



FAConstructor: an interactive tool for geometric modeling of nerve fiber architectures in the brain

Jan André Reuter¹ · Felix Matuschke¹ · Miriam Menzel¹ · Nicole Schubert¹ · Kévin Ginsburger³ · Cyril Poupon³ · Katrin Amunts^{1,2} · Markus Axer¹

Received: 25 February 2019 / Accepted: 5 August 2019 / Published online: 11 August 2019
© The Author(s) 2019

Abstract

Purpose The technique *3D polarized light imaging* (3D-PLI) allows to reconstruct nerve fiber orientations of postmortem brains with ultra-high resolution. To better understand the physical principles behind 3D-PLI and improve the accuracy and reliability of the reconstructed fiber orientations, numerical simulations are employed which use synthetic nerve fiber models as input. As the generation of fiber models can be challenging and very time-consuming, we have developed the open source *FAConstructor* tool which enables a fast and efficient generation of synthetic fiber models for 3D-PLI simulations.

Methods The program was developed as an interactive tool, allowing the user to define fiber pathways with interpolation methods or parametric functions and providing visual feedback.

Results Performance tests showed that most processes scale almost linearly with the amount of fiber points in *FAConstructor*. Fiber models consisting of < 1.6 million data points retain a frame rate of more than 30 frames per second, which guarantees a stable and fluent workflow. The applicability of *FAConstructor* was demonstrated on a well-defined fiber model (*Fiber Cup* phantom) for two different simulation approaches, reproducing effects known from 3D-PLI measurements.

Conclusion We have implemented a user-friendly and efficient tool that enables an interactive and fast generation of synthetic nerve fiber configurations for 3D-PLI simulations. Already existing fiber models can easily be modified, allowing to simulate many different fiber models in a reasonable amount of time.

Keywords Nerve fiber structure · White matter phantom · Interactive visualization · Simulation model · 3D polarized light imaging

Jan André Reuter and Felix Matuschke have contributed equally to this work.

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s11548-019-02053-6>) contains supplementary material, which is available to authorized users.

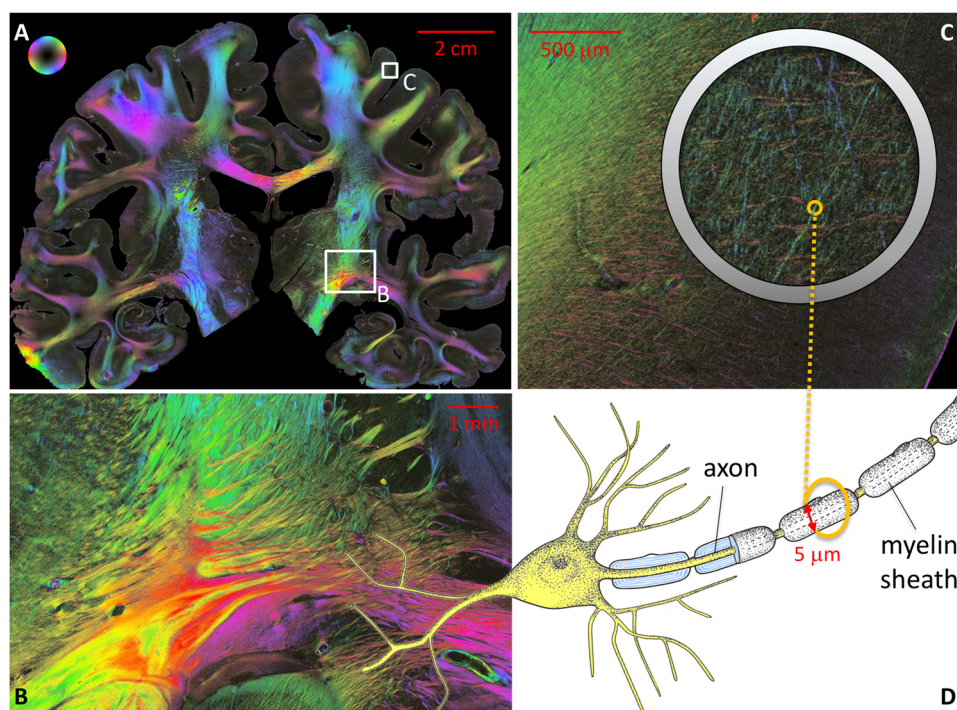
✉ Jan André Reuter
j.reuter@fz-juelich.de

- ¹ Institute of Neuroscience and Medicine (INM-1), Forschungszentrum Jülich GmbH, 52425 Jülich, Germany
- ² Cécile and Oskar Vogt Institute for Brain Research, University Hospital Düsseldorf, University of Düsseldorf, 40204 Düsseldorf, Germany
- ³ Imaging and Spectroscopy Laboratory, CEA Saclay, Neurospin, Gif-sur-Yvette, France

Introduction

Decoding the highly complex distribution and connectivity of neurons in the human brain is of major interest in neuroscience. Various neuroimaging techniques arose in recent years, each revealing specific characteristics of the brain's nerve fiber architecture (e.g. individual myelinated axons, bundles of fibers, pathways) [3]. The most prominent techniques are diffusion magnetic resonance imaging (dMRI) [29], lightsheet fluorescence microscopy [26], (polarization-sensitive) optical coherence tomography [13,27], and *3D polarized light imaging* (3D-PLI) [1,2,30]. None of these techniques has a real ground truth to compare with, leading indispensably to the idea of cross-modality research and technology-specific simulation approaches. Well-defined bio-mimicking numerical phantoms subjected to dMRI simulators have already been proven to be essential tools for better understanding of the underlying physical

Fig. 1 Fiber orientation maps of a coronal section through the human brain obtained with 3D-PLI. The determined nerve fiber orientations are color-coded as indicated by the color sphere on the top left. **a** Whole brain section with long- and short-range fibers at a glance. **b** Deep white matter fascicles of fibers (close-up of the region shown in **a**). The fascicle dimensions vary from mm to μm scales. **c** Cortical fiber tracts and individual fibers (close-up of the region shown in **a**). Typical fiber diameters lie below $10\ \mu\text{m}$ [12]. **d** Schematic drawing of a neuron with a myelinated axon



mechanisms (e. g. water diffusion, magnetic field dependencies) and, ultimately, improving the interpretation of the measured signals [7,9,10,14].

