



HAL
open science

MedINRIA: Medical Image Navigation and Research Tool by INRIA

Nicolas Toussaint, Jean-Christophe Souplet, Pierre Fillard

► **To cite this version:**

Nicolas Toussaint, Jean-Christophe Souplet, Pierre Fillard. MedINRIA: Medical Image Navigation and Research Tool by INRIA. Proc. of MICCAI'07 Workshop on Interaction in medical image analysis and visualization, 2007, Brisbane, Australia, Australia. inria-00616047

HAL Id: inria-00616047

<https://inria.hal.science/inria-00616047v1>

Submitted on 7 Jan 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MedINRIA*: Medical Image Navigation and Research Tool by INRIA

Nicolas Toussaint¹, Jean-Christophe Souplet¹, and Pierre Fillard¹

Asclepios Research Team, INRIA Sophia Antipolis, France.
{Nicolas.Toussaint, Jean-Christophe.Souplet,
Pierre.Fillard}@sophia.inria.fr

Abstract. Processing and visualization of 3D medical data is nowadays a common problem. However, it remains challenging because the diversification and complexification of the available sources of information, as well as the specific requirements of clinicians, make it difficult to solve in a computer science point of view. Indeed, clinicians need ergonomic, efficient, intuitive and reactive softwares. Moreover, they need new solutions to fully exploit their data, but they often cannot access state-of-the-art methods as those are mostly available in complicated softwares. The MedINRIA software was born to fill this lack and consists of a collection of tools that optimally exploit various types of data (e.g., 3D images, diffusion tensor fields, neural fibers as obtained in DT-MRI). It provides state-of-the-art algorithms while keeping a user-friendly graphical interface. For each of these tools, we first introduce its dedicated application and the processing methods it contains. Then, we focus on the features that make interactions with data even more intuitive. MedINRIA is a free software, available on Windows, Linux and MacOSX. Other MedINRIA tools are underway to make cutting edge research in medical imaging rapidly available to clinicians. The interest clinicians have shown in MedINRIA so far indicates that the need of such simple, yet powerful softwares is real and increasing.

1 Introduction

The increase and diversification of sources of information in medical imaging has raised the need for adapted tools for visualization and analysis of these data. For instance, the progress made in magnetic resonance imaging (MRI) for the past two decades has prodigiously improved the quality and resolution of 3D images. Moreover, these images can take various forms: anatomical MRI, functional MRI (fMRI) and diffusion tensor MRI (DT-MRI or DTI) [1] are only a few examples. The analysis of these data is challenging too. For instance, computing neural fibers in DTI or segmenting lesions in multiple sclerosis (MS) require adapted and sophisticated methods.

Most of state-of-the-art algorithms are not easily accessible to clinicians. These methods are, however, mostly available in relatively complex softwares,

* Free download at: <http://www-sop.inria.fr/asclepios/software/MedINRIA>

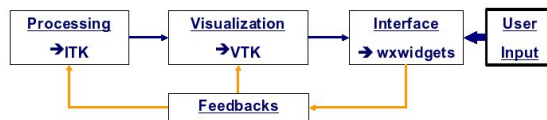


Fig. 1. The MedINRIA architecture. MedINRIA is composed of three main blocks: 1. The processing block based upon ITK, 2. The visualization block using VTK and 3. The user interface block made in wxWidgets. The blue arrows indicate the data flow, from the processing block to the user interface. The user-interface can only send controls to the processing and visualization blocks (the opposite is forbidden). Thus, the interface can be re-written using another library with minimum effort.

like Slicer [2] or BrainVisa [3]. Although these softwares are extremely powerful, they can be quite hard to fully master. Clinicians needs are simple yet very difficult to satisfy in a computer science point of view: they want ergonomic, reactive and intuitive softwares that offer real-time interactions with data. The MedINRIA project was born to fulfill this need.

MedINRIA is a collection of softwares targeting the clinicians. It has the native look&feel on any system (Windows, MacOSX and Linux), and effort has been made to simplify the user interface while keeping state-of-the-art algorithms. Up to now, MedINRIA contains softwares for image visualization, DT-MRI analysis, tensor visualization, and semi-automatic segmentation of MS lesions. The paper is organized as follows: In Sec. 2, some basics and concepts of MedINRIA are introduced, and a description of each application is given, from their general purpose to the analysis and interactions with medical data they introduce. We conclude in Sec. 3 with a discussion on the future work.

2 The MedINRIA Software

Schematically, MedINRIA is composed of three main blocks, as shown in Fig. 1. First, the processing block is based upon ITK [4]. It has all ITK features: flexibility (it can process any type of data), multithreaded and a pipelined architecture. Second, the visualization block made with VTK [5], itself based on OpenGL, offers the same pipelined structure as ITK. Finally comes the user interface block written in wxWidgets [6]. This block is responsible for transmitting user-defined parameters to the two first blocks. Unlike the processing and visualization blocks, this one can only transmit data in one direction, and consequently cannot receive any information from the other blocks. The main reason is to keep the user-interface detached from the processing and visualization part. Thus, one could easily re-write the user-interface with another library, or even write command line tools instead of graphical applications.

When started, all MedINRIA's tools can be launched from a single menu. Up to now, five applications are available, each of them being dedicated to a specific task or type of data. We give below a description of each of them: we start by

briefly introducing the general purpose, and continue by focusing more on the user-interactions they propose to facilitate the intuitive exploration of data.

2.1 ImageViewer: A Simple yet Powerful Image Viewer

This first application aims at displaying 3D images and interacting with them. It is also used to convert clinical data (DICOM) into volumetric images. The *ImageViewer* introduces interesting new features like: image tabbed browsing, a preview screen (Fig. 2 left) where all images can be compared (interactions are synchronized: slice or contrast change of one window affects all the other windows as well), which gives an ergonomic navigation through images. In the following, we will only focus on the 3D interactions with images, as this is the most innovative part w.r.t. the simplification of data manipulation.

3D Image Visualization: Images are displayed in 3D either using multiplanar reconstruction (MPR) (Fig. 2 left) or volume rendering (VR) (Fig. 2 middle). MPR consists in displaying a slice in every orthogonal direction, while VR uses 3D textures with an opacity transfer function and is hardware-accelerated. The opacity transfer function for the VR is controlled by the contrast of the 2D views. The transfer function that controls opacity in VR is given by two parameters, *window* and *level*: any voxel whose scalar value is less than $level - window/2$ will be transparent, and any value above $level + window/2$ will be opaque. The ramp in-between is linear.

Volume Of Interest (VOI) extraction: Manual extraction of a VOI in VR is a desirable feature. Clinicians may need to have an insight into the patient's brain and remove only one block from his head by visually clicking on the screen. This is done in MedINRIA using a *cropping box*: it consists of a white box that can be scaled and translated by clicking on white handles present on each face (Fig. 2 right). The volume inside the box is automatically removed and cropping is done in real-time (hardware-accelerated).

2.2 DTI Track: Log-Euclidean DT-MRI Processing

The *DTI Track* application is dedicated to DT-MRI processing and visualization. This imaging modality measures the anisotropy of water molecules diffusion within the brain [1]. From several MRI measures (at least 6), one is able to reconstruct a diffusion tensor at each voxel. This tensor, represented by a 3D ellipsoid, gives an insight on the directionality of the underlying tissue. If water molecules are free to move in every direction of space, the diffusion tensor will be a sphere. By contrast, if the diffusion is restricted by structures (myelinated axons of the brain white matter), the tensor will be cigar-shaped and aligned along the oriented tissues. This is a unique modality to assess in vivo the directionality of brain white matter, and can lead eventually to the reconstruction of nervous fibers via a process called tractography [7].

DTI Track guides the user through the whole process, from importing the measures to the reconstruction of nervous fibers. The processing pipeline consists of three steps. First, the diffusion tensor field is estimated from the raw data: this

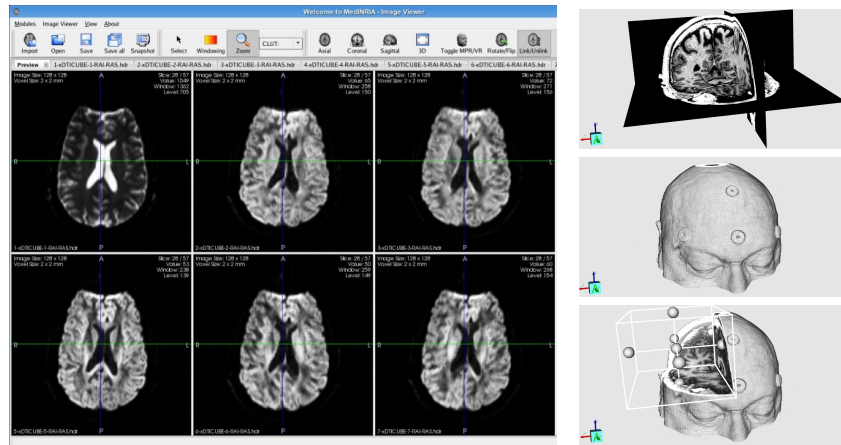


Fig. 2. Image visualization with MedINRIA. **Left:** The preview screen. **Right:** Image rendering in 3D. **Top:** Multiplanar reconstruction (MPR) of a T1 image. An axial, sagittal and coronal slices is displayed. **Middle:** Volume rendering (VR) of the same image. **Bottom:** The cropping box was used to extract a VOI. By clicking on the handles, it can be resized (face handles) and translated (center handle).

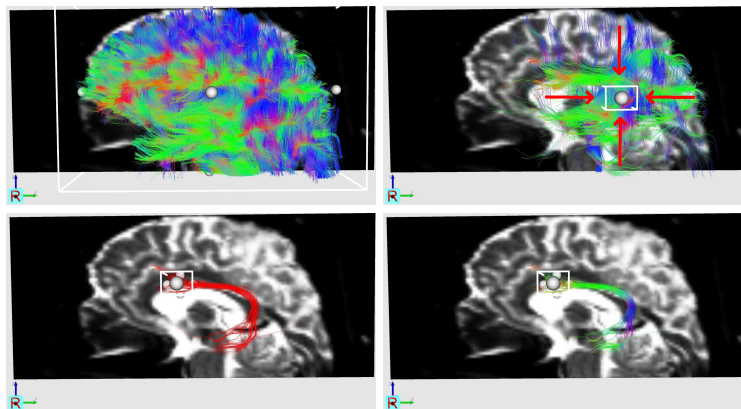


Fig. 3. Example of fiber bundling with MedINRIA. **Top left:** The complete set of reconstructed fibers. The white box displayed is the *cropping box*. Only fibers that go through that box are shown. **Top right:** The box was shrunk by clicking on the face handles. The red arrows show the direction of the shrinking. The user can choose to navigate only in this sub-sample of fibers. **Bottom right:** The box was moved to another position, thus defining a second VOI. The bundle of interest (BOI) is ready to be tagged (i.e. labeled and saved). **Bottom left:** The BOI is now tagged and colored in red. It can be saved for further analysis.

generates an image of diffusion tensors that will be used for fiber reconstruction. Second, the diffusion tensor is smoothed using Log-Euclidean metrics [8]: these

metrics have proved to be very efficient practically and are among state-of-the-art methods for tensor processing [9]. In third comes the tractography. This last step generates thousands of nervous fibers that requires specific visualization and interaction techniques. Special attention was taken to make these two last steps as fast as possible, not to lower the overall reactivity of the software. This includes a multithreaded implementation with optimized numerical libraries. In the following, we will only focus on the manipulation and analysis of these fibers.

Fiber Manipulation: Depending on the image resolution, tractography can generate up to 100 000 fibers with an average of 100 points each. The task is not easy: one should display a maximum of information while keeping a real-time rendering. We chose to: 1. display fibers using only lines and not more complex geometric primitive like tubes, and 2. introduce a *cropping box* that intrinsically defines a VOI and only fibers that go through it will be displayed (Fig. 3). The cropping box can be scaled and translated with handles on each of its sides. Moving and scaling the box are real-time actions, giving clinicians a good insight of the brain connectivity.

Interactive Fiber Bundling: Navigating randomly into the reconstructed neural fibers may not be sufficient. Experts might be willing to identify and label a specific fiber bundle of interest (BOI). For this purpose, we offer two solutions. The first one is to manually define a region of interest (ROI) slice by slice on a 3D image. When this image is the patient anatomy (T1), critical structures can be easily spotted and segmented. Accessing the exact bundle that goes through these structures can be of interest. The second method relies on the cropping box. It can be used recursively, i.e., multiple VOIs can be defined by positioning it at various locations. Thus, one can easily isolate a single fiber using this method. Note that both methods can be used alternatively. Once the user is satisfied with the BOI, he can label it, color it and save it for further analysis. Quantitative values of the bundle can be computed: apparent diffusion coefficient, fractional anisotropy [10], etc.

2.3 TensorViewer: Tensor Fields Visualization

Visualizing diffusion tensor fields produced by *DTI Track* is important for quality-control. In DTI, it is frequent that tensor fields are flipped in the X, Y or Z direction. This is caused by a misalignment of the acquisition parameters with the patient acquisition frame. *TensorViewer* allows to visually control and correct for flipped tensors (Fig. 4 bottom). In *TensorViewer*, tensor fields are considered as 3D images, i.e. three slices in three orthogonal directions (namely axial, coronal and sagittal) are displayed (Fig. 4 top). However, one may question whether using ellipsoids for representing tensors is the best choice in terms of speed of refresh rate.

Which tensor representation to choose? The choice of the geometric primitive to represent a tensor is an active area of research [11]. As we said earlier, a tensor is fully represented by an ellipsoid: it can be divided into three eigenvectors (axis of the ellipsoid) and eigenvalues (axis magnitude). However, ellipsoids

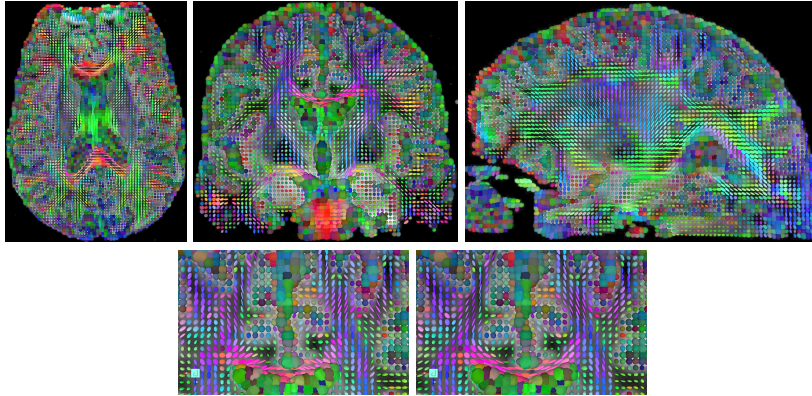


Fig. 4. Tensor visualization with MedINRIA. Top: An axial (left), coronal (middle) and sagittal (right) slice of a 3D tensor field. **Bottom:** The tensor field (left) looks flipped left-right. We applied a flipping (right) to correct this defect.

require lots of polygons to be rendered nicely (about 64 polygons). For instance, a dense field of $128 \times 128 \times 60$ voxels (typical resolution of DTI) gives a total of 2 millions polygons to display three orthogonal slices, by far too many for a real-time rendering on a regular computer. To fasten the rendering, we give the possibility to downsample the field and lower the number of polygons. Moreover, we propose to represent a tensor not only by an ellipsoid, but by any other geometric primitives that can correctly represent the shape (from a sphere to a cigar-shaped ellipsoid) with fewer polygons. For instance, a cube, which has only 6 faces, gives a good approximation of the true tensor shape (the cube is rotated and scaled w.r.t. the eigenvectors/eigenvalues). To go one step further, if we get rid of the smallest eigenvalue (which is meaningless most of the time in DTI), planar shapes are possible too: a disk or a square are potential primitives.

Eventually, one can simply represent the principal eigenvector of a tensor instead of the full tensor itself: in DTI, it is admitted that the main eigenvector is aligned with the underlying oriented structure under specific conditions. These conditions are when the fractional anisotropy (FA) [10] is high. FA tells how far a tensor is from the sphere: 0 means that the tensor is a sphere (absence of structure), 1 means that a tensor is cigar-shaped (presence of structure). In that case, a simple line is an acceptable shape for displaying tensors. MedINRIA proposes a listbox where all geometric shapes discussed above can be selected.

In addition to the shape chosen, we color code a tensor by its main eigenvector (absolute values of the coordinates are considered as RGB): red is for a left-right oriented tensor, green is for antero-posterior and blue is for infero-superior. The color magnitude is controlled by the FA: a dark tensor means absence of fibers. Conversely, a bright tensor means presence of fibers, and the color indicates the principal direction of the structure, according to the color code above. This provides a fast and accurate insight of the locations and directions of brain fibers.

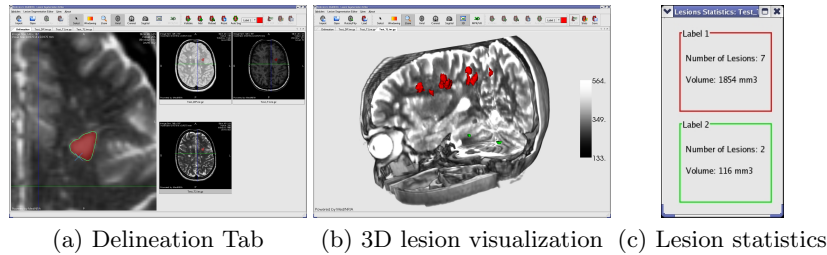


Fig. 5. LSE Interaction, Visualization and Results

2.4 LesionSegmentationEditor: MS Lesion Segmentation Tool

Lesion segmentation is a mandatory step in Multiple Sclerosis (MS) research. However, a precise manual segmentation is a fastidious task and is subject to inter- and intra-expert variability. *LesionSegmentationEditor* (LSE) aims to ease the segmentation of lesions on conventional brain MRI by experts and to compute meaningful statistics on segmentations (number of lesions, volume).

A lesion can have a different pattern and contrast depending on the MRI modality (T1, T2, Proton Density), and is evolving between two exam time-points. Consequently, segmentation of lesions is frequently done on several images. To cover this point, the user can load several images in LSE. Each image can be visualized in a tab containing three 2D views and one 3D view. In these views, interactions are similar to *ImageViewer* (Sec. 2.1). Moreover, a "segmentation mask" is associated to each image: this is a binary image where a voxel is set to 1 in a presence of a lesion. This allows to overlap segmentations on images.

Practically, the segmentation is done on a window called the "Delineation Tab" (Figure 5 (a)): on the right is a 2D preview for each image (e.g. T1, T2, Proton Density). These views are synchronized to show the same part of the brain at any time. The bigger 2D view on the left is used for the actual segmentation. The image displayed can be either one of the images on the right. Lesions can be segmented either manually or automatically. In the manual mode, the user segments a lesion slice by slice with the mouse. In the automatic mode, the user only needs to spot the lesion and click inside it, and the segmentation is achieved automatically. For each lesion, a unique color can be set.

After segmenting a lesion on a specific image, the segmentation can be "transmitted" to the other images. This allows to segment a lesion on the modality where it has the best contrast and to see the result on any image. Similarly, a segmentation can be added or removed from any previous segmentation. Obtained segmentation can be visualized in 2D and 3D (Figure 5 (b)). They can be saved and statistics can be computed (Figure 5 (c)).

3 Conclusion

The software MedINRIA presented in this paper is a collection of applications dedicated to medical image processing and visualization. The goal was to pro-

vide to clinicians an intuitive, reactive and powerful processing and visualization system. For instance, MedINRIA proposes to extract a 3D volume of interest of an image interactively in real-time using a 3D box and volume rendering techniques. In addition, it introduces an interactive method to extract of a neural fiber bundle of interest among thousands of fibers, using the same type of 3D box. MedINRIA does not only to offer user-interactions for manipulating 3D data, but it also provides state-of-the-art algorithms that are most of the time not accessible to clinicians, as these algorithms are generally available in more research software. Here, we propose to process DT-MRI using Log-Euclidean metrics, to register images using a recent diffeomorphic implementation of the demons algorithm, to automatically segment lesions in MS, and to visualize tensors using a panel of geometric shapes. The choice of the methods and their implementation was driven by the constraint of efficiency.

Future work includes the development of new applications: we are currently working on a registration tool that provides a full set of automatic image registration methods, a tumor growth simulation software, and cardiac image processing. MedINRIA is growing fast, and will remain free. A Windows, Linux and MacOSX version can be downloaded at:
<http://www-sop.inria.fr/asclepios/software/MedINRIA>.

References

1. Basser, P., Mattiello, J., Bihan, D.L.: MR diffusion tensor spectroscopy and imaging. *Biophysical Journal* **66** (1994) 259–267
2. Pieper, S., Halle, M., Kikinis, R.: 3d slicer. In: Proc. of ISBI'04. (2004)
3. Cointepas, Y., Mangin, J.F., Garnero, L., Poline, J.B., Benali, H.: Brainvisa: Software platform for visualization and analysis of multi-modality brain data. *NeuroImage* **13**(6) (2001)
4. Ibanez, L., Schroeder, W., Ng, L., Cates, J.: The ITK Software Guide. Kitware, Inc. ISBN 1-930934-10-6, <http://www.itk.org/ItkSoftwareGuide.pdf>. First edn. (2003)
5. : The Visualization ToolKit: <http://www.vtk.org/> (2007)
6. : wxWidgets: <http://www.wxwidgets.org/> (2007)
7. Weinstein, D.M., Kindlmann, G.L., Lundberg, E.C.: Tensorlines: Advection-diffusion based propagation through diffusion tensor fields. *vis* **00** (1999) 40
8. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine* **56**(2) (2006) 411–421
9. Fillard, P., Arsigny, V., Pennec, X., Ayache, N.: Clinical DT-MRI estimation, smoothing and fiber tracking with Log-Euclidean metrics. *IEEE Transactions on Medical Imaging* (2007) In Press.
10. Westin, C.F., Maier, S., Mamata, H., Nabavi, A., Jolesz, F., Kikinis, R.: Processing and visualization for diffusion tensor MRI. *Medical Image Analysis* **6**(2) (2002) 93–108
11. Kindlmann, G.: Superquadric tensor glyphs. In: IEEE TCVG Symposium on Visualization. (2004) 147–154