODUÇÃO

O uso de traçado de raios para a simulação de propagação de ondas de rádio é fundamental para diversas aplicações avançadas, incluindo algoritmos transceptores baseados em aprendizado de máquina (ML) [2]. Essa abordagem permite, por exemplo, a criação de gêmeos digitais onde uma cena física é replicada virtualmente para simular e testar diferentes aspectos do comportamento do sinal [1].

A correspondência espacialmente consistente entre uma localização física e a resposta ao impulso do canal (CIR) é crucial para garantir que as simulações sejam precisas e reflitam a realidade. Isso é realizado utilizando softwares específicos de traçado de raios que ajudam a prever e otimizar a qualidade dos canais de comunicação dos sistemas 5G, por exemplo. Com esses modelos virtuais é possível avaliar interferências e perdas de sinal que podem não ser evidentes em análises teóricas simples, identificar potenciais problemas antes da implementação física, permitindo correções antecipadas, e ajustar parâmetros para maximizar a eficiência e o desempenho dos sistemas de comunicação [1].

O uso de gêmeos digitais é uma prática que está se tornando cada vez mais comum, não apenas em 5G, mas em várias áreas de engenharia e tecnologia [2]. Ele oferece uma plataforma robusta para inovação, permitindo que os desenvolvedores experimentem e otimizem suas soluções em um ambiente controlado antes de implementá-las no mundo real.

Alex Ferreira, Instituto de Geociência e Engenharia, UNIFESSPA, Canaã dos Carajás-PA, e-mail: alex.ferreira@unifesspa.edu.br; Diego Gomes, Instituto de Geociência e Engenharia, UNIFESSPA, Marabá-PA, e-mail: diagogomes@unifesspa.edu.br

Neste trabalho, apresentamos como utilizar o OpenStreetMap, o Blender e o Sionna para Python para gerar simulações de traçado de raios para avaliar a propagação de sinais em frequências do 5G em ambientes urbanos, como uma etapa para futuras simulações de gêmeos digitais no formato de tutorial.

II. MATÉRIAS E MÉTODOS

Para exportar uma cena do Blender para o formato Mitsuba, primeiramente, utilize o addon Blender-OSM para importar dados do OpenStreetMap [8]. Para instalar o addon Blender-OSM, vá para "Edit" > "Preferences", clique em "Add-ons" e depois em "Install". Selecione o arquivo .zip do Blender-OSM que você baixou e clique em "Install Add-on". Em seguida, ative o addon na lista de add-ons. Com o addon ativado, vá para a barra lateral (pressione N) e clique na aba OSM. Clique em "Select" e selecione a área que deseja importar. Após o download, copie os dados em "Paste" e clique em "Import" para trazer os dados para o Blender.

Configure os materiais das cenas utilizando a tabela de materiais de rádio fornecidos pelo Sionna [9]. No Blender, selecione os objetos aos quais deseja aplicar os materiais. Configure os materiais de acordo com as propriedades fornecidas na tabela, ajustando parâmetros como reflectância e rugosidade.

Por fim, exporte a cena para o formato Mitsuba utilizando o addon Mitsuba 3 [4]. Instale o addon Mitsuba 3 baixando o arquivo e, no Blender, vá para "Edit" > "Preferences" > "Add-ons" > "Install". Selecione o arquivo .zip do Mitsuba 3 e clique em "Install Add-on". Ative o addon na lista de add-ons. Com a cena pronta, vá para "File" > "Export" > "Mitsuba Scene (.xml)", configure as opções de exportação conforme necessário, escolha o local para salvar o arquivo .xml e clique em "Export Mitsuba Scene".

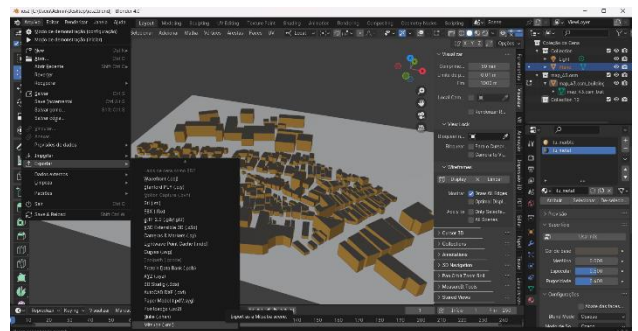


Fig. 1. Exporta Cena em formato .xml para Mitsuba.

O Sionna é uma biblioteca de simulação desenvolvida pela NVIDIA para redes de comunicação com foco em tecnologias sem fio, como 5G, sendo altamente modular e

flexível para permitir simulações detalhadas e precisas [1]. Para configurar o Google Colab para usar Mitsuba e Sionna RT, comece abrindo o Google Colab e criando um novo notebook. No primeiro bloco de código, insira os comandos para instalar o Mitsuba e Sionna RT:

```
!pip install mitsuba
!pip install sionna-rt
```

Para carregar no Colab uma cena gerada pelo Blender, comece fazendo o upload do arquivo (.xml) exportado do Blender para o Google Drive. Para configurar o Mitsuba e o Sionna RT no Google Colab, informe o caminho do arquivo (.xml) no Google Drive e configure o ambiente Mitsuba. Em seguida, inicialize a biblioteca Sionna RT e carregue a cena Mitsuba (.xml) com o seguinte código:

```
import mitsuba
import sionna.rt as rt
mitsuba.set_variant('scalar_rgb')
scene = rt.Scene()
scene.load(mitsuba_xml_path)
```

O código em Python utilizado neste trabalho pode ser baixado em [10].