In the present study, we focus on fiber modeling and simulation aspects of the microscopic neuroimaging technique 3D-PLI, which enables the reconstruction of the nerve fiber architecture of whole postmortem brains at micrometer resolution [1,2,30]. With 3D-PLI, the spatial orientations of fibers are derived from unstained histological brain sections in a polarimetric system that measures the local birefringence. This birefringence is mainly caused by the highly ordered molecular arrangement of the myelin sheaths wrapped around many axons in the white matter. Figure 1 demonstrates the unique potential of 3D-PLI to carve out the entire spectrum of real nerve fiber structures ranging from individual cortical fibers (μm scale) to long-distance fiber bundles or fascicles (mm to cm scales), even within one coronal human brain section. However, in brain regions where fiber structures show dispersed orientations within one volume element, the derived orientations might be misinterpreted. In order to underpin or improve fiber orientation estimations by means of 3D-PLI, two complementary numerical simulation approaches have recently been implemented: (i) the matrix calculus simulation (*simPLI*) [6,16,17] and (ii) the *finite-difference time-domain* (FDTD) simulation [18,19].

Both types of simulation require numerical fiber phantoms as input, describing anatomical and optical properties and mimicking brain tissue to a certain degree of real-

ism. In the present study, the focus was on white matter phantoms defined by numerous fiber models with specified fiber diameter, myelin thickness, spatial course, refractive indices, etc. The generation of such fiber models and resultant phantoms, however, is challenging and time-consuming, especially when targeting complex and large-scale models.

We introduce the easy-to-use open source software tool *Fiber Architecture Constructor* (*FAConstructor*) enabling fast and efficient generation of white matter phantoms. It allows to define *fiber objects* (e. g. individual fibers, tracts, bundles) which can be rotated, (re)moved, duplicated, and spatially integrated into complex large-scale phantoms. To enable modification of a large amount of fibers, fiber objects can be arranged in groups. By this means, the brain's multi-scale characteristics can be built from small numerical tissue units complying with the specific requirements posed by the different simulation approaches. The implemented graphical user interface allows the user to examine and interact with the generated fiber models in 3D space and provides direct visual feedback prior to simulation.

We demonstrate the applicability of *FAConstructor* by using the example of the so-called *Fiber Cup* phantom [7], which has initially been designed for an international competition on fiber reconstruction approaches (i. e. tractography algorithms) based on dMRI-like data. The resulting phantom is finally subjected to both the *simPLI* and the FDTD simulations.

The fiber architecture constructor (FAConstructor)

The *FAConstructor* is a simple, effective, and interactive tool to generate and visualize fiber objects as input for simulations or similar purposes. An open source version of the code and a detailed description of compilation and usage are provided through the following GitHub repository: <https://github.com/3d-pli/FAConstructor>.

The tool allows an exact definition of curves using parametric functions or interpolation methods. Each curve contains a number of data points consisting of coordinates in three-dimensional space and radii for each data point. By this means, individual fibers or entire fiber bundles will be described in the following and referred to as *fiber objects*.

Generation of fiber objects

A key feature of the *FAConstructor* is the generation of fiber objects with defined pathways and geometries. The user is able to define arbitrary fiber pathways using either interpolation methods with given coordinates or parametric functions. As the tool allows fiber objects to overlap, postprocessing might become necessary to avoid fiber intersections.

The manual input of three-dimensional coordinates allows a precise definition of fiber pathways. A different radius can be assigned to each coordinate, allowing each object to grow or shrink along its trajectory. In addition, the user can apply cubic spline interpolation to smooth the fiber object. In this case, both coordinates and radii are interpolated.

A major advantage of the developed tool is that individual fiber objects can be modified without need to regenerate the whole model. Both whole fiber objects and individual fiber coordinates can be rotated, translated, or duplicated. In this way, the user can modify an already existing fiber model and build large-scale models without much effort. Furthermore, it is possible to group several fiber objects together and change or duplicate them at the same time. The groups can be exported for future usage.

To facilitate the generation of nested fiber objects, the *FAConstructor* allows to fill a parent fiber object with multiple child objects, which can be filled by other fiber objects and so on. Here, we only consider fiber objects that were once filled with individual fibers, so we refer to the parent fiber object as fiber bundle and to the child objects as fibers.

When filling a fiber bundle with fibers, the cross section with minimum radius is used as starting point. The fibers are arranged on a triangular grid (see Fig. 2b), and the radius of every fiber and the minimum distance between neighbored fibers are defined by the user. The triangular grid is rotated from one data point to the next using the tangents of each point and computing the rotation angle using the Rodrigues formula [5].

If the radius increases along the course of the fiber bundle, the distance between the fibers increases accordingly. This guarantees that the fibers in the bundle do not intersect with each other, provided that the user-defined minimum distance does not fall below the diameter of the fibers. The generated fibers are stored as a group, allowing efficient modifications of the fiber bundle.

