MITK Diffusion

2014.10.00

Generated by Doxygen 1.8.8

Mon Dec 1 2014 12:12:38

Contents

Part I

General information

Chapter 1

The MITK User Manual

Welcome to the basic MITK user manual. This document tries to give a concise overview of the basic functions of MITK and be an comprehensible guide on using them.

1.1 About MITK

MITK is an open-source framework that was originally developed as a common framework for Ph.D. students in the Division of Medical and Biological Informatics (MBI) at the German Cancer Research Center. MITK aims at supporting the development of leading-edge medical imaging software with a high degree of interaction.

MITK re-uses virtually anything from VTK and ITK. Thus, it is not at all a competitor to VTK or ITK, but an extension, which tries to ease the combination of both and to add features not supported by VTK or ITK.

Research institutes, medical professionals and companies alike can use MITK as a basic framework for their research and even commercial (thorough code research needed) software due to the BSD-like software license.

Research institutes will profit from the high level of integration of ITK and VTK enhanced with data management, advanced visualization and interaction functionality in a single framework that is supported by a wide variety of researchers and developers. You will not have to reinvent the wheel over and over and can concentrate on your work.

Medical Professionals will profit from MITK and the MITK applications by using its basic functionalities for research projects. But nonetheless they will be better off, unless they are programmers themselves, to cooperate with a research institute developing with MITK to get the functionalitiy they need. MITK and the MITK applications are not certified medical products and may be used in a research setting only. They must not be used in patient care.

1.2 The User Interface

The layout of the MITK applications is designed to give a clear distinction between the different work areas. The following figure gives an overview of the main sections of the user interface.

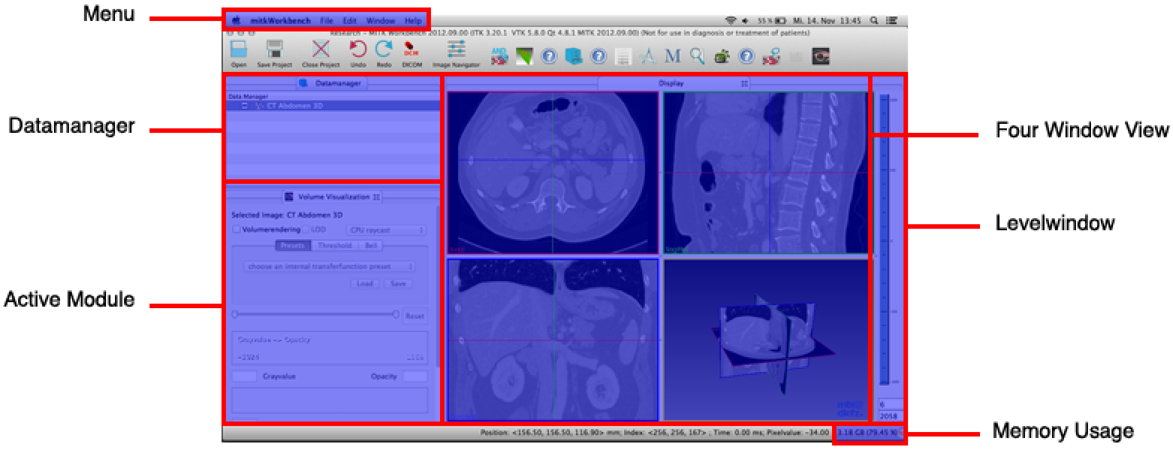


Figure 1.1: The Common MITK Application Graphical User Interface

The datamanager and the Perspectives have their own help sections. This document explains the use of:

- The Four Window View
- The Menu
- The Levelwindow
- The System Load Indicator
- The Views

1.3 Four Window View

1.3.1 Overview

The four window view is the heart of the MITK image viewing. The standard layout is three 2D windows and one 3D window, with the axial window in the top left quarter, the sagittal window in the top right quarter, the coronal window in the lower left quarter and the 3D window in the lower right quarter. The different planes form a crosshair that can be seen in the 3D window.

Once you select a point within the picture, informations about it are displayed at the bottom of the screen.

1.3.2 Navigation

Left click in any of the 2D windows centers the crosshair on that point. Pressing the right mouse button and moving the mouse **zooms** in and out. By scrolling with the mouse wheel you can **navigate through** the slices of the active window and pressing the mouse wheel while moving the mouse **pans** the image section.

In the 3D window you can **rotate** the object by pressing the left mouse button and moving the mouse, **zoom** either with the right mouse button as in 2D or with the mouse wheel, and **pan** the object by moving the mouse while the mouse wheel is pressed. Placing the cursor within the 3D window and holding the "F" key allows **free flight** into the 3D view.

1.3.3 Customizing

By moving the cursor to the upper right corner of any window you can activate the window menu. It consists of three buttons.

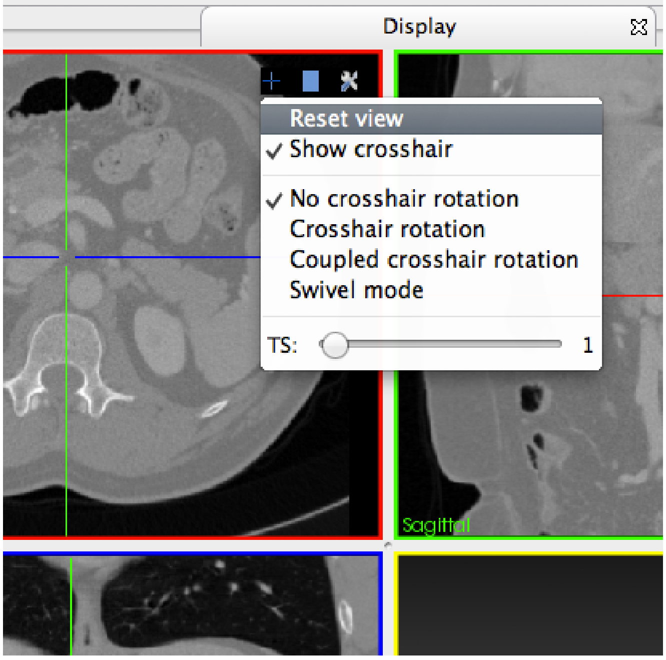


Figure 1.2: Crosshair

The crosshair button allows you toggle the crosshair, reset the view and change the behaviour of the planes.

Activating either of the rotation modes allows you to rotate the planes visible in a 2D window by moving the mouse cursor close to them and click and dragging once it changes to indicate that rotation can be done.

The swivel mode is recommended only for advanced users as the planes can be moved freely by clicking and dragging anywhere within a 2D window.

The middle button expands the corresponding window to fullscreen within the four window view.