III. RESULTADOS

A simulação foi executada com uma frequência de portadora utilizada em sistemas de comunicação 5G de 3,6 GHz, com um transmissor e um receptor. A Fig. 2 mostra o cenário renderizado, o qual foi tomado da cidade de Ilhéus - BA, e os raios que foram calculados, onde podem ser observadas as localizações do transmissor e do receptor. Nesta situação, o algoritmo de traçado de raios calculou 3 raios: um de caminho direto e outros dois que chegaram por multipercurso.

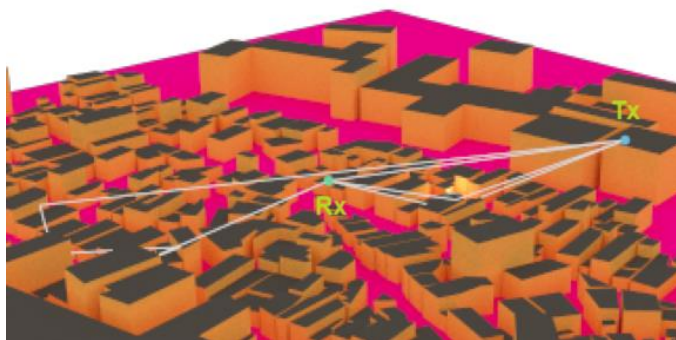


Fig. 2. Renderização do cenário 3D junto com os raios que foram calculados.

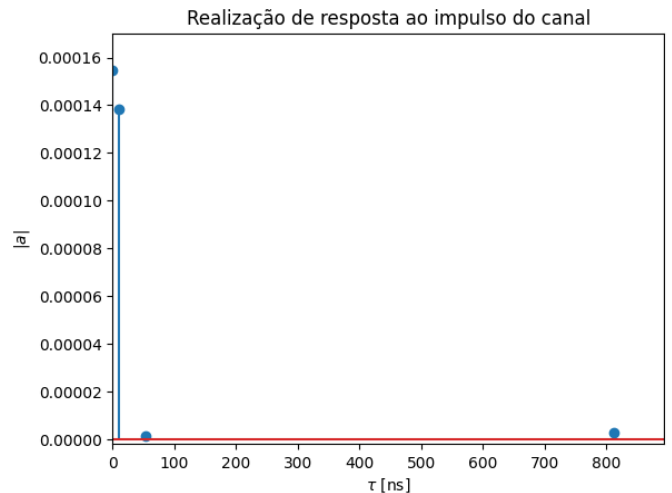


Fig. 3. Resultado da resposta ao impulso do canal.

Adicionalmente, foi calculada a Resposta ao Impulso do Canal (CIR), que descreve como um sinal é afetado ao passar pelo canal de comunicação. A CIR é uma função do tempo que mostra o atraso e a amplitude dos múltiplos caminhos que o sinal percorre. Na Fig. 3, podemos observar o primeiro impulso com maior intensidade, atribuído ao caminho de visada direta, e mais outros dois impulsos (associados aos outros dois raios) que chegam depois devido ao maior caminho percorrido.

CONCLUSÕES

Este trabalho mostrou como utilizar os softwares OpenStreetMap, Blender, Google Colaboratory e o pacote Sionna para gerar simulações de traçado de raios. Foram fornecidos detalhes sobre pacotes adicionais que devem ser instalados no Blender, bem como a importação de bibliotecas para o Python. Como resultado, foi apresentada a resposta ao impulso do canal. Além disso, foi compartilhado o link para download do código desenvolvido.

REFERÊNCIAS

- [1] Jakob Hoydis, Fayc, al A'it Aoudia, et al., "Sionna RT: Differentiable Ray Tracing for Radio Propagation Modeling". 19 junho 2023.
- [2] Airton Antunes, "A importância dos softwares de simulação dentro da indústria 4.0: Uma análise da inserção do Digital Twin nos contextos industriais". Universidade Federal de Ouro Preto – UFOP, Março/2023
- [3] "Sionna RT: Scene Creation with Blender using OpenStreetMap", 2023, <https://www.youtube.com/watch?v=7xHLDxUaQ7c>.
- [4] "Mitsuba 3 Renderer," 2022, <https://mitsuba-renderer.org>.
- [5] OpenStreetMap Foundation, "OpenStreetMap," 2023, <https://www.openstreetmap.org>.
- [6] Blender Foundation, "Blender". 2023, <https://www.blender.org>.
- [7] Mitsuba, "Mitsuba Blender Add-on,". 2023, <https://github.com/mitsubarenderer/mitsuba-blender>.
- [8] Prochitecture, "Blender-OSM: OpenStreetMap and Terrain for Blender,". 2023, <https://prochitecture.gumroad.com/1/blender-osm>
- [9] "Ray Tracing", Copyright 2021-2024 NVIDIA CORPORATION. <https://nvlabs.github.io/sionna/api/rt.html#>.
- [10] Sionna RayTracer - https://github.com/alexferreira1002/Sionna-RT/blob/cc9773b9ba932ac006ce1fbefdf720aaca655a18/Syonna_RT.ipynb.

