Fiber Config Generator
An attempt at creating a 3D white matter fiber crossing simulator using preexisting tools

Benoît Dubreuil
Computer Science department
Université du Québec à Montréal
Montréal, Canada H3C 3P8
dubreuil.benoit.2@courrier.uqam.ca

Joël Lefebvre
Computer Science department
Université du Québec à Montréal
Montréal, Canada H3C 3P8
lefebvre.joel@uqam.ca

September 17, 2022

ABSTRACT

Computer-generated phantoms are extensively used to validate diffusion-weighted imaging algorithms. There is, however, an absence of fiber-centric phantom generation tools independent of brain-imaging techniques. The goal of the project Fiber Config Generator is to create a 3D white matter fiber crossing simulator to generate synthetic data in order to validate our new algorithms for processing of light sheet microscopy data. For this purpose, we explored the possibility of developing an application that utilizes an MRI-centric library (Simulation Generator) in order to generate high-level fiber configurations.

Through the development of this application, our requirements evolved to include the parameterization of the axon distribution and biological neural network, and we discovered no MRI-centric libraries could not satisfy them. Furthermore, the complications encountered, such as bugs, missing features, and output unpredictability, rendered the selected library and its MRI-centric homologues unworthwhile.

In conclusion, due to the absence of fiber bundle generation-centric software, the development of a novel application program is recommended. Its requirements are of being automatable, ergonomic, functionally independent, coded in Python, open source, and parameterized at a minimum with the axon distribution, the biological neural network and the equivalent of Simulation Generator's parameters.

1 Introduction

The goal of the project Fiber Config Generator is to create a 3D white matter fiber crossing simulator to generate synthetic data in order to validate our new algorithms for processing of light sheet microscopy data [1]. To do so, the project was divided into two parts: identify potential libraries to delegate the white matter phantom generation to and single out one according to predefined selection criteria (1), and develop an application that utilizes the selected library in order to generate high-level fiber configurations (2). The first part was achieved in the course "initiation to research" that preceded this internship, back in the winter 2022 academic term [2]. The second part is the objective of this internship, which takes place in the summer 2022 academic term [3, 4].

In the first part of the project, it was realized that there was an absence of fiber bundle generation-centric software. Consequently, non-fiber-centric software was included in the search and MRI-centric ones were the only group to come close to the project's needs (M. Descoteaux & A. Valcourt Caron, virtual conversation, 21 March 2022). The project's requirements at that time were automation capability, ergonomics, functional independence, and being coded in Python for simplicity due to the fact that it is the main programming language used within the LINUM [2]. Thus, the library Simulation Generator was identified and then selected.

Simulation Generator is a high-level Python API to voXSim. voXSim is an amelioration of Fiberfox which, contrarily to the latter, allows its users to control the white matter bundle generation through the command-line interface (CLI) [6, 7]. There are no Simulation Generator equivalents for Fiberfox.

2 Methods

GitHub was chosen as the host platform for the project's source code repository because it is free, publicly available and cloud-based, and it features free runners mainly used for DevOps practices. In order to be used by anyone or integrated with a bigger software, the project has to respect a minimum level of code quality. To do so, a continuous integration (CI) workflow was set up. This workflow begins with cleaning the code, then proceeds to perform a static analysis on it.

In the static analysis workflow, a type checking pass is run. In order to work with external dependencies, their code must be type hinted. Therefore, Simulation Generator was type hinted via a custom GitHub fork [4, 6]. It was also done to grasp the conception of that library. The opportunity was taken to upgrade the library's original code. By programming these modification, bugs were found in the original code and had to be fixed.

Thereafter, Simulation Generator's dependencies were added to the project's dependency list ("requirements.txt"). Those shared with Linumpy [8], a library developed by the LINUM, were updated to the latest version. The listing of dependencies was patched as some were not installable out of the box. The reason is that their compiler versions were not directly supported by the Ubuntu versions (20.04 LTS and 22.04 LTS) used for developing the project.

Afterwards, Simulation Generator's significant fiber bundle geometry parameters were encapsulated into new data classes in the project. As the parameters were better understood, those related to MRI were extracted as constants. Then, their values were minimized in order to decrease their impact on the resource consumption and time performance during the generation of the phantom. Bugs emerged from that last action and there was a lack of documentation about the acceptable values of the constants and their precise impact on the DWI simulation. Thereupon, the values of the extracted MRI constants were debugged by trial and error.

To generate the white matter phantom, the modified newest Simulation Generator's voXSim runner was employed. Unfortunately, there were no means to know the outcome of the runner's execution. Wherefore, this feature was implemented in the custom GitHub fork. The fiber configuration generation parameter values are provided directly from the source code, due to the prototyping nature of the project. In the same manner, the fiber configuration is constructed using the builder design pattern for modularity purposes.