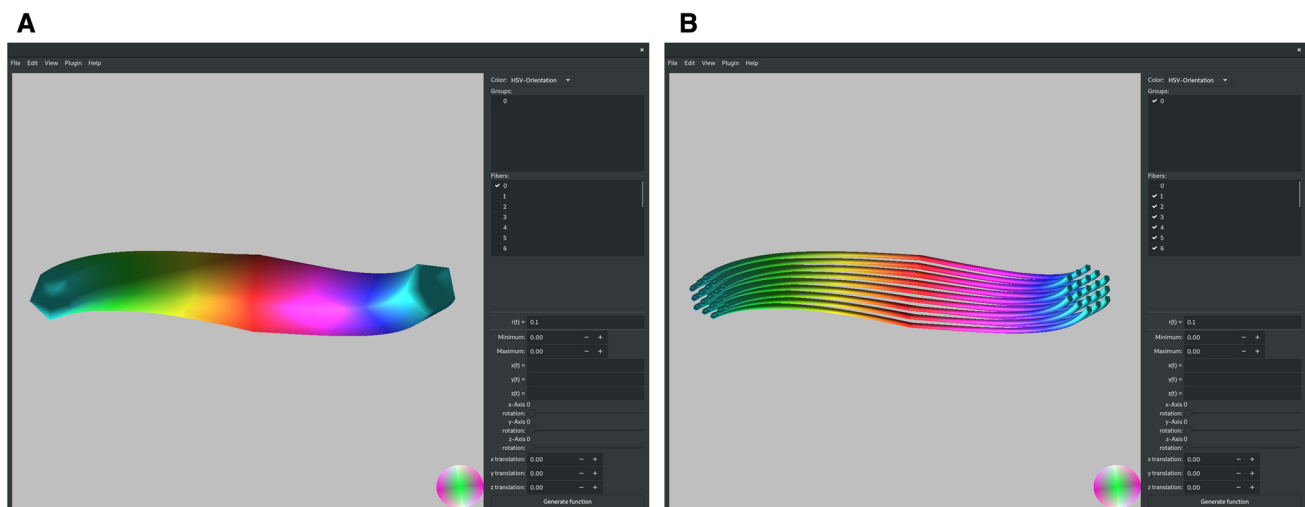


Fig. 2 **a** Fiber object representing a fiber bundle (envelope) as generated by *FAConstructor*. The main monitor shows the fiber object from a certain view. Objects can be rotated, scaled, or translated by the user. Lists of all generated fibers and groups are shown on the top right. Objects can be selectively shown by the corresponding checkboxes next to their labels. The parametric functions describing the coordinates and radii of the displayed objects are also displayed.

Here, the object has been generated by the parametric functions $(x(t) = 50 \cdot \sin(t \cdot \pi), y(t) = 50 \cdot \cos(t \cdot \pi), z(t) = 50 \cdot t, r(t) = -|15 \cdot t| + 15, t \in [-0.5, 0.5])$. All other options are accessible through the menu bar at the top. **b** Fiber object filled with fibers. With increasing radius of the fiber object, the distance between the fibers increases accordingly

Coordinates and radii of the fiber objects are defined as single-precision floating-point values, which can be saved either in a hierarchical file system-like data format (HDF5, <https://www.hdfgroup.org>) or, alternatively, in a plain text file format. HDF5 is advantageous, because it allows to handle metadata, it has no limitation on the number or size of data objects in the collection, giving great flexibility for big data, and it provides high-performance I/O with a rich set of integrated performance features.

Rendering

As the *FAConstructor* allows to generate fiber objects interactively, an efficient rendering of the data is crucial. Therefore, a new frame is only rendered when the user interacts with the fiber objects by rotating or translating them, allowing fluent interaction. Rendering is realized with the *OpenGL* 3.3 framework of *Qt5* [4,25] which enables the use of widgets for applying options and generating fibers, while rendering is done in a single central widget. In addition, *OpenMP* [22] can be used optionally for several tasks granting performance improvements. Fiber objects and individual fibers are approximated by cylindrical shapes using the defined coordinates and radii. As circular objects cannot be rendered without approximation, they are represented by n -sided regular polygons. Even when displaying a large amount of fibers, the tool maintains a sufficient frame rate: The parameter n is dynamically adjusted to ensure a frame rate above 30. In addition, double-buffered *V-Sync* is used to reduce stutter which might occur when encountering a fluctuating frame rate. Nonvisible fiber objects are culled out using the *Cohen–Sutherland* algorithm [8]. As the fiber object trajectory is a line until drawn in the *OpenGL* shaders, this algorithm can be applied. The user is able to interact with the rendering area by moving, rotating, and scaling the shown content. When losing focus of the current object, the user can reset the view, displaying the whole object in the center. It is also possible to display different orthogonal views of the model and walk through the model

using keyboard shortcuts. The user is able to hide certain fiber objects or whole groups by deselecting the checkboxes next to their corresponding labels (cf. Fig. 2b). This helps to keep an overview and improves performance.

Fiber bundles are colored according to their respective orientation in three-dimensional space, using either an *RGB* (red–green–blue) or an *HSV* (hue–saturation–value) color map which are also used to visualize 3D-PLI data shown in Fig. 1. In the right bottom corner of the rendering area, a color sphere displays the orientation of the fiber objects. The sphere changes when the user rotates the model. To distinguish individual fibers, it is also possible to assign each fiber bundle a randomized color based on the fiber index.

Performance measurements

To ensure that the user can work fluently with a large number of fiber points in *FAConstructor*, performance optimizations were implemented as mentioned in “Rendering” section and tested on the generation and rendering of fiber objects. Performance measurements were performed using a PC with an Intel Core i5-3570 processor paired with an NVIDIA GTX 1070 and 16 GB RAM running Ubuntu 18.04.

Figure 3 shows the measured performance of the program. The corresponding datasets were created with the following parameters:

$$(A), (C) : x(t) = t, \quad y(t) = t, \\ z(t) = t, \quad r(t) = t, \quad t \in [0, 1, \dots, N] \quad (1)$$

$$(B) : x(t) = \sin(t), \quad y(t) = \cos(t), \\ z(t) = t, \quad r(t) = 1, \quad t \in [0, 1, \dots, N] \quad (2)$$

where $x(t)$, $y(t)$, $z(t)$ are space coordinates and $r(t)$ is the radius of the fiber object at this position. The functions describe fiber objects consisting of $N - 1$ segments. This allows to conduct performance measurements as a function of the number of objects.

