OpenWalnut
Multi-modal Visualization of the Human Brain

S. Eichelbaum, M. Goldau, S. Philips,
A. Reichenbach, R. Schurade and A. Wiebel

Bild- und Signalverarbeitung
UNIVERSITÄT LEIPZIG

Max-Planck-Institut
Kognitions- und Neurowissenschaften

Abstract

In the course of the ongoing research into neurological diseases and the function and anatomy of the brain, a large variety of examination techniques has evolved. The different techniques aim at findings for different research questions or different viewpoints of a single task. Considering the different applications, it is evident that, for many research areas, only a combination of mutliple techniques can help answering the posed questions. To name only one example, the combination of dwMRI and fMRI with an anatomical context provided by T1 MRI images is very common. To be able to analyze the data measured by the different techniques, a tool that can efficiently visualize the different modalities simultaneously is needed. The software (called *OpenWalnut*) we will present in this poster aims at exactly this task. It does not only allow to display the different modalities together, but also provides tools to analyze their interdependence and relations.
In the following, we will provide you with an overview of OpenWalnut's features. Its simplicity, making it a useful visualization tool, on the one hand and its powerful and generic framework for computer scientist researchers on the other hand.

OpenWalnut - the easy-to-use Visualization Tool

One of the main development requirements of OpenWalnut was to provide an easy and intuitive graphical user interface (GUI). We achieved this by mapping complex operations, like combining data and algorithms, to simple GUI operations. Data and algorithms get combined by using a tree representation of it. This way, the user can apply a module to the results of other modules and datasets simply by clicking on it and selecting the wanted algorithm. Aside from this, the overall appearance of the GUI has been kept clean. No cluttered configuration dialogs, no confusing tabs or panels, just a clean sidebar providing the user all the tools he needs to get the visualization job done. The whole algorithm- and dataset- configuration is available through this panel (called *Datasetbrowser*). Figure 1 and 2 show the customizable GUI. Each element in the GUI can be de- or activated separately. This clean and structured user interface makes it also suitable for scientists who simply want to **use** visualization tools for their data but are not that familiar with the visualization details.
As OpenWalnut is developed in cooperation with neuroscientist researchers at the Max-Planck-Institut for Cognitive- and Brain Sciences in Leipzig, it already contains an extensive collection of algorithms applicable to many kinds of data, like scalar data, vector datasets, DTI datasets and even tensor data of higher order. And the algorithm collection is constantly growing. State-of-the-art as well as standard techniques get added all along during research use.

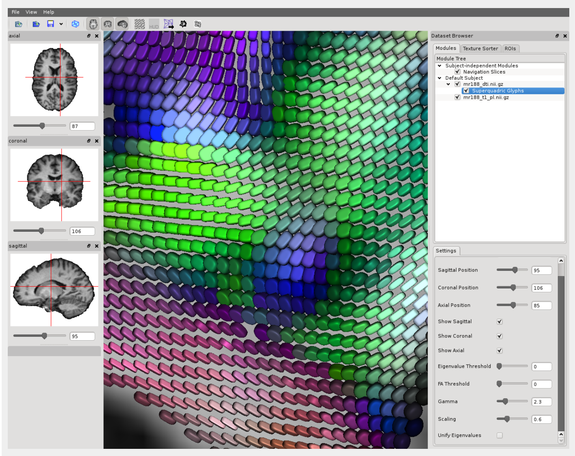

Figure 1: Superquadric Tensor Glyphs [Kin04] rendered with OpenWalnut. The simple and intuitive GUI allows simple modification of glyph parameters.

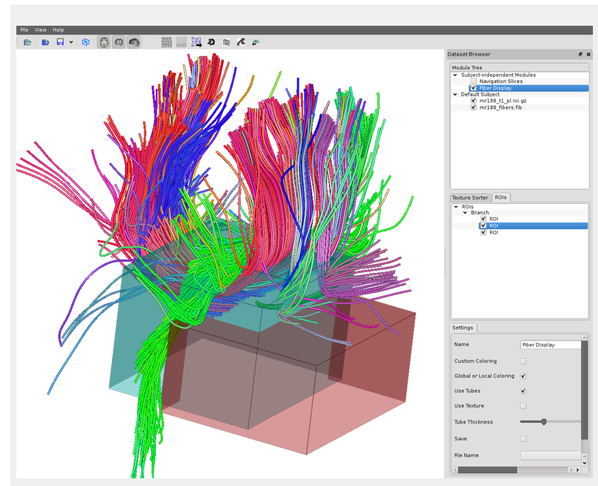

Figure 2: In a pre-computed deterministic fiber tracking, regular regions of interest (ROI) have been used to select a part of the corpus callosum and the cingulum.