With the aim of simulating 3D microscopy data from output fiber configurations, a point spread function (PSF) was generated using the tool PSF Generator [9]. Then, the class movie_simulator [10] from the TRAIT2D software package [11] was integrated directly into the project's code and modernized. The interpretation of .fib file format, which is the undocumented voXSim file format of generated fiber bundles, was being worked upon when it was interrupted because the internship finished.

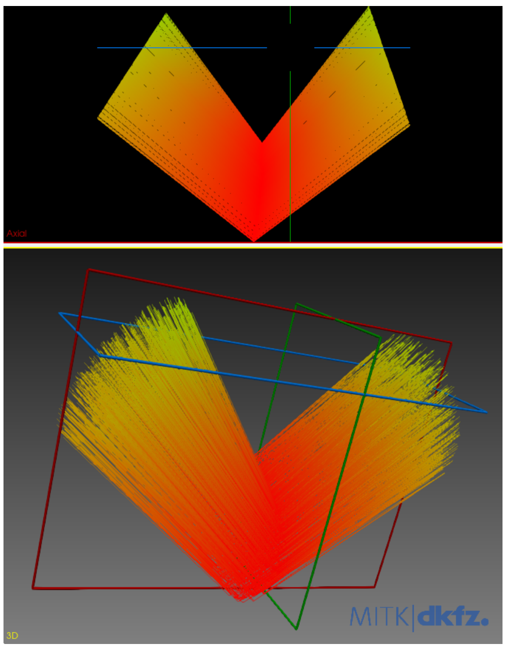
3 Results

In the first part of the project, the software library Simulation Generator was singled out as the one that is best suited for the project's needs of automation capability, ergonomics, functional independence, and being coded in Python for simplicity due to the fact that it is the main programming language employed within the LINUM [2]. These requirements sufficed the project's goal at the time, as the LINUM team lacked firsthand experience with third-party white matter phantom simulation software. However, as we patched, fixed and modernized Simulation Generator through a custom GitHub fork [4, 6], the amount of complications encountered and the required efforts to mend them increased monumentally.

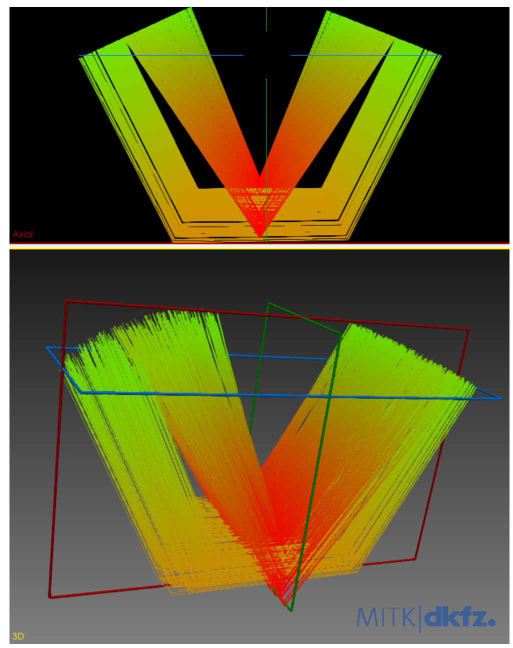
The following inadequacies were discerned:

- The source code of voXSim is inaccessible.
- A lot of time and efforts must be undertaken to patch, fix and adapt the library. These would be greatly diminished, if a CI workflow with the purpose of imposing a minimum level of code quality was implemented in the software's repository.
- Simulation Generator does not have any feature related to the new requirements of the project.

- Its strong cohesion with MITK Diffusion Fiberfox pertains to its diffusion MRI (DWI) roots. It is impossible to truly dissociate the brain white matter phantom from its DWI simulation, ergo a considerable amount of computer performance is wasted on this unneeded feature when generating a phantom.
- The generated geometric shape is unpredictable. Unfortunately, through trial and error, generating a simple V-shaped bundle with three anchors seems impossible. See the Figure 1a and its code Listing 1. It is possible to generate a simple V-shaped bundle with more than three anchors. However, as seen in Figure 1b and its code Listing 2, the anchors spatial positions offer next to no control over the actual shape of the resulting bundle.
- There is a lack of important documented features. Simulation Generator does not offer a possibility to modify the tension, bias and continuity of the centroid, contrarily to the description on the concepts documentation page [6]. Although there is a piece of code about those crucial features, it is related to the XML persistence of the DWI simulation. In that piece of code, the parameters pertaining to the said features are in effect never set, and thus there is a possibility that may not be implemented in voXSim.