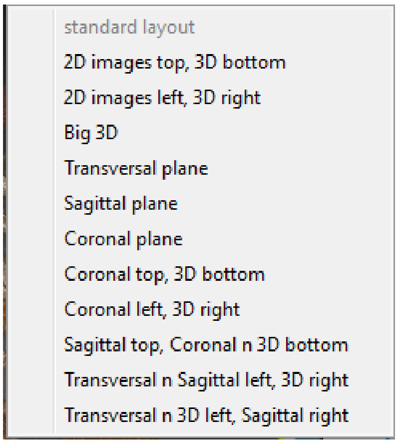


Figure 1.3: Layout Choices

The right button allows you to choose between many different layouts of the four window view to use the one most suited to your task.

1.4 Menu

1.4.1 File

This dialog allows you to save, load and clear entire projects, this includes any nodes in the data manager.

1.4.2 Edit

This dialog supports undo and redo operations as well as the image navigator, which gives you sliders to navigate through the data quickly.

1.4.3 Window

This dialog allows you to open a new window, change between perspectives and reset your current one to default settings.

If you want to use an operation of a certain perspective within another perspective the "Show View" menu allows to select a specific function that is opened and can be moved within the working areas according to your wishes. Be aware that not every function works with every perspective in a meaningful way.

The Preferences dialog allows you to adjust and save your custom settings.

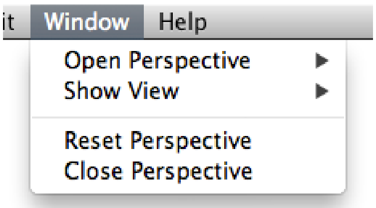


Figure 1.4: Preferences

1.4.4 Help

This dialog contains this help, the welcome screen and information about MITK.

1.5 Levelwindow

Once an image is loaded the levelwindow appears to the right hand side of the four window view. With this tool you can adjust the range of grey values displayed and the gradient between them. Moving the lower boundary up results in any pixels having a value lower than that boundary to be displayed as black. Lowering the upper boundary causes all pixels having a value higher than it to be displayed as white.

The pixels with a value between the lower and upper boundary are displayed in different shades of grey. This way a smaller levelwindow results in higher contrasts while cutting of the information outside its range whereas a larger levelwindow displays more information at the cost of contrast and detail.

You can pick the levelwindow with the mouse to move it up and down, while moving the mouse cursor to the left or right to change its size. Picking one of the boundaries with a left click allows you to change the size symmetrically. Holding CTRL and clicking a boundary adjusts only that value.

With line edit fields below you can directly adjust the levelwindow. The upper field describes the center of the levelwindow, the bottom the span of the window around the center. By selecting one of fields and typing any number you can set these two parameters.

1.6 System Load Indicator

The System Load Indicator in the lower right hand corner of the screen gives information about the memory currently required by the MITK application. Keep in mind that image processing is a highly memory intensive task and monitor the indicator to avoid your system freezing while constantly swapping to the hard drive.

1.7 Views

Each solution for a specific problem that is self contained is realized as a single view. Thus you can create a workflow for your problem by combining the capabilities of different views to suit your needs. One elegant way to do this is by combining views in Perspectives.

By pressing and holding the left mouse button on a views tab you can move it around to suit your needs, even out of the application window.

1.8 Perspectives

The different tasks that arise in medical imaging need very different approaches. To acknowledge this circumstance MITK supplies a framework that can be build uppon by very different solutions to those tasks. These solutions are called perspectives, each of them works independently of others although they might be used in sequence to achieve the solution of more difficult problems.

It is possible to switch between the perspectives using the "Window"->"Open Perspective" dialog.

See Menu for more information about switching perspectives.

Chapter 2

Using The Diffusion Imaging Application

2.1 What is the Diffusion Imaging Application

The Diffusion Imaging Application contains selected views for the analysis of images of the human brain. These encompass the views developed by the `Medical Image Computing Group` of the Division Medical and Biological Informatics as well as basic image processing views such as segmentation and volumevisualization.

For a basic guide to MITK see The MITK User Manual .

Part II

Diffusion imaging manuals

Chapter 3

MITK Diffusion Imaging (MITK-DI)

This module provides means to diffusion weighted image reconstruction, visualization and quantification. Diffusion tensors as well as different q-ball reconstruction schemes are supported. Q-ball imaging aims at recovering more detailed information about the orientations of fibers from diffusion MRI measurements and, in particular, to resolve the orientations of crossing fibers.

3.1 Components

MITK Diffusion consists of further components, which have their own documentation, see:

3.1.1 Data Import

- Importing Data

3.1.2 Preprocessing and Reconstruction

Register or denoise original data and convert it into tensor or q-ball images.

- DWI Registration
- Preprocessing of Diffusion Images
- Reconstruction
- ODF Visualization
- ODF Details View
- Peak Extraction View
- DWI Denoising

3.1.3 Fiber Tracking

Create and work with fiber tractographies.

- Fiber Processing View
- Gibbs Tracking View
- Stochastic Tracking View

- Fiber Extraction View
- Fiber Processing View
- Streamline Tracking View
- Fiberfox
- Fieldmap Generator View

3.1.4 Quantification

Create parameter maps and quantify different properties.

- Scalar Indices
- Intra-voxel incoherent motion estimation (IVIM)
- Partial Volume Analysis
- The TBSS View

3.1.5 Connectomics

Create and analyse connectome networks.

- The Connectomics Perspective

3.2 Known Issues

- **Dicom Import:** The dicom import has so far only been implemented for Siemens dicom images. MITK-DI is capable of reading the nrrd format, which is documented elsewhere [1, 2]. These files can be created by combining the raw image data with a corresponding textual header file. The file extension should be changed from ∗.nrrd to ∗.dwi or from ∗.nhdr to ∗.hdwi respectively in order to let MITK-DI recognize the diffusion related header information provided in the files.

3.3 Technical Information for Developers

The diffusion imaging module uses additional properties beside the ones in use in other modules, for further information see DiffusionImagingPropertiesPage .

3.4 References

Further reading regarding diffusion:

1. `http://teem.sourceforge.net/nrrd/format.html`
2. `http://www.cmake.org/Wiki/Getting_Started_with_the_NRRD_Format`
3. C.F.Westin, S.E.Maier, H.Mamata, A.Nabavi, F.A.Jolesz, R.Kikinis, "Processing and visualization for Diffusion tensor MRI", Medical image Analysis, 2002, pp 93-108
4. Tuch, D.S., 2004. Q-ball imaging. Magn Reson Med 52, 1358-1372.

5. Descoteaux, M., Angelino, E., Fitzgibbons, S., Deriche, R., 2007. Regularized, fast, and robust analytical Q-ball imaging. Magn Reson Med 58, 497-510.

6. Aganj, I., Lenglet, C., Sapiro, G., 2009. ODF reconstruction in q-ball imaging with solid angle consideration. Proceedings of the Sixth IEEE International Symposium on Biomedical Imaging Boston, MA.

7. Goh, A., Lenglet, C., Thompson, P.M., Vidal, R., 2009. Estimating Orientation Distribution Functions with Probability Density Constraints and Spatial Regularity. Med Image Comput Comput Assist Interv Int Conf Med Image Comput Comput Assist Interv LNCS 5761, 877 ff.

8. J.-D. Tournier, S. Mori, A. Leemans., 2011. Diffusion Tensor Imaging and Beyond. Magn Reson Med 65, 1532-1556.

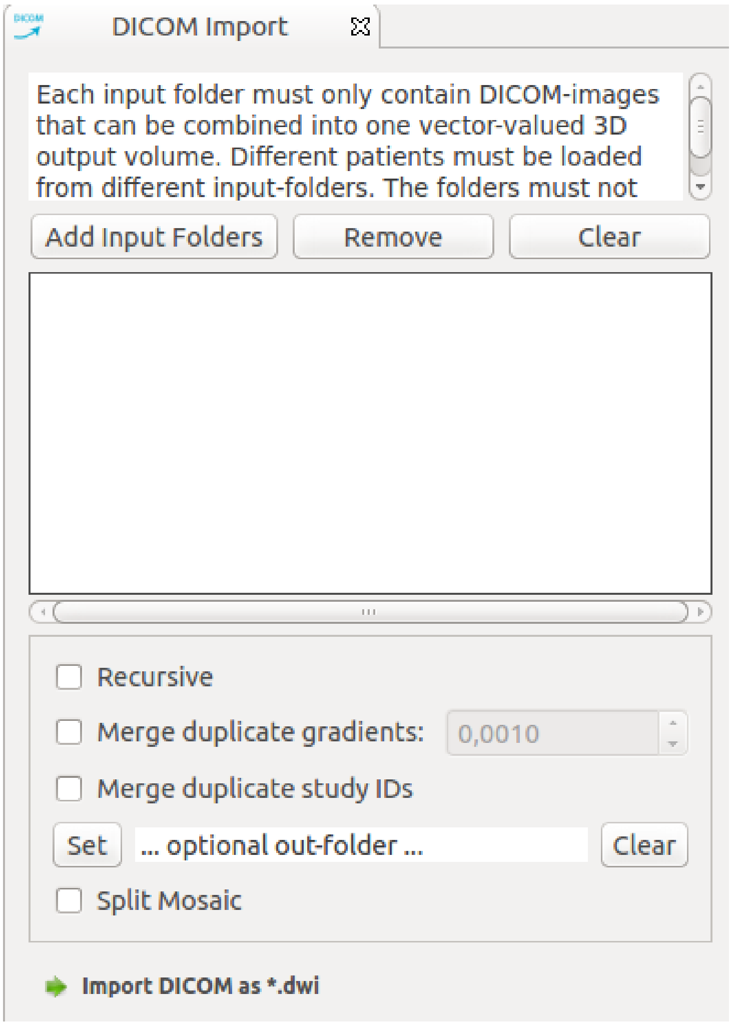
3.5 Importing Data

3.5.1 Dicom Import

The dicom import does not cover all hardware manufacturers but only Siemens dicom images. MITK-DI is also capable of reading the nrrd format, which is documented elsewhere [1, 2]. These files can be created by combining the raw image data with a corresponding textual header file. The file extension should be changed from ∗.nrrd to ∗.dwi or from ∗.nhdr to ∗.hdwi respectively in order to let MITK-DI recognize the diffusion related header information provided in the files.

In case your dicom images are readable by MITK-DI, select one or more input dicom folders and click import. Each input folder must only contain DICOM-images that can be combined into one vector-valued 3D output volume. Different patients must be loaded from different input-folders. The folders must not contain other acquisitions (e.g. T1,T2,localizer).

In case many imports are performed at once, it is recommended to set the the optional output folder argument. This prevents the images from being kept in memory.



Each input folder must only contain DICOM-images that can be combined into one vector-valued 3D output volume. Different patients must be loaded from different input-folders. The folders must not

☐ Recursive

☐ Merge duplicate gradients: 0,0010

☐ Merge duplicate study IDs

... optional out-folder ...

☐ Split Mosaic

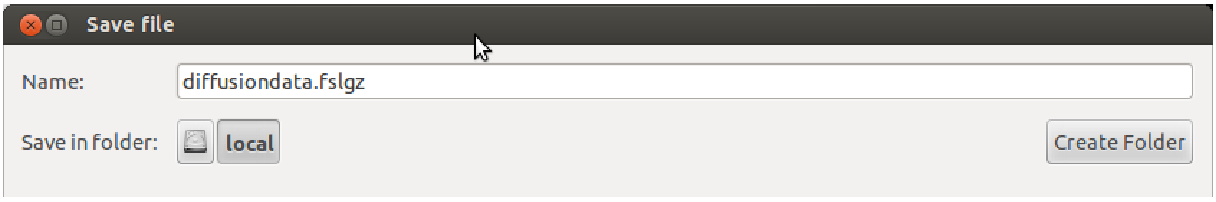
Figure 3.1: Dicom import

The option "Average duplicate gradients" accumulates the information that was acquired with multiple repetitions for one gradient. Vectors do not have to be precisely equal in order to be merged, if a "blur radius" $>$ 0 is configured.

3.5.2 FSL Import

FSL diffusion data can be imported with MITK Diffusion. FSL diffusion datasets consist of 3 files: a nifty file (filename.nii.gz or filename.nii), a bvecs file (filename.bvecs), which is a text file containing the gradient vectors, and a bvals file (filename.bvecs), containing the b-values. Due to the system that selects suitable file readers, MITK will not recognize these files as diffusion datasets. In order to make MITK recognize it as diffusion, the extension must be changed from .nii.gz to .fslgz (so the new name is filename.fslgz) or from filename.nii to filename.fsl. The bvecs and bvals files have to be renamed as well(to filename.fsl.bvecs/filenames.fsl.bvecs or to filename.fslgz.bvecs/filename.fslgz.bvals).

MITK can also save diffusion weighted images in FSL format. To do this the extension of the new file should be changed to .fsl or .fslgz upon saving the file.



Name: diffusiondata.fslgz

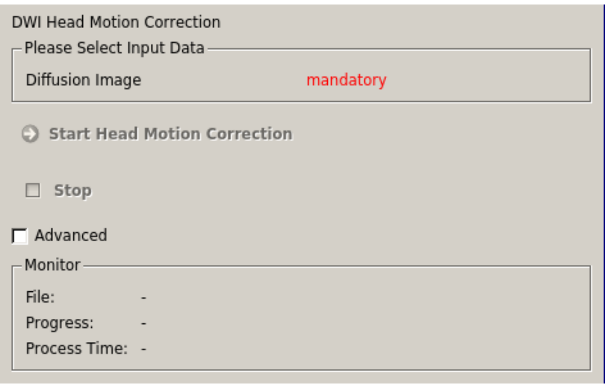
Save in folder: local

Figure 3.2: Save a dwi dataset as fsl

3.6 DWI Registration

3.6.1 DWI Registration

The DWI registration view allows head-motion and eddy-current correction. First the b=0 images (if more than 1) are registered and averaged. The averaged image then serves as fixed image for the alignment of the remaining (weighted, b>0) images.



Diffusion Image mandatory

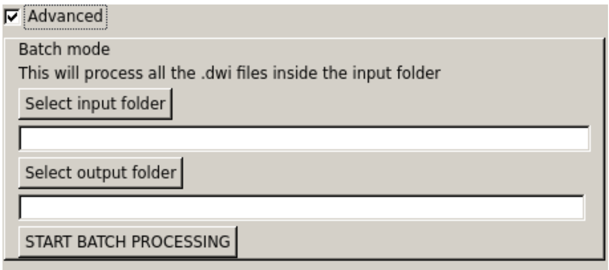
File: -

Progress: -

Process Time: -

Figure 3.3: Basic DWI Registration Interface

In the basic settings, a selected DW image in the Data Manager is marked as Input Data and the *Start Head Motion Correction* button is enabled. If more than one DW images are selected, they will be processed in a consecutive manner. For larger sets of dw images to be processed, the *Advanced* settings can be used.



This will process all the .dwi files inside the input folder

Figure 3.4: Batch Processing DWI Registration Interface

Here all valid .dwi data located in the given input folder will be processed through the head motion correction and the output will be stored in the current working directory or in a custom folder if specified in the user interface. The output image will have the _MC postfix attached to the input's name.

3.7 Preprocessing of Diffusion Images

3.7.1 Preprocessing

The Preprocessing View bundles a selection of features and tools for dealing with diffusion weighted MR images. It can be used to modify header and geometry information, manipulate gradients and gradient images as well as calculating ADC maps.

3.7.2 Details

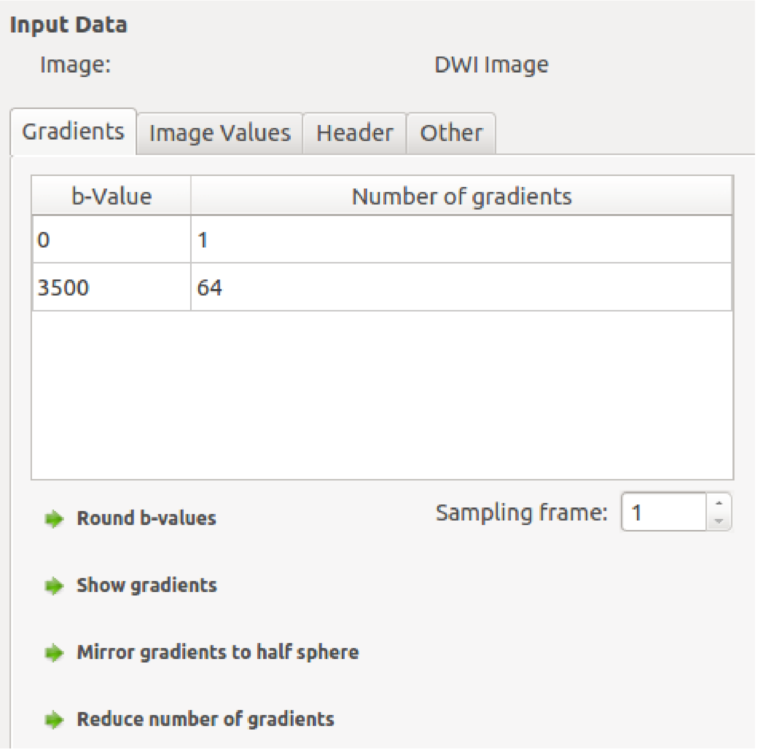

Input Data

Image: DWI Image

b-Value	Number of gradients
0	1
3500	64

Sampling frame: 1

Figure 3.5: Gradients tab

The Gradients tab allows you to examine the number of b values and how many gradients were acquired for each. Here you can also visualize the gradients as points on a sphere, mirror opposite gradients so all are oriented within the same half sphere, round b values and merge similar gradients.

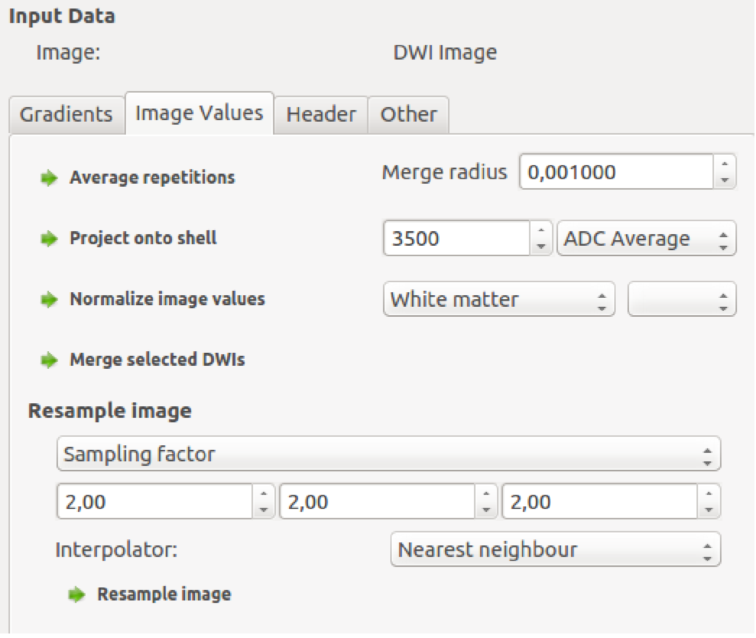


Image: DWI Image

Merge radius 0,001000

3500 ADC Average

White matter

Sampling factor

2,00 2,00 2,00

Interpolator: Nearest neighbour

Figure 3.6: Image values tab

The Image values tab enables you to manipulate the voxel values by e.g. resampling images, averaging repetitions of the same gradient, normalizing the image values or merging several DWIs into one.

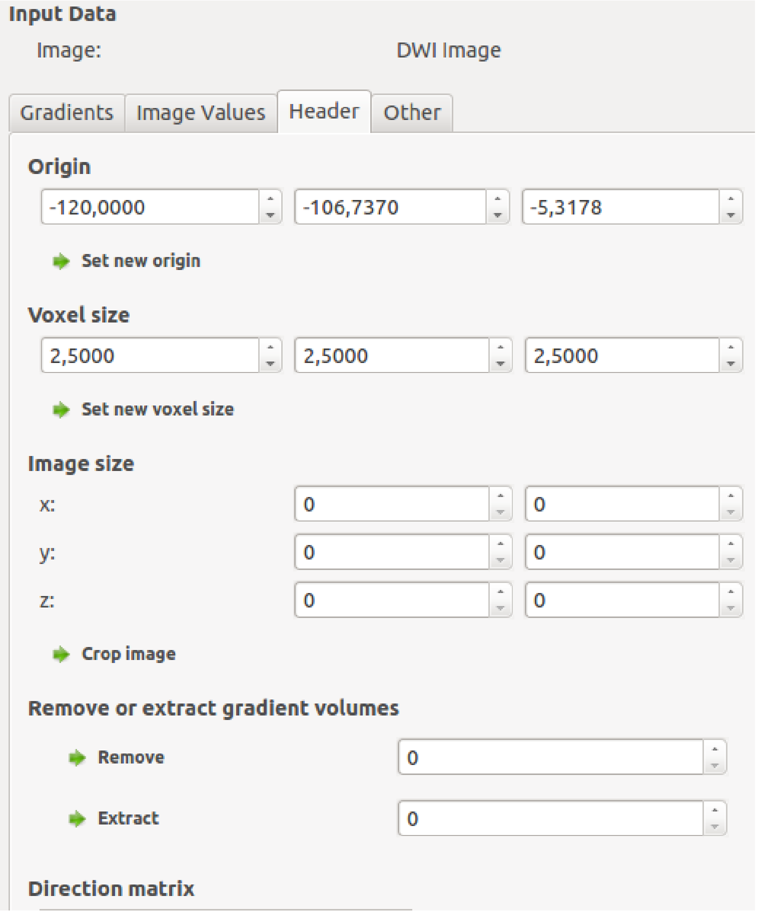


Image: DWI Image

Origin: -120,0000 -106,7370 -5,3178

Voxel size: 2,5000 2,5000 2,5000

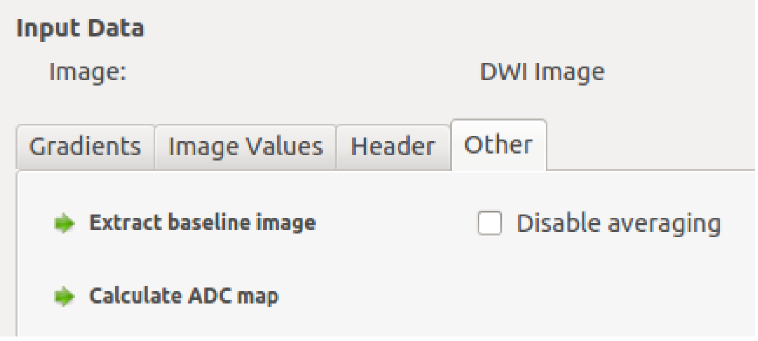
Image size:
x: 0 0
y: 0 0
z: 0 0

Remove: 0

Extract: 0

Figure 3.7: Header tab

In the Header tab you can selectively change geometry information of the image and delete or extract specific gradient volumes.



Input Data

Image: DWI Image

☐ Disable averaging

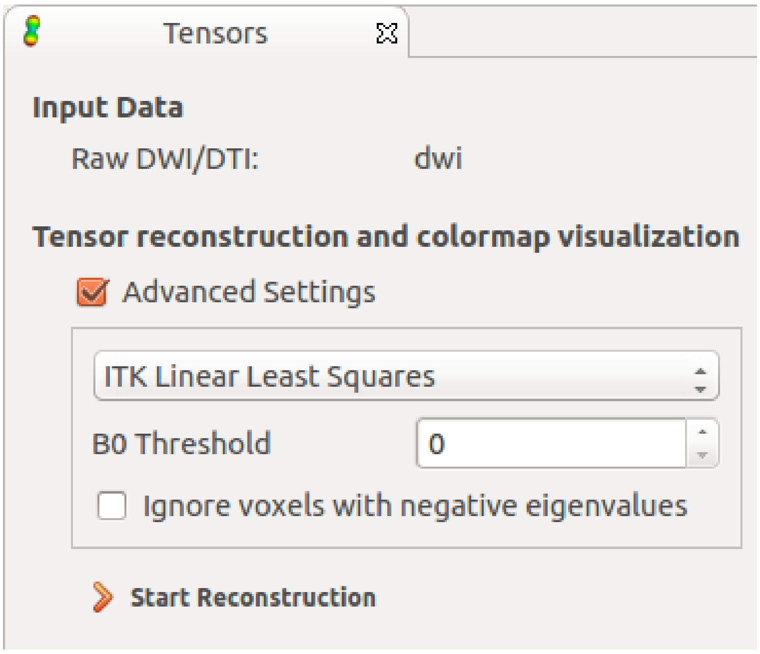
Figure 3.8: Other tab

Miscellaneous tools in the Other tab allow for ADC map calculation and the extraction of the b0 image with the possibility of averaging all b0 images.

3.8 Reconstruction

3.8.1 Tensor Reconstruction

The tensor reconstruction view allows ITK based tensor reconstruction [3]. The advanced settings for ITK reconstruction let you configure a manual threshold on the non-diffusion weighted image. All voxels below this threshold will not be reconstructed and left blank. It is also possible to check for negative eigenvalues. The according voxels are also left blank.



Input Data

Raw DWI/DTI: dwi

Tensor reconstruction and colormap visualization

☑ Advanced Settings

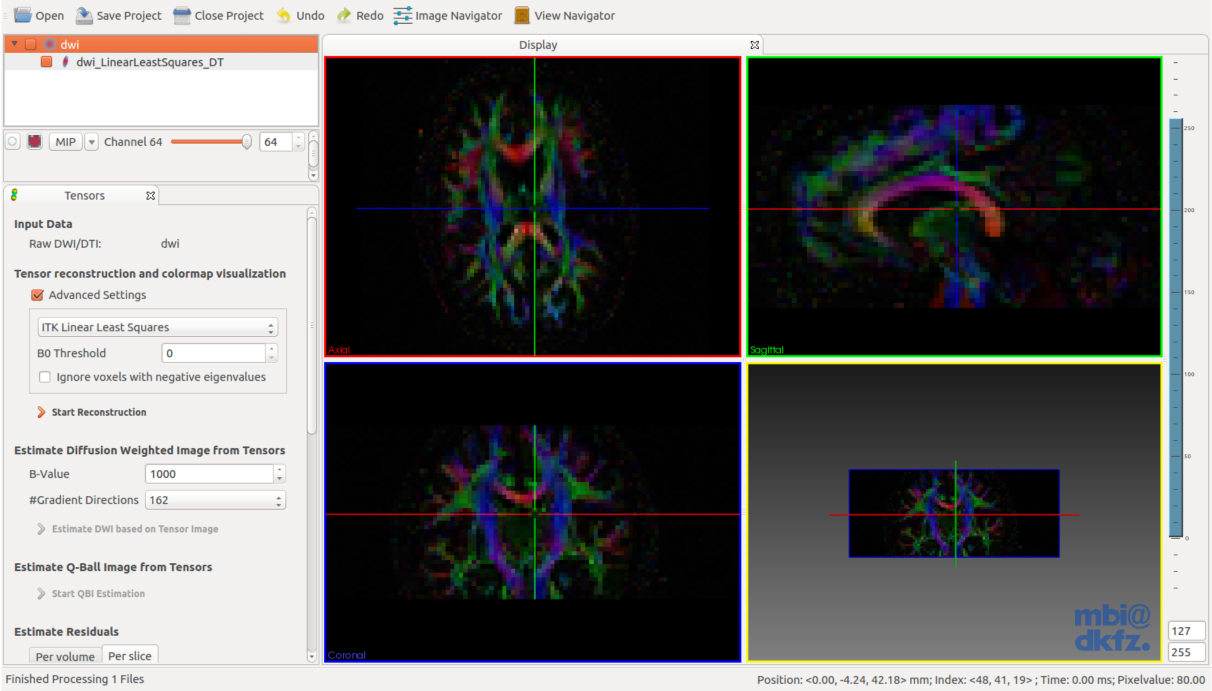
ITK Linear Least Squares

B0 Threshold: 0

☐ Ignore voxels with negative eigenvalues

Figure 3.9: ITK tensor reconstruction

A few seconds (depending on the image size) after the reconstruction button is hit, a colored image should appear in the main window.



Input Data
Raw DWI/DTI: dwi

Tensor reconstruction and colormap visualization
☑ Advanced Settings
ITK Linear Least Squares
B0 Threshold 0
☐ Ignore voxels with negative eigenvalues

Estimate Diffusion Weighted Image from Tensors
B-Value 1000
#Gradient Directions 162

Estimate Q-Ball Image from Tensors

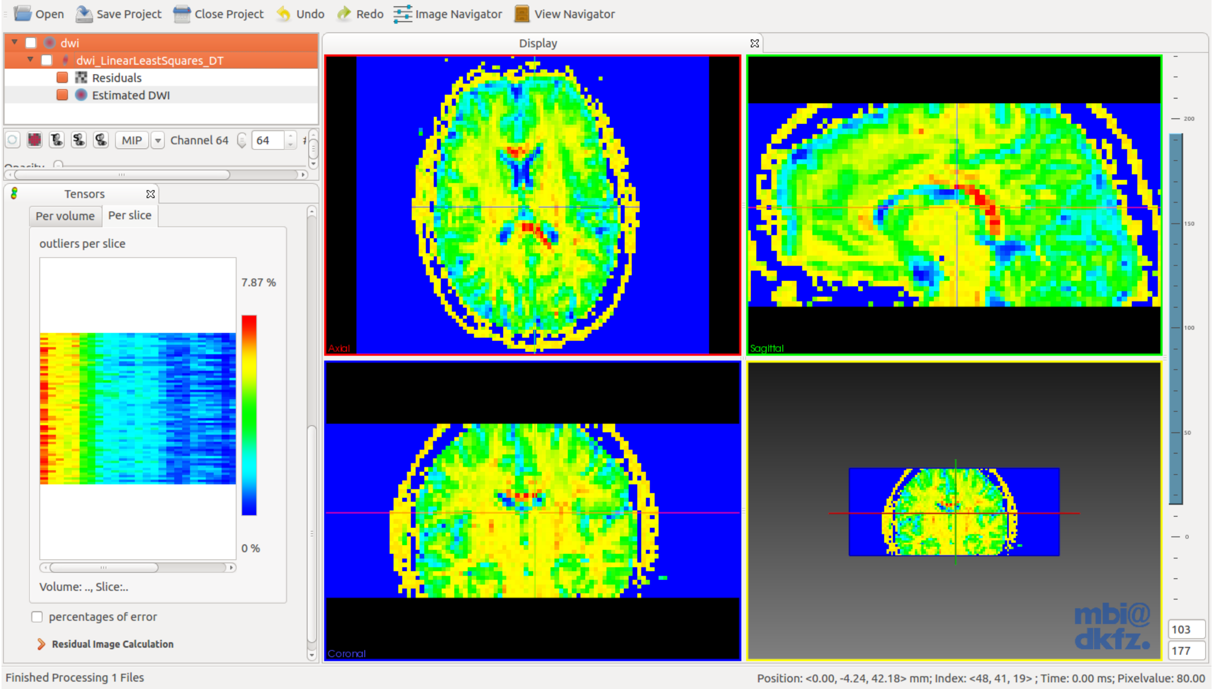
Estimate Residuals

Finished Processing 1 Files

Position: <0.00, -4.24, 42.18> mm; Index: <48, 41, 19> ; Time: 0.00 ms; Pixelvalue: 80.00

Figure 3.10: Tensor image after reconstruction

To assess the quality of the tensor fit it has been proposed to calculate the model residual [9]. This calculates the residual between the measured signal and the signal predicted by the model. Large residuals indicate an inadequacy of the model or the presence of artefacts in the signal intensity (noise, head motion, etc.). To use this option: Select a DWI dataset, estimate a tensor, select both the DWI node and the tensor node in the datamanager and press Residual Image Calculation. MITK-Diffusion can show the residual for every voxel averaged over all volumes or (in the plot widget) summarized per volume or for every slice in every volume. Clicking in the widget where the residual is shown per slice will automatically let the cross-hair jump to that position in the DWI dataset. If Percentage of outliers is checked, the per volume plot will show the percentage of outliers per volume. Otherwise it will show the mean together with the first and third quantile of residuals. See [9] for more information.



Finished Processing 1 Files

Position: <0.00, -4.24, 42.18> mm; Index: <48, 41, 19> ; Time: 0.00 ms; Pixelvalue: 80.00

Figure 3.11: The residual widget

The view also allows the generation of artificial diffusion weighted or Q-Ball images from the selected tensor image. The ODFs of the Q-Ball image are directly initialized from the tensor values and afterwards normalized. The diffusion weighted image is estimated using the l2-norm image of the tensor image as B0. The gradient images are afterwards generated using the standard tensor equation.

3.8.2 Q-Ball Reconstruction

The q-ball reonstruction view implements a variety of reconstruction methods. The different reconstruction methods are described in the following:

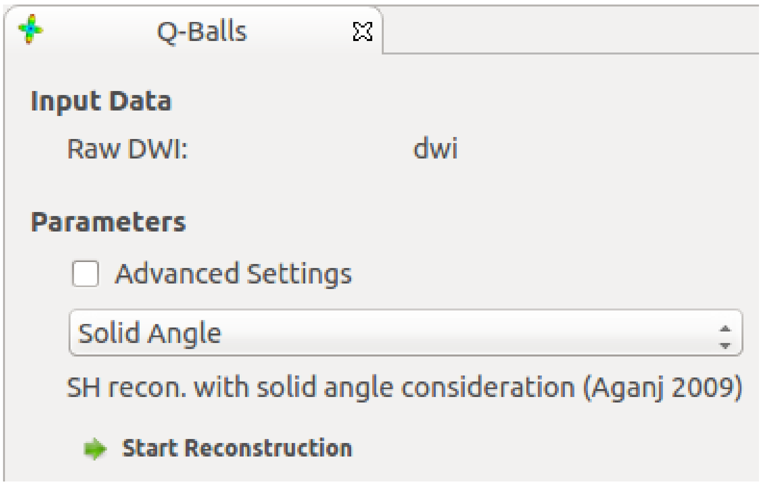
- **Numerical:** The original, numerical q-ball reconstruction presented by Tuch et al. [5]

- **Standard (SH):** Descoteaux's reconstruction based on spherical harmonic basis functions [6]

- **Solid Angle (SH):** Aganj's reconstruction with solid angle consideration [7]

- **ADC-profile only:** The ADC-profile reconstructed with spherical harmonic basis functions

- **Raw signal only:** The raw signal reconstructed with spherical harmonic basis functions



Input Data

Raw DWI: dwi

Parameters

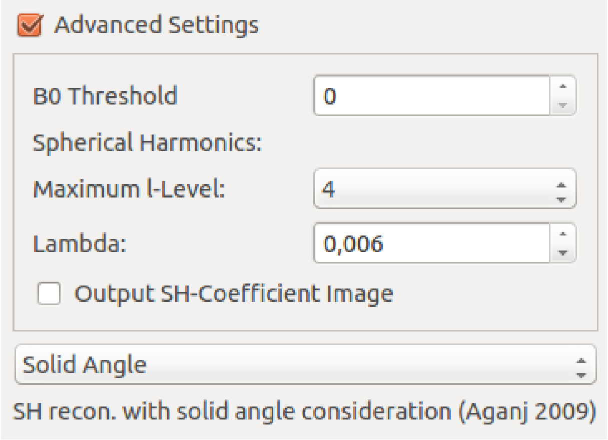
☐ Advanced Settings

Solid Angle

SH recon. with solid angle consideration (Aganj 2009)

Figure 3.12: The q-ball resonstruction view

B0 threshold works the same as in tensor reconstruction. The maximum l-level configures the size of the spherical harmonics basis. Larger l-values (e.g. l=8) allow higher levels of detail, lower levels are more stable against noise (e.g. l=4). Lambda is a regularisation parameter. Set it to 0 for no regularisation. lambda = 0.006 has proven to be a stable choice under various settings.



☑ Advanced Settings

B0 Threshold 0

Spherical Harmonics:

Maximum l-Level: 4

Lambda: 0,006

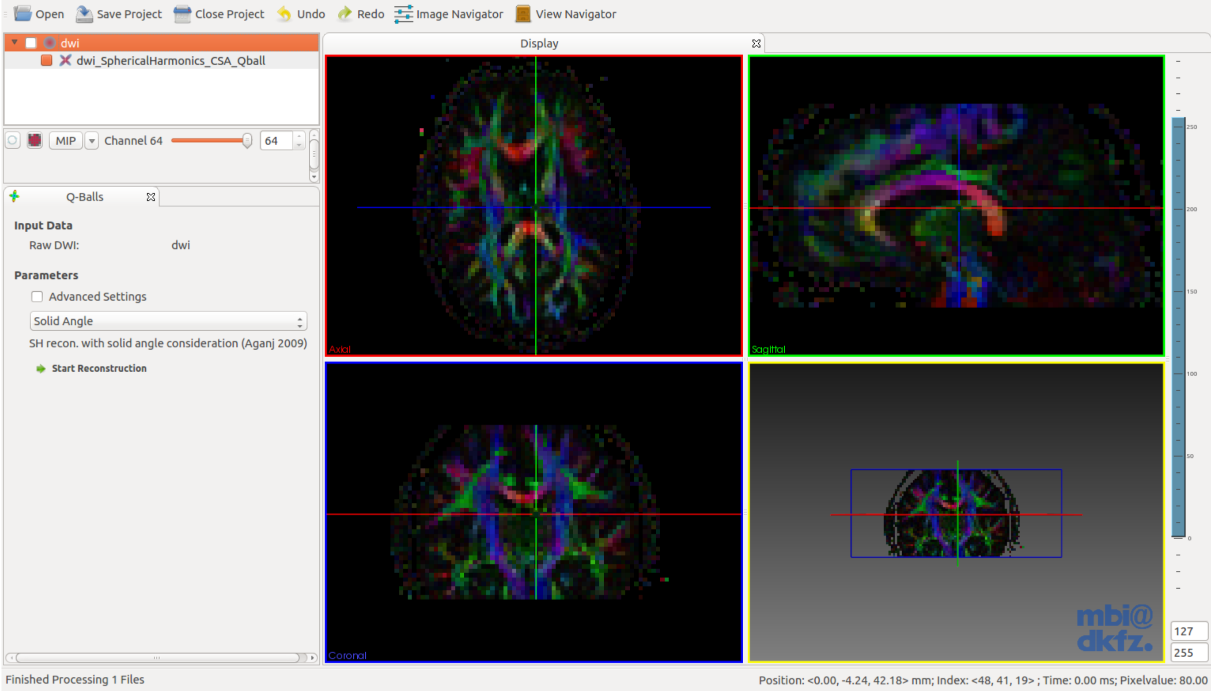
☐ Output SH-Coefficient Image

Solid Angle

SH recon. with solid angle consideration (Aganj 2009)

Figure 3.13: Advanced q-ball reconstruction settings

This is how a q-ball image should initially look after reconstruction. Standard q-balls feature a relatively low GFA and thus appear rather dark. Adjust the level-window to solve this.



Input Data

Raw DWI: dwi

Parameters

☐ Advanced Settings

Solid Angle

SH recon. with solid angle consideration (Aganj 2009)

Start Reconstruction

Finished Processing 1 Files

Position: <0.00, -4.24, 42.18> mm; Index: <48, 41, 19> ; Time: 0.00 ms; Pixelvalue: 80.00

Figure 3.14: q-ball image after reconstruction

3.9 ODF Visualization

3.9.1 ODF Visualization Setting

In this small view, the visualization of ODFs and diffusion images can be configured. Depending on the selected image in the data storage, different options are shown here.

For tensor or q-ball images, the visibility of glyphs in the different render windows (T)ransversal, (S)agittal, and (C)oronal can be configured here. The maximal number of glyphs to display can also be configured here for. This is usefull to keep the system response time during rendering feasible. The other options configure normalization and scaling of the glyphs.

In diffusion images, a slider lets you choose the desired image channel from the vector of images (each gradient direction one image) for rendering. Furthermore reinit can be performed and texture interpolation toggled.

This is how a visualization with activated glyphs should look like:

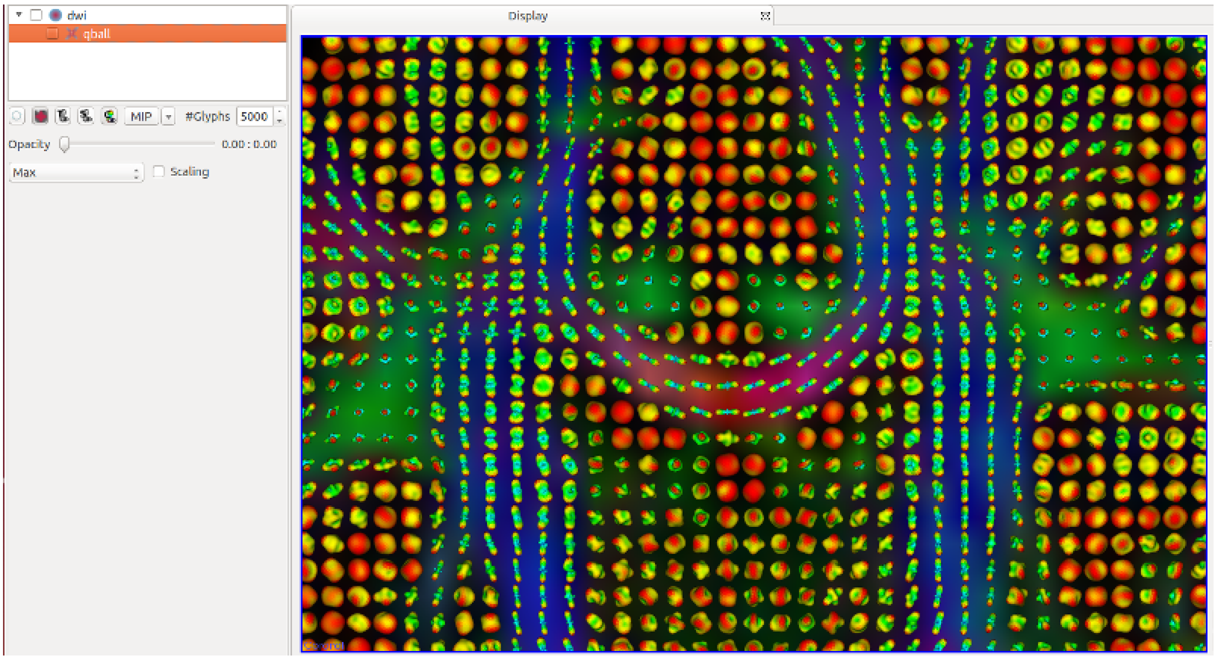
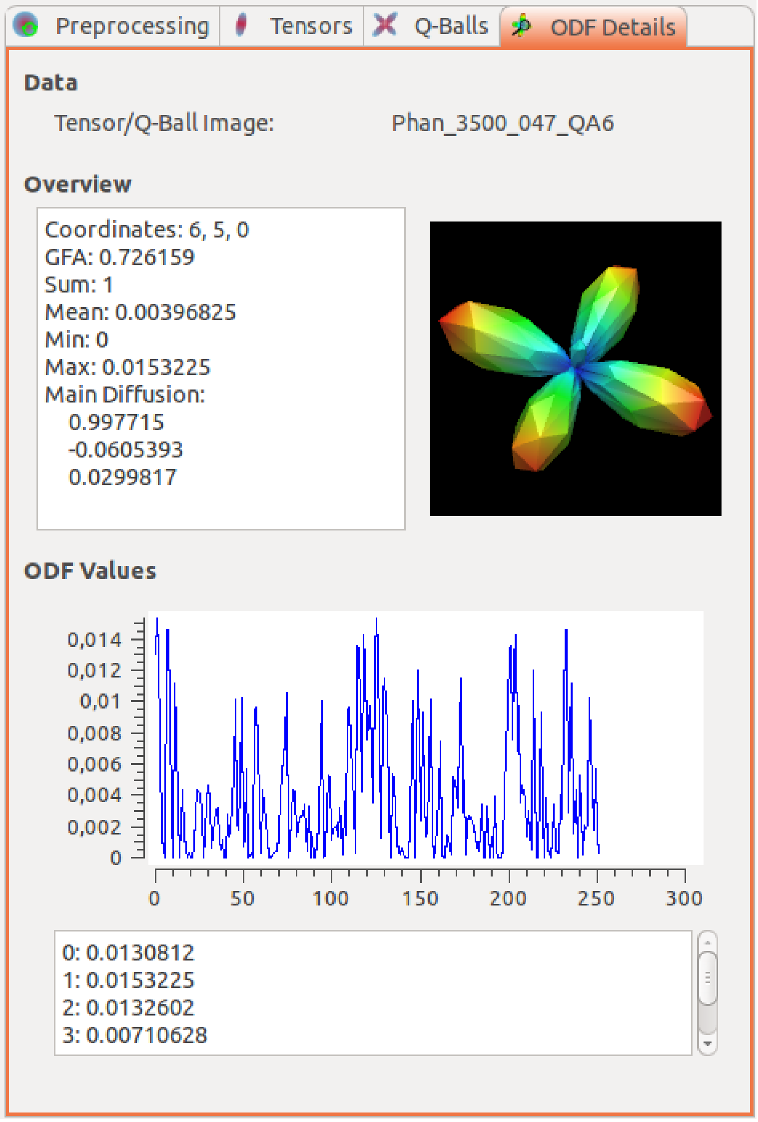


Figure 3.15: Q-ball image with ODF glyph visibility toggled ON

3.10 ODF Details View

This view provides detailed information about the orentation distribution function at the current crosshair position (if a Tensor/Q-Ball image is selected). A visualization of the ODF as well as statistical information are displayed.



Data

Tensor/Q-Ball Image: Phan_3500_047_QA6

Overview

Coordinates: 6, 5, 0
GFA: 0.726159
Sum: 1
Mean: 0.00396825
Min: 0
Max: 0.0153225
Main Diffusion:
0.997715
-0.0605393
0.0299817

ODF Values

0: 0.0130812
1: 0.0153225
2: 0.0132602
3: 0.00710628

Figure 3.16: The ODF Details View

3.11 Peak Extraction View

This view provides the user interface to extract the peaks of tensors and the spherical harmonic representation of Q-Balls.

Available sections:

- Input Data
- Output Data
- Peak Extraction Methods
- Input Parameters
- References

3.11.1 Input Data

Mandatory Input:

- DTI image or image containing the spherical harmonic coefficients. The SH coefficient images can be obtain from the Q-Ball reconstruction view by enabling the checkbox in the advanced options.

Optional Input:

- Binary mask to define the extraction area.

3.11.2 Output Data

- Vector field: 3D representation of the resulting peaks. Only for visualization purposes (the peaks are scaled additionally to the specified normalization to improve the visualization)!

- # Directions per Voxel: Image containing the number of extracted peaks per voxel as image value.

- Direction Images: One image for each of the extracted peaks per voxel. Each voxel contains one direction vector as image value. Use this output for evaluation purposes of the extracted peaks.

3.11.3 Peak Extraction Methods

- If a tensor image is used as input, the output is simply the largest eigenvector of each voxel.

- The finite differences extraction uses a higly sampled version of the image ODFs, extracts all local maxima and clusters the resulting directions that point in a similar direction.

- For details about the analytical method (experimