

# fastPLI: A Fiber Architecture Simulation Toolbox for 3D-PLI

Felix Matuschke<sup>1</sup>, Katrin Amunts<sup>1, 2</sup>, and Markus Axer<sup>1</sup>

<sup>1</sup> Institute of Neuroscience and Medicine (INM-1), Forschungszentrum Jülich GmbH, 52425, Jülich, Germany <sup>2</sup> Cécile and Oskar Vogt Institute for Brain Research, University Hospital Düsseldorf, University of Düsseldorf, 40204, Düsseldorf, Germany.

DOI: [10.21105/joss.03042](https://doi.org/10.21105/joss.03042)

## Software

- [Review ↗](#)
- [Repository ↗](#)
- [Archive ↗](#)

---

Editor: Olivia Guest ↗

Reviewers:

- [@vigji](#)
- [@glyg](#)
- [@RealPolitiX](#)

Submitted: 18 December 2020

Published: 03 May 2021

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Statement of need

3D Polarized Light Imaging (3D-PLI) is a microscopic neuroimaging technique used to study the nerve fiber architecture in unstained histological brain sections at the micrometer scale ([Axer et al., 2011](#)). It provides image contrast for fibers and fiber tracts, and ultimately enables reconstruction of 3D nerve fiber orientations. The physical effect behind 3D-PLI is the optical property of the nerve fibers called birefringence. Due to this intrinsic birefringence, it is possible to use polarized light, pass it through a thin brain section and observe the change of the polarization state of light. This change is directly related to the 3D orientation of the fibers and also provides strong contrasts between fibers and other tissue components.

To understand the influence of the underlying fiber structure, simulations are an essential tool. They allow testing different hypotheses, knowing the ground truth. It has already been shown that simulations with scattered light within tissue sections require models with irregularities to mimic the behavior of the scattered light. This knowledge can now be used to understand structures such as fiber crossings that have been difficult to interpret in 3D-PLI ([Menzel et al., 2020](#)).

In addition, the generated nerve fiber models can be used in other imaging simulation techniques such as diffusion magnetic resonance imaging (dMRI).

In recent years, various software tools have been developed to design fibre models. Nerve fiber modeling is commonly used in dMRI. But many modeling techniques do not use volumetric representations, especially collision-free ones. In the last decade, an increasing number of algorithms for non-trivial overlapping structures have been developed ([Altendorf & Jeulin, 2011](#); [Chapelle et al., 2015](#); [Ginsburger et al., 2019](#); [Mingasson et al., 2017](#)). While these algorithms are specialized in their field, *fastPLI* also provides a dedicated tool for 3D-PLI simulation based on linear optics. Here, the focus is on the previously developed algorithm ([Matuschke et al., 2019](#)), which provides a fast method to generate collision-free results for white matter structures in the brain.

Different types of simulations for polarized light are for example described in ([Jiang & Li, 2020](#); [RamellaRoman2005](#); [Turnhout et al., 2009](#)). However, to our knowledge, none of these techniques have been used to simulate the effects of polarized light on nerve fibers, except simPLI ([Dohmen et al., 2015](#)), which is included in this toolbox.

## Summary

*fastPLI* is an open source toolbox based on Python and C++ for modeling myelinated axons, i.e. nerve fibers and simulating the results of measurement of fiber orientations with a polarization microscope using 3D-PLI.

The *fastPLI* package includes the following modules:

1. **Fiber Modelling Modules:** A detailed 3D modelling of nerve fibers at the micrometer level is essential as input for the measurement simulation. In order to recreate biological tissue as a model, it is important that the nerve fibers do not spatially overlap. We have decided to implement a solver module that takes any configuration of fiber bundles as input and converts it over several iterations into a collision-free configuration. In order to generate collision free fiber arrangements, a dedicated algorithm to prohibit such overlaps has been developed ([Matuschke et al., 2019](#)).
2. **Simulation Module:** The 3D-PLI simulation is based on Stokes vector and Müller matrix approaches as described in ([Dohmen et al., 2015](#); [Menzel et al., 2015](#)). For the simulation the polarimetric setup can be equipped with a tiltable specimen stage ([Aixer et al., 2011](#); [Schmitz et al., 2018](#)). By this means the brain section can be scanned from oblique views which adds important information to unambiguously analyze the 3D fiber orientation.
3. **Analysis Module:** The resulting simulated measurements (i.e., image stacks of a section acquired at different polarizing filter rotation angles and, optionally, at different oblique views) can be processed similarly to the real, experimental 3D-PLI ([Aixer et al., 2011](#); [Schmitz et al., 2018](#)).