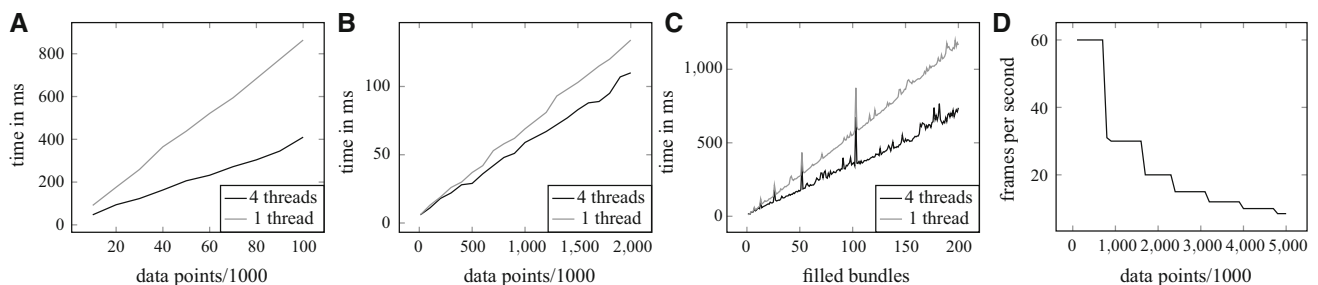


Fig. 3 Performance measurements of four scenarios during the generation of a fiber object. The x -axis shows the number of data points used for the respective measurements, and the y -axis shows the elapsed time or the frame rate. The used parameters are shown in Eqs. 1–2: **a** Generation of fiber objects by manual input. The data points are defined

and connected using cubic spline interpolation. **b** Generation of a helix using parametric functions. **c** Filling n fiber objects containing 1000 data points with 21 nested fiber objects each. The peaks in plot are most likely due to the memory utilization of the data type. **d** Frame rate while interacting with n fiber objects on screen

The test procedures to generate Fig. 3a, b, d consisted of starting the program, creating or loading the corresponding data points, and measuring the time or frame rate. Afterward, the program was closed to avoid possible influence on the following iteration. This procedure was repeated ten times for each data point. Measuring the performance when filling the fiber bundles was done by creating a dataset containing n fiber objects with 1000 data points each using Eq. 1. The fiber objects were filled with fibers with a distance of $d = 0.2$ and a radius of $r = 0.1$, yielding 21 fibers. The measured time of 20 separate runs was taken and averaged. As the program can use *OpenMP* for parallelization, both variants were measured individually.

As seen in Fig. 3a–c, the program tends to scale linearly with the produced amount of data points. The usage of *OpenMP* shows a speedup of 1.21 to 2.1 depending on the measured part of the program. The frame rate in (D) maintains the 30 fps target when the amount of fiber points lies below 1.6 million. This is due to the insufficient performance of the graphics card to display all data points simultaneously. Even when rendering around 60 million data points, 11 GiB RAM and 2.3 GiB video memory are consumed, indicating that neither is a limiting factor. When rendering, the CPU usage stays around 0% to 10%.

Usecase: Fiber Cup phantom

The *Fiber Cup* phantom introduced by Fillard et al. [7] was used for qualitative and quantitative evaluation of tractography methods in dMRI. It was inspired by nerve fiber configurations in a coronal human brain section, including regions with fiber crossings, fanning, and bending. Both the reconstruction of crossing and that of fanning fibers represent a challenge for 3D-PLI. The reconstructed phantom used for our simulations was similar to the original *Fiber Cup* phantom, only deviating slightly in the crossing angles between the fiber bundles.

The *Fiber Cup* phantom was built using parametric functions as well as user-defined fiber points with cubic spline interpolation. When generating the *Fiber Cup* phantom, some of the fiber objects partially overlap. This leads to a higher density of fibers when filling the objects. Fiber bundles 4 to 6, for example, merge into one bundle (see Fig. 4), resulting in a three times higher fiber density as compared to the transverse bundle 3.

The fiber objects were filled with equidistant seed points on a triangular grid (see “Generation of fiber objects” section), yielding 10,149 fibers in total. The fibers have diameters of $1.5 \mu\text{m}$ and lie $1.5 \mu\text{m}$ apart, and the diameter of the fiber bundles is $120 \mu\text{m}$. The result is shown in Fig. 4.

However, as two distinct nerve fibers in real brain tissue do not occupy the same point in space in a crossing

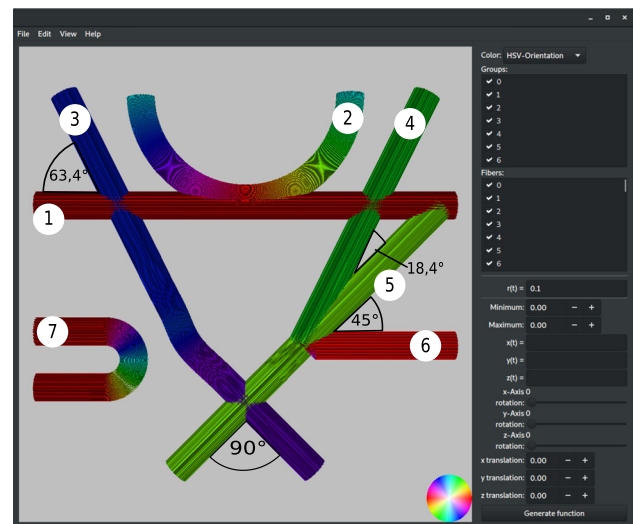


Fig. 4 *Fiber Cup* phantom filled with fibers. The different fiber bundles (numbered from 1 to 7) were generated as described in “Usecase: Fiber Cup phantom” section

situation, the generated fiber configurations were checked for such unrealistic collisions and solved using an additional in-house developed software [15] before performing the imaging/measurement simulations. Depending on the initial objects’ fraction of overlap, the density of the fibers can change during this process. Most likely, this results in a decrease in density, since the fiber objects are designed as rigid objects. In the *Fiber Cup* example, the fiber densities within the intersecting regions 1 to 3 in Fig. 5 decreased by 29% (1), 25% (2), and 3% (3), respectively.

Three-dimensional polarized light imaging (3D-PLI)

3D-PLI is a postmortem imaging technique for unstained histological brain sections. Due to the birefringence (optical anisotropy) of the nerve fibers, the state of polarization changes when light passes through brain tissue. In 3D-PLI, this change of polarization is measured with a polarimetric setup (polarizer, specimen, quarter-wave retarder, analyzer), allowing to determine the spatial orientations of the nerve fibers with micrometer resolution. The measurement and signal analysis of 3D-PLI have been described in detail by Axer et al. [2]. As the investigated brain sections are about $60 \mu\text{m}$ thick, each measured volume element contains several nerve fibers. 3D-PLI extracts only the predominant fiber orientation, and the orientations of individual nerve fibers within deep white matter structures can therefore not be determined. Inhomogeneous nerve fiber structures, like crossing fibers or regions with varying fiber densities, influence the measured birefringence signal and may lead to misinterpretations in the reconstructed fiber orientations.

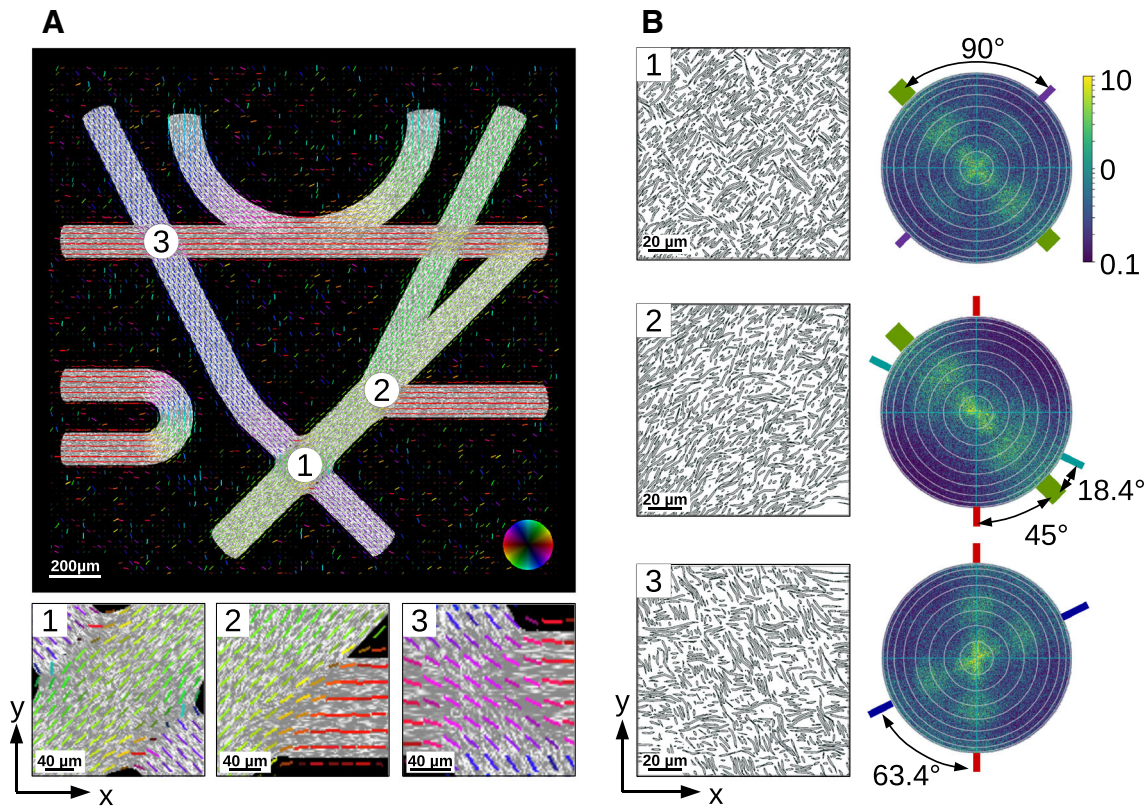


Fig. 5 **a** Vector field obtained from *simPLI* simulations of the collision-free *Fiber Cup* phantom, overlaid with the employed fiber model. The orientation of the vectors are color-coded. Visualization has been realized with in-house developed software [24]. **b** FDTD simulations of the three selected regions (fiber crossings) in the collision-free *Fiber Cup* phantom shown in (a). The images on the left show a cross section (xy -plane) through the middle of the simulated fiber volume

($128 \times 128 \times 60 \mu\text{m}^3$). The images on the right show the corresponding scattering patterns (intensity per wave vector). The white circles indicate steps of 10° , from 0° (center) to 90° (outer circle). The straight colored lines around the scattering patterns indicate the axes perpendicular to the fiber bundles in the phantom (colors were chosen according to the colors of the fiber bundles shown in a)

Simulation methods

Matrix calculus (*simPLI*) The in-house developed software *simPLI* [6] computes the change of polarization of a light beam when passing through a brain tissue sample. The light beam is modeled by Stokes vectors, the optical elements of the polarimeter and the brain tissue by Müller matrices [21]. The birefringence of the nerve fibers is modeled by optical retarder matrices, with the retarder axis oriented along the fiber axis [16]. To account for the imaging system, the image is downsampled and blurred with a Gaussian function with $\sigma = 0.714 \Delta x$ as described by Menzel et al. [16]. Finally, noise is modeled according to Schmitz et al. [23]. The resulting signals are analyzed using the *ROFL* algorithm [23], resulting in fiber orientation maps.