OpenWalnut is worth a try. It is a lively open source project with an active developer community and is developed to fulfill the needs of users and developers, of neuroscientists and computer scientists and surely yours too. Have a look at it on http://www.openwalnut.org.

OpenWalnut - the framework

To achieve the challenging goal of combining multiple modalities and multiple approaches of analysis and visualization, OpenWalnut uses the idea of data flow networks internally to combine several algorithms and provide a processing flow. Each algorithm in OpenWalnut is called "Module" and can provide inputs and outputs, so called "connectors". These connectors are typed by the kind of data they can receive or transmit. This way, each module can define the kind of data it wants to process or provide to others, which somehow correlates to the Unix philosophy "One program for one purpose", as each module resides in its own small world without knowledge about other modules or the module graph (cf. Figure 3). Modules get connected to each other under respect of the type of connectors they provide and as every module runs in a separate thread, changes in an output-connector of one module get propagated to all connected modules, which wake up from their sleep state and process the new data in parallel.
But OpenWalnut is not only a data flow network. From the developers point of view, it is a sophisticated and feature rich framework for programming new algorithms, processing tools and visualization techniques. It provides programming interfaces to handle user input comfortably using an abstract interface, to handle in- and output of data through connectors, to work and process data in multiple threads or to create realtime three-dimensional graphics. Although, the framework is designed in a clean and strict fashion, it does not constrain the developer in his possibilities. The framework is simply designed to make the module programmer's life easier; to allow the programmer of a new visualization or processing technique to concentrate on the technique itself.

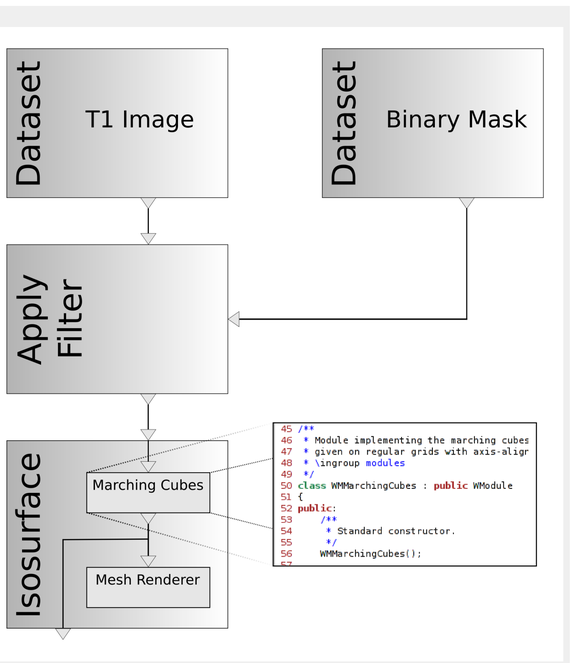

Figure 3: Example module graph showing the workflow of masking a scalar dataset and rendering an iso-surface of it.

Despite its importance, documentation is often underestimated in many software projects. Not in OpenWalnut. We strictly enforce documentation in code, with our ticket system and for user documentation in our wiki.

OpenWalnut - powerful, extensible, generic and multi-modal

OpenWalnut is able to load a multitude of different image modalities in a variety of standard file formats (e.g. NIfTI, .fib and Vista). For all loadable modalities, a default module is available to provide a simple standard visualization of it.

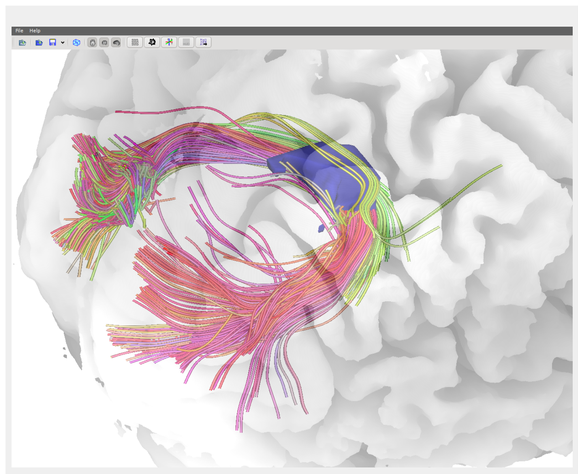

Figure 4: Forceps occipitalis selected by a segmented part (blue) of the corresponding T1-weighted MRI image, which is used as a region of interest.

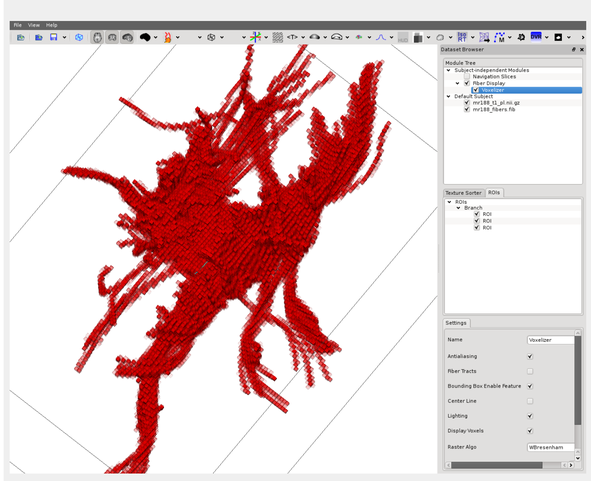

Figure 5: The selection from Figure 2 voxelized by applying the voxelizer module to the fiber display module.

Although, the standard visualizations for all modalities are useful, they do not always yield further insight into the modalities' relation to each other. OpenWalnut's internal module-graph-structure allows data of different modalities to be combined in a very simple and uncomplicated way. For modules, it is equally simple to provide multiple kinds of data to other modules, such as intermediate calculation results, helping other algorithms to reuse them in their visualization. An example for this is the uncertainty field occurring in a fiber tracking algorithm, which can be used by other modules to take it into account during their calculations.
The outputs of the modules can furthermore be used to define regions of interest, which select parts of other datasets (cf. Figure 4). The possibilities of how modules can be combined to create new visualizations are nearly endless. Unlike MeVisLab [mev] or SCIRun [SCI], OpenWalnut provides these endless possibilities through an easy-to-use and clean user interface and programming framework.

Features in a Nutshell

In this section, we summarize the key-features of OpenWalnut to give you a fast overview.

- Designed to be interactive
- Easy-to-use GUI
- Advanced features in GUI for advanced users
- Endless possibilities for combining and re-using modules
- Powerful framework
  - Easy-to-use for module programmer
  - Clean interface
  - Not restrictive (very generic)
  - Documented!
- Multi-modal
- Makes extensive use of multi-core architectures
- Already contains many standard visualization methods
- Supports many kinds of data and data-formats and can be extended easily
- Open-source
- Very active development community
- Used by neuroscientists, ensuring real user feedback.

Conclusion

In the last sections, we introduced OpenWalnut from three different perspectives; the user-oriented, interactive visualization tool, the powerful framework and the generic, extensible and multi-modal tool for advanced users. Its universality allows it to be easily extended and used in any kind of application case. It is both, a tool for the scientific user and a powerful framework for the visualization researcher. OpenWalnut is licensed under the terms of the GNU Lesser General Public License. More information, documentation and downloads can be found on the project website http://www.openwalnut.org.

Acknowledgments The authors are grateful for the advice, help and ideas of Gerik Scheuermann, Faisal Cheema, Christian Heine, Alfred Anwander, Thomas Knösche, Frank Zanow, Robin Ledig, Cornelius Müller and Mario Hlawitschka. This work was supported by the AIF (ZIM grant KF 2034701SS8).

References

[Kin04] Gordon Kindlmann. Superquadric tensor glyphs. In *Proceedings of IEEE TCVG/EG Symposium on Visualization 2004*, pages 147–154, May 2004.
[mev] MeVisLab - development environment for medical image processing and visualization. http://www.mevislab.de/.
[SCI] SCIRun: A scientific computing problem solving environment, scientific computing and imaging institute (SCI). http://www.scirun.org.