All computationally intensive calculations are optimized either with *numba* on the Python side or with multithreading *C++* algorithms, which can be accessed via *pybind11* inside the Python package ([Jakob et al., 2017](#); [Lam et al., 2015](#)). Additionally, the simulation module supports the Message Passing Interface (MPI) to facilitate the simulation of very large volumes on multiple computer nodes.

## Installation

The *fastPLI* package has to be built with a *C++17* compiler. A Makefile allows a simple local installation. It generates the necessary libraries inside a build folder and a matching *setup.py* file, which can be used for a second installation process with pip.

```
make fastpli  
pip3 install .
```

All necessary configurations are handled in the background with CMake. All required software libraries are listed in the [software repository](#).

## Usage & Examples

A more detailed description of *fastPLI* including examples, jupyter notebooks and tutorials can be found in the software package,<sup>1</sup> at the wiki pages<sup>2</sup> and the API documentation.<sup>3</sup>

```
python3 examples/sandbox.py  
python3 examples/solver.py  
python3 examples/simulation.py  
python3 examples/optic_chiasm.py
```

<sup>1</sup><https://github.com/3d-pli/fastpli>

<sup>2</sup><https://github.com/3d-pli/fastpli/wiki>

<sup>3</sup><https://3d-pli.github.io/fastpli/>

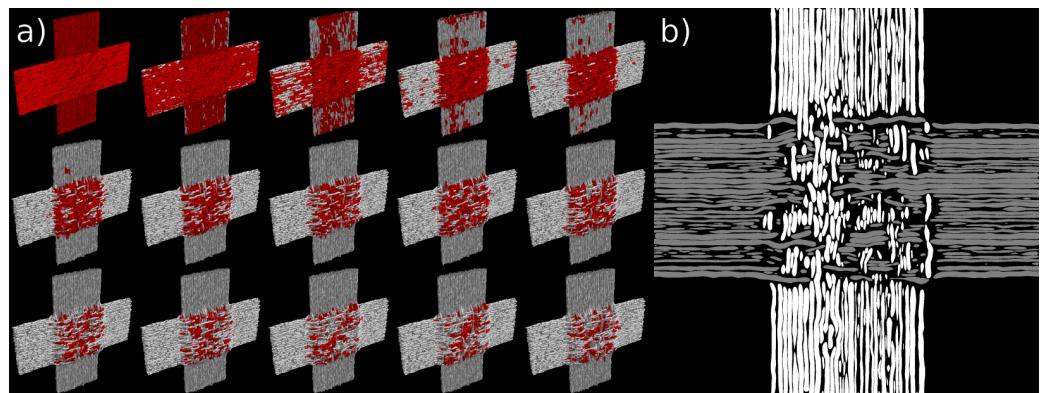
## Fiber Modelling

Two modules exist to allow the user to build non-colliding white matter nerve fiber models: `fastpli.model.sandbox` and `fastpli.model.solver`.

The `fastpli.model.sandbox` module contains multiple functions to build from coordinates single individual fibers up to fiber bundles composed of individual fibers.

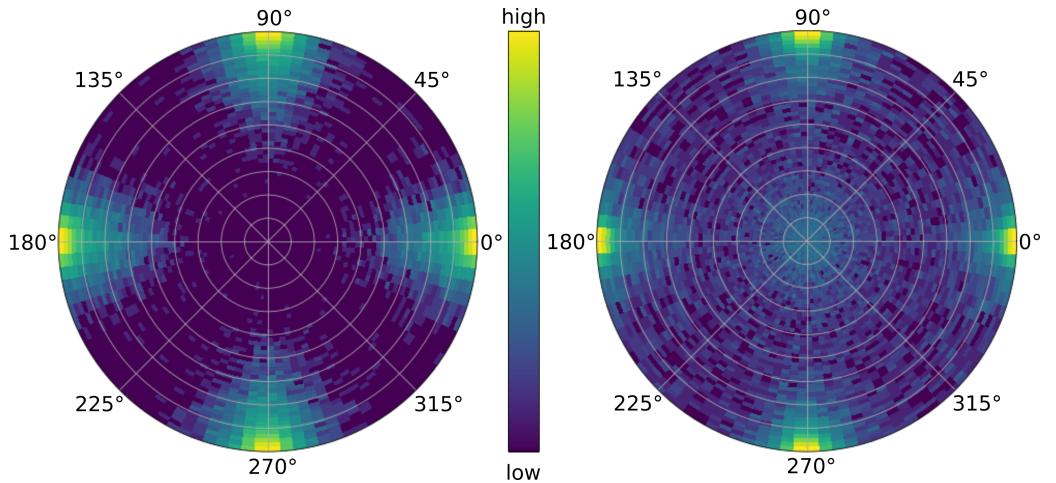
The `fastpli.model.solver` module contains a `Solver` class to generate non-colliding fibers constellations from an initial input dataset (Matuschke et al., 2019).

An example of the solving process is shown in fig. 1 a). The red colored segments indicate that a collision with this segment is detected. With further iterations, the number of collisions decreases. At the end, a collision-free configurations results. Figure 1 b) shows the cross-section through the resulting fiber configuration. Each fiber bundle has a different gray value.



**Figure 1:** a) Solving process of two crossing fiber bundles. Individual colliding segments are colored in red. b) Cross-section of the collision solving process.

The resulting orientations for each segment can be then visualized in a polar histogram (see fig. 2).

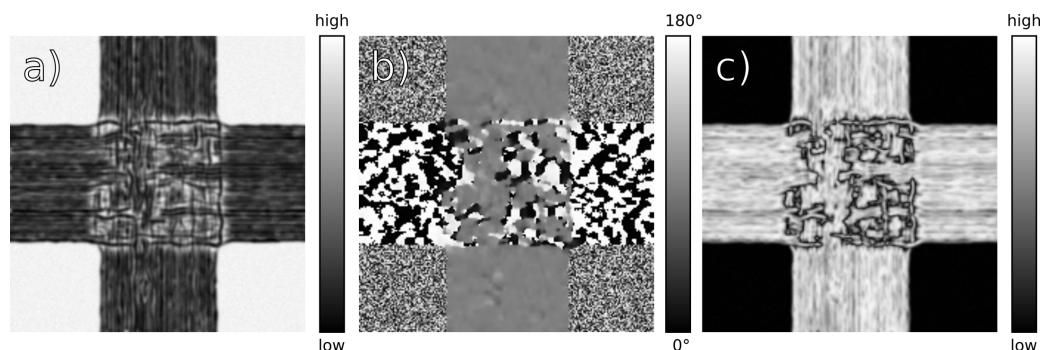


**Figure 2:** orientation distribution a) initial, b) resulting fiber configuration.

## Simulation and Analysis

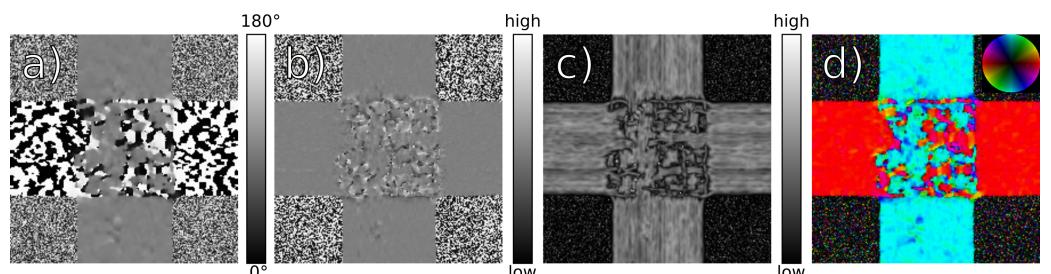
The module `fastpli.simulation` contains the class `Simpli`. This class contains all tools required to simulate the 3D-PLI setup (Dohmen et al., 2015; Menzel et al., 2015) and to analyze the generated images. Final results include the stack of images as well as the derived modalities referred to as transmittance, direction, retardation, inclination, relative thickness, and 3D fiber orientation (FOM) maps (Schmitz et al., 2018).

An example of the simulation results is shown in fig. 3. a) indicates the resulting transmittance map. It represents the mean light intensity transmission through the cut tissue. b) represents the fiber direction in the tissue plane. c) shows the resulting retardation, which is a measure of the amplitude of the resulting sinusoidal signal from the simulation in each pixel.



**Figure 3:** PLI modalities: a) transmittance, b) direction, c) retardation

The data can be further analyzed with an advanced tilting simulation and analysis 4. a) represents the fiber direction in the tissue plane. b) visualizes the fiber inclination in the tissue plane. c) shows the relative thickness of the fibers within a pixel volume. d) represents the resulting FOM. The colors encode the 3D orientation.



**Figure 4:** Tilting analysis results: a) direction, b) inclination, c) relative thickness, d) FOM

The analysis is accessible inside the simulation module via a `pipeline` inside the `fastpli.simulation.Simpli` class. For further information see `examples/simulation_pipeline.py`.

## Acknowledgements

This study was funded by the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 785907 (Human Brain Project SGA2) and No. 945539 (Human Brain Project SGA3). We gratefully acknowledge the computing time granted through JARA-HPC on the supercomputer JURECA (Jülich Supercomputing Centre, 2018) at Forschungszentrum Jülich, Germany.