Finite-Difference Time-Domain (FDTD) simulation To study more complex light–matter interactions like scattering and interference, *Finite-Difference Time-Domain* (FDTD) simulations are used. With FDTD simulations, it is possible

to study light transmitted through artificial nerve fiber models as well as scattering of light [18,19]. The propagation of the electromagnetic light wave through the sample was computed by *TDME3D*TM, a massively parallel three-dimensional Maxwell Solver based on a conditionally stable FDTD algorithm [20,28]: The electromagnetic field components are computed numerically by discretizing space and time and approximating the spatial and temporal derivatives in Maxwell’s curl equations by second-order central differences. More details about the algorithm can be found in Menzel et al. [17].

Simulation results

Matrix calculus (*simPLI*) The collision-free *Fiber Cup* phantom was used as input for *simPLI* (see “Simulation methods” section). The fibers were modeled as homogeneous materials with an absorption of $\mu = 5 \text{ mm}^{-1}$, and the birefringence was modeled with a strength of $\Delta n = 0.001$ and one axis

of anisotropy oriented along the longitudinal fiber axis [16]. For the simulation, a section of $60\ \mu\text{m}$ thickness from the mid-plane of the *Fiber Cup* phantom was used, discretized into voxels of $1\ \mu\text{m}$ side length. The simulations were performed with wavelength $\lambda = 525\ \text{nm}$, ingoing light intensity $I_0 = 26,000$, and image pixel size $\Delta x = 20\ \mu\text{m}$.

Figure 5a shows the resulting fiber orientations as a vector field, overlaid with the original fiber model. In regions with straight fiber bundles, the fiber orientations are in good agreement with the underlying fiber structure. In crossing regions, however, some fiber orientations are misinterpreted. In region (3), where both crossing fiber bundles contain the same amount of fibers, the resulting vector field shows the average orientation of the two underlying fiber bundles (pink). If fiber tractography, algorithms were applied to the vector field, and the crossing fibers (\times) could be misinterpreted as kissing fibers ($><$). In region (1), the resulting vector field is dominated by the orientation of the fiber bundle (green) that consists of three bundles (nos. 4, 5, and 6, see Fig. 4). The orientation of the other fiber bundles (purple) is not visible in the resulting vector field so that the crossing of the fiber bundles is not detected. In region (2), the three fiber bundles (4, 5, and 6) join each other. Bundles 4 and 5 (green) have similar orientations and are already indistinguishable when they merge with bundle 6 (red). The resulting orientation vectors of bundle 6 assimilate with the orientation vectors of the other two bundles. The original course of bundle 6 in the crossing regions is not visible in the resulting vector field.

Finite-Difference Time-Domain (FDTD) simulation The three selected regions of the *Fiber Cup* phantom in Fig. 5a were also studied with FDTD simulations to investigate the scattering of light. For this purpose, a volume of $128 \times 128 \times 60\ \mu\text{m}^3$ was selected in each region and the propagation of the light wave through the sample was simulated as described in the “Simulation methods” section. The simulations were performed for uniaxial perfectly matched layer absorbing boundaries with a thickness of $1\ \mu\text{m}$, a Yee mesh size of $25\ \text{nm}$, a duration of 400 periods, a Courant factor of 0.8, and normally incident light with $550\ \text{nm}$ wavelength (modeling the polarizing microscope) [18,19]. All tissue components of the sample (fibers and surrounding tissue) were modeled by dielectric materials with real refractive indices (neglecting absorption). Each fiber was modeled by an inner axon and a surrounding myelin sheath with two layers, as described by Menzel et al. [18]. The simulations were performed on the supercomputer *JURECA* [11] at Forschungszentrum Jülich GmbH, Germany. Figure 5b shows the resulting scattering patterns, i.e. the intensity per wave vector, for the three selected regions. The scattering patterns reveal the underlying fiber structure and crossing angles of the fiber bundles. The light is always scattered under angles perpendicular to

the principal axis of the corresponding fiber bundle. For reference, the perpendicular axes are indicated by straight colored lines around the scattering patterns. (Colors were chosen according to the colors of the fiber bundles shown in Fig. 5a.) In contrast to *simPLI*, the FDTD simulations allow to resolve individual fiber orientations also in regions with crossing fibers. The determined angles correspond to the angles of the *Fiber Cup* phantom (cf. Figs. 4 and 5b). Bundles with a larger amount of fibers (regions (1), (2), in green) cause stronger scattering than bundles with a smaller amount of fibers.

Discussion and conclusion

We have introduced the *FAConstructor* tool enabling the interactive generation of synthetic nerve fiber models both in an accurate and efficient way. The fast generation of many different fiber models is a prerequisite for imaging simulation tools to study physical effects in virtual brain tissue. Functionalities like modification of fiber objects (translation, rotation, scaling, filling), grouping of several fiber objects (e.g. fibers to a fiber bundle) and successive generation of building blocks (as groups) up to the final model allow a fast generation and easy (re)production of phantoms. By defining the radius as a function along the fibers, it is also possible to build multiple layers (e.g. myelin with nodes of Ranvier) around an axon.

Evaluating the performance of the developed tool shows that most processes scale almost linearly with the amount of data points used to describe the fiber models. When working with simple, unfilled objects, the program guarantees a smooth interaction proven by the performance measurements shown. Models with less than 1.6 million data points retain a frame rate of more than 30 frames per second. Assuming a constant diameter of nerve fibers in the human brain of $0.5\ \mu\text{m}$ [12] and, for a high detailed model, a step size of about $1\ \mu\text{m}$, the here used hardware could render a stable frame rate for a filled cubic volume of about $80\ \mu\text{m}^3$. It is therefore also capable of visualizing a high detailed model for the FDTD simulation. For building larger models, the usage of hiding groups can improve the rendering temporarily, where the final model can still be rendered when necessary.

As the 3D-PLI simulations only model sections of brain tissue with $60\ \mu\text{m}$ thickness, more than 1.6 million data points are rarely necessary, which guarantees a stable and fluent workflow. For comparison, the newly generated *Fiber Cup* phantom contains 426,300 data points after filling and the original fiber bundle contains only 290 data points. This indicates that even large fiber models can be rendered on similar hardware. However, to create even more detailed and/or larger brain tissue models beyond pure white

matter phantoms (e.g. cerebral cortex phantoms including astrocytes, glial cells, and neuropil), a supercomputing environment is required. First tests of the *FAConstructor* on one visualization node of the supercomputer *JURECA* [11] at the Supercomputing Center at Forschungszentrum Jülich GmbH, Germany, showed promising performance. This aspect will be addressed in future studies.

The applicability of *FAConstructor* was demonstrated for two different simulation approaches using the example of the *Fiber Cup* phantom. The generation of the phantom took around ten minutes for an instructed user, indicating the user-friendly environment of *FAConstructor*. Since the tool can be used to generate large-scale fiber models with small-scale detail, it is suited for both large-scale *simPLI* and small-scale FDTD simulations. It will help to generate the models quickly and, if necessary, to modify them accordingly so that the underlying fiber structure can be tested from experimental data in an appropriate time.

In order to enable faster and more intuitive changes to the models, mouse interactions with the scene need to be implemented. This would allow more intuitive modification of fiber objects, such as moving fiber points or changing entire objects or groups. Another improvement would be the visualization of the 3D-PLI data or more general gray-scale and RGB images, or three-dimensional vector data. By this means, it would be possible to visualize the generated structures with the experimental results and the resulting simulation data. Therefore, not only the construction process of the models would be improved, but also the interpretation and analysis process of the simulation results in relation to the models used. A future implementation could also provide an interface for plug-ins that would allow algorithms, analyses, or even the simulation process to be run. Since a model without colliding fibers is very important for 3D-PLI simulations, the colliding solving process could be executed and visualized step-by-step.

Due to the general definition of fibers as geometric objects defined by points and radii, other neuroimaging simulation techniques like dMRI are also capable of using the models generated by *FAConstructor*. Since the technological and analytical methods in dMRI improve continuously, highly detailed microstructured models are required for simulations to address the impact of underlying neuronal components on the diffusion behavior [9].

In conclusion, we have implemented a user-friendly and efficient tool that enables a fast generation of complex and realistic synthetic nerve fiber configurations (i.e. numerical brain tissue phantoms) ideally suited for the simulation of neuroimaging techniques like 3D polarized light imaging. The generated fiber bundles can be visualized simultaneously and edited interactively. This ultimately enables the production of application-specific well-defined ground truth datasets.

Acknowledgements This study was funded by the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement No. 785907 (HBP SGA2). We gratefully acknowledge the computing time granted through JARA-HPC on the supercomputer *JURECA* [11] at Forschungszentrum Jülich. This open source software code was developed in part or in whole in the Human Brain Project, funded from the European Union's Horizon 2020 Framework Program for Research and Innovation under the Specific Grant Agreement No. 785907 (HBP SGA2).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Axer M, Amunts K, Grassel D, Palm C, Dammers J, Axer H, Pietrzyk U, Zilles K (2011) A novel approach to the human connectome: ultra-high resolution mapping of fiber tracts in the brain. *NeuroImage* 54(2):1091–1101
2. Axer M, Grassel D, Kleiner M, Dammers J, Dickscheid T, Reckfort J, Hütz T, Eiben B, Pietrzyk U, Zilles K, Amunts K (2011b) High-resolution fiber tract reconstruction in the human brain by means of three-dimensional polarized light imaging. *Front Neuroinform* 5(34):1–13
3. Caspers S, Axer M (2017) Decoding the microstructural correlate of diffusion MRI. *NMR Biomed* 0(0):e3779
4. Company TQ (2018) About Qt. https://wiki.qt.io/About_Qt
5. Dai JS (2015) Euler-Rodrigues formula variations, quaternion conjugation and intrinsic connections. *Mech Mach Theory* 92:144–152
6. 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
7. Fillard P, Descoteaux M, Goh A, Gouttard S, Jeurissen B, Malcolm J, Ramirez-Manzanares A, Reisert M, Sakaie K, Tensaouti F, Yo T, Mangin JF, Poupon C (2011) Quantitative evaluation of 10 tractography algorithms on a realistic diffusion MR phantom. *NeuroImage* 56(1):220–234
8. Foley JD, Van Dam A, Van Dam A, Feiner SK, Hughes JF, ANGEL E (1996) *Computer graphics: principles and practice*. Addison-Wesley, Boston
9. Ginsburger K, Poupon F, Beaujourn J, Estournet D, Matuschke F, Mangin JF, Axer M, Poupon C (2018) Improving the realism of white matter numerical phantoms: a step toward a better understanding of the influence of structural disorders in diffusion MRI. *Front Phys* 6:12
10. Jeurissen B, Leemans A, Tournier JD, Jones DK, Sijbers J (2013) Investigating the prevalence of complex fiber configurations in white matter tissue with diffusion magnetic resonance imaging. *Hum Brain Mapp* 34(11):2747–66

11. Jülich Supercomputing Centre (2016) JURECA: General-purpose supercomputer at Jülich Supercomputing Centre. *J Large Scale Res Facil* 2:A62
12. Liewald D, Miller R, Logothetis N, Wagner HJ, Schüz A (2014) Distribution of axon diameters in cortical white matter: an electron-microscopic study on three human brains and a macaque. *Biol Cybern* 108(5):541–557
13. Magnain C, Augustinack JC, Tirrell L, Fogarty M, Frosch MP, Boas D, Fischl B, Rockland KS (2019) Colocalization of neurons in optical coherence microscopy and Nissl-stained histology in Brodmann's area 32 and area 21. *Brain Struct Funct* 224(1):351–362
14. Maier-Hein KH, Neher PF, Houde JC, Côté MA, Garyfallidis E, Zhong J, Chamberland M, Yeh FC, Lin YC, Ji Q, Reddick WE, Glass JO, Chen DQ, Feng Y, Gao C, Wu Y, Ma J, Renjie H, Li Q, Westin CF, Deslauriers-Gauthier S, González JOO, Paquette M, St-Jean S, Girard G, Rheault F, Sidhu J, Tax CMW, Guo F, Mesri HY, Dávid S, Froeling M, Heemskerk AM, Leemans A, Boré A, Pinsard B, Bedetti C, Desrosiers M, Brambati S, Doyon J, Sarica A, Vasta R, Cerasa A, Quattrone A, Yeatman J, Khan AR, Hodges W, Alexander S, Romascano D, Barakovic M, Auría A, Esteban O, Lemkaddem A, Thiran JP, Cetingul HE, Odry BL, Mailhe B, Nadar MS, Pizzagalli F, Prasad G, Villalon-Reina JE, Galvis J, Thompson PM, Requejo FDS, Laguna PL, Lacerda LM, Barrett R, Dell'Acqua F, Catani M, Petit L, Caruyer E, Daducci A, Dyrbj TB, Holland-Letz T, Hilgetag CC, Stieltjes B, Descoteaux M (2017) The challenge of mapping the human connectome based on diffusion tractography. *Nat Commun* 8(1):1349
15. Matuschke F, Ginsburger K, Poupon C, Amunts K, Axer M (2019) Dense fiber modeling for 3D-Polarized Light Imaging simulations. [arXiv:1901.10284](https://arxiv.org/abs/1901.10284)
16. Menzel M, Michielsen K, De Raedt H, Reckfort J, Amunts K, Axer M (2015) A Jones matrix formalism for simulating three-dimensional polarized light imaging of brain tissue. *J R Soc Interface* 12:20150734
17. Menzel M, Axer M, De Raedt H, Michielsen K (2016) Finite-difference time-domain simulation for three-dimensional polarized light imaging. In: Amunts K, Grandinetti L, Lippert T, Petkov N (eds) *Brain-inspired computing*. Springer, Cham, pp 73–85
18. Menzel M, Axer M, De Raedt H, Costantini I, Silvestri L, Pavone FS, Amunts K, Michielsen K (2019) Finite-difference time-domain simulations of transmission microscopy enable a better interpretation of 3D nerve fiber architectures in the brain. [arXiv:1806.07157](https://arxiv.org/abs/1806.07157)
19. Menzel M, Axer M, Amunts K, De Raedt H, Michielsen K (2019) Diattenuation Imaging reveals different brain tissue properties. *Sci Rep* 9:1939
20. Michielsen K, De Raedt H, Stavenga DG (2010) Reflectivity of the gyroid biophotonic crystals in the ventral wing scales of the Green Hairstreak butterfly, *Callophrys rubi*. *J R Soc Interface* 7(46):765–771
21. Müller H (1943) Memorandum on the polarization optics of the photoelastic shutter. Report No 2 of the OSRD project OEMsr-576
22. OpenMP Architecture Review Board (2015) OpenMP application program interface version 3.0. <https://www.openmp.org/wp-content/uploads/openmp-4.5.pdf>
23. Schmitz D, Muenzing SEA, Schober M, Schubert N, Minnerop M, Lippert T, Amunts K, Axer M (2018) Derivation of fiber orientations from oblique views through human brain sections in 3D-polarized light imaging. *Front Neuroanat* 12:75
24. Schubert N, Axer M, Pietrzyk U, Amunts K (2018) 3D Polarized Light Imaging portrayed: visualization of fiber architecture derived from 3D-PLI. In: Halefoğlu AM (ed) *High-resolution neuroimaging—basic physical principles and clinical applications*, InTech, Rijeka, chap 03
25. Shreiner D, Sellers G, Kessenich J, Licea-Kane B (2013) *OpenGL programming guide: The Official guide to learning OpenGL, version 4.3*. Addison-Wesley, Boston
26. Silvestri L, Costantini I, Sacconi L, Pavone FS (2016) Clearing of fixed tissue: a review from a microscopist's perspective. *J Biomed Opt* 21(8):081205
27. Wang H, Akkin T, Magnain C, Wang R, Dubb J, Kostis WJ, Yaseen MA, Cramer A, Sakadžić S, Boas D (2016) Polarization sensitive optical coherence microscopy for brain imaging. *Opt Lett* 41(10):2213–2216. <https://doi.org/10.1364/OL.41.002213>, <http://ol.osa.org/abstract.cfm?URI=ol-41-10-2213>
28. Wilts BD, Michielsen K, De Raedt H, Stavenga DG (2014) Sparkling feather reflections of a bird-of-paradise explained by finite-difference time-domain modeling. *PNAS* 111(12):4363–4368
29. Wu W, Miller KL (2017) Image formation in diffusion MRI: a review of recent technical developments. *J Magn Reson Imaging* 46(3):646–662
30. Zilles K, Palomero-Gallagher N, Gräbel D, Schlömer P, Cremer M, Woods R, Amunts K, Axer M (2015) High-resolution fiber and fiber tract imaging using polarized light microscopy in the human, monkey, rat, and mouse brain. Elsevier, London, pp 369–389

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.