(a) Satisfactorily generated V-shaped fiber bundle



(b) Erroneously generated V-shaped fiber bundle

Figure 1: The unpredictability of Simulation Generator's fiber bundle generation. Displayed in MITK Diffusion Fiberfox [7].

```
anchors = [
    (0.0, 0.0, 0.0),
    (0.25, 0.15, 0.0),
    (0.5, 0.5, 0.0),
    (0.75, 0.15, 0.0),
    (1.0, 0.0, 0.0),
]
```

Listing 1: Centroid anchors used to generate the fiber bundle in Figure 1a

```
anchors = [
    (0.0, 0.0, 0.0),
    (0.5, 0.8, 0.0),
    (1.0, 0.0, 0.0),
]
```

Listing 2: Centroid anchors used to generate the fiber bundle in Figure 1b

After careful deliberation, it was decided to abandon Simulation Generator because of critical bugs, and its utility does not fit with the project's goal, especially since the main purpose of that tool is to simulate DWI signals on generated fiber bundles. Simultaneously, stricter requirements and more specific to the fields of tractography and biomedical microscopy were defined. Thenceforth, in order to delegate the white matter phantom generation to an external library, in addition to the preceding general requirements, it is imperative that it's completely accessible and open to modifications, and that it parameterizes and allows its users to control the axon distribution and biological neural network. Thus, Simulation Generator is incompatible with the new sine quibus non.

Explicitly, the new and more specific parameters are defined like so:

- **Axon distribution:** Be able to define the spatial distribution of the axons within the constrained shape of a bundle. By implementing a simple strategy design pattern, the default programmed distribution procedure could be, per bundle, swapped with an AI algorithm trained for a specific neuronal morphology.
- **Biological neural network:** Be able to subdivide individual axons into multiple paths.

4 Conclusion

In conclusion, in order to create a 3D white matter fiber crossing simulator to generate synthetic data, the usage of external MRI-centric tools with white matter generation features to circumvent the absence of fiber bundle generation-centric software (M. Descoteaux & A. Valcourt Caron, virtual conversation, 21 March 2022) ought to be avoided. Accordingly, the development of a novel application program is recommended. Its requirements are of being automatable, ergonomic, functionally independent, coded in Python, open source, and parameterized at a minimum with the axon distribution, the biological neural network and the equivalent of Simulation Generator's parameters.

Acknowledgments

I would like to thank my research professor-supervisor Joël Lefebvre who took me under his wing and trained me in the fields of image processing and neuroimaging.

References

[1] Gaël De Oliveira-Sicard, Jingxing Tu, Frédéric Massi, and Joël Lefebvre. Détection au niveau cellulaire de l'orientation 3D de la matière blanche pour la tractographie basée sur la microscopie par nappe de lumière. In *3e colloque scientifique du CERMO*, 2021. URL: `https://linum.info.uqam.ca/fr/recherche/2021-stage-deOliveira-Sicard`.

[2] Benoît Dubreuil and Joël Lefebvre. Rapport final: INF6200, Initiation à la recherche. Université du Québec à Montréal, 2022. URL: `https://github.com/linum-uqam/inf6200-h2022-benoit-dubreuil/blob/main/report/2022_inf6200_benoit_dubreuil.pdf`.

[3] Benoît Dubreuil. Fiber Config Generator, 2022. Accessed: 2022-09-05. URL: `https://github.com/linum-uqam/inm5803-ete2022-benoit-dubreuil`.

[4] Benoît Dubreuil. Fiber Config Generator: fork of Simulation Generator, 2022. Accessed: 2022-09-05. URL: `https://github.com/linum-uqam/voxsim`.

[5] Maxime Descoteaux and Alex Valcourt Caron. Are there any other tools than fiberfox to generate white matter fiber bundles? virtual conversation, March 2022.

[6] Alex Valcourt Caron. Simulation Generator, 2022. Accessed: 2022-06-16. URL: `https://github.com/AlexVCaron/voxsim`.

[7] Peter F. Neher, Frederik B. Laun, Bram Stieltjes, and Klaus H. Maier-Hein. Fiberfox: facilitating the creation of realistic white matter software phantoms. *Magnetic Resonance in Medicine*, 72(5):1460–1470, 2014. MITK Diffusion Fiberfox URL: `https://github.com/MIC-DKFZ/MITK-Diffusion`. `doi:10.1002/mrm.25045`.

[8] LINUM. Linumpy, 2022. Accessed: 2022-06-29. URL: `https://github.com/linum-uqam/linumpy/`.

[9] Hagai Kirshner and Daniel Sage. PSF Generator, 2017. Downloaded: 2022-08-29. URL: `http://bigwww.epfl.ch/algorithms/psfgenerator/`.

[10] Joël Lefebvre. TRAIT2D: movie_simulator, 2021. Accessed: 2022-09-01. URL: `https://github.com/Eggeling-Lab-Microscope-Software/TRAIT2D/blob/c0e78a61f58bd12f5e2b63e99dbc5a130ef740bb/trait2d/simulators.py#L412`.

[11] Francesco Reina, John Maxwell Andreas Wigg, Mariia Dmitrieva, Bela Vogler, Joël Lefebvre, Jens Rittscher, and Christian Eggeling. TRAIT2D: a Software for Quantitative Analysis of Single Particle Diffusion Data, June 2021. URL: `https://github.com/Eggeling-Lab-Microscope-Software/TRAIT2D`, `doi:10.5281/zenodo.4725268`.