## Author Contributions

F.M. developed the software and wrote the documentation. M.A. provided guidance throughout the project. K.A. and M.A. supervised the project. F.M. wrote the manuscript with input from all authors.

## References

- Altendorf, H., & Jeulin, D. (2011). Random-walk-based stochastic modeling of three-dimensional fiber systems. *Physical Review E*, 83(4). <https://doi.org/10.1103/physreve.83.041804>
- Axer, M., Grässel, D., Kleiner, M., Dammers, J., Dickscheid, T., Reckfort, J., Hütz, T., Eiben, B., Pietrzyk, U., Zilles, K., & Amunts, K. (2011). High-resolution fiber tract reconstruction in the human brain by means of three-dimensional polarized light imaging. *Frontiers in Neuroinformatics*, 5. <https://doi.org/10.3389/fninf.2011.00034>
- Chapelle, L., Lévesque, M., Brøndsted, P., Foldschack, M. R., & Kusano, Y. (2015). Generation of non-overlapping fiber architecture. *Proceedings of the 20th International Conference on Composite Materials*.
- Dohmen, M., Menzel, M., Wiese, H., Reckfort, J., Hanke, F., Pietrzyk, U., Zilles, K., Amunts, K., & Axer, M. (2015). Understanding fiber mixture by simulation in 3D polarized light imaging. *NeuroImage*, 111, 464–475. <https://doi.org/10.1016/j.neuroimage.2015.02.020>
- Ginsburger, K., Matuschke, F., Poupon, F., Mangin, J.-F., Axer, M., & Poupon, C. (2019). MEDUSA: A GPU-based tool to create realistic phantoms of the brain microstructure using tiny spheres. *NeuroImage*, 193, 10–24. <https://doi.org/10.1016/j.neuroimage.2019.02.055>
- Jakob, W., Rhinelander, J., & Moldovan, D. (2017). *pybind11 – seamless operability between c++11 and python*.
- Jiang, Y., & Li, Z. (2020). Monte carlo simulation of mueller matrix of randomly rough surfaces. *Optics Communications*, 474, 126113. <https://doi.org/10.1016/j.optcom.2020.126113>
- Jülich Supercomputing Centre. (2018). JURECA: Modular supercomputer at Jülich Supercomputing Centre. *Journal of Large-Scale Research Facilities*, 4(A132). <https://doi.org/10.17815/jlsrf-4-121-1>
- Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC - LLVM 15*. <https://doi.org/10.1145/2833157.2833162>
- Matuschke, F., Ginsburger, K., Poupon, C., Amunts, K., & Axer, M. (2019). Dense fiber modeling for 3D-polarized light imaging simulations. *Advances in Parallel Computing*, 34(Future Trends of HPC in a Disruptive Scenario), 240–253. <https://doi.org/10.3233/APC190017>
- Menzel, M., Axer, M., Raedt, H. D., Costantini, I., Silvestri, L., Pavone, F. S., Amunts, K., & Michielsen, K. (2020). Toward a high-resolution reconstruction of 3D nerve fiber architectures and crossings in the brain using light scattering measurements and finite-difference time-domain simulations. *Physical Review X*, 10(2). <https://doi.org/10.1103/physrevx.10.021002>
- Menzel, M., Michielsen, K., Raedt, H. D., Reckfort, J., Amunts, K., & Axer, M. (2015). A jones matrix formalism for simulating three-dimensional polarized light imaging of brain

tissue. *Journal of The Royal Society Interface*, 12(111), 20150734. <https://doi.org/10.1098/rsif.2015.0734>

Mingasson, T., Duval, T., Stikov, N., & Cohen-Adad, J. (2017). AxonPacking: An open-source software to simulate arrangements of axons in white matter. *Frontiers in Neuroinformatics*, 11. <https://doi.org/10.3389/fninf.2017.00005>

Schmitz, D., Muenzing, S. E. A., Schober, M., Schubert, N., Minnerop, M., Lippert, T., Amunts, K., & Aixer, M. (2018). Derivation of fiber orientations from oblique views through human brain sections in 3D-polarized light imaging. *Frontiers in Neuroanatomy*, 12. <https://doi.org/10.3389/fnana.2018.00075>

Turnhout, M. C. van, Kranenbarg, S., & Leeuwen, J. L. van. (2009). Modeling optical behavior of birefringent biological tissues for evaluation of quantitative polarized light microscopy. *Journal of Biomedical Optics*, 14(5), 054018. <https://doi.org/10.1117/1.3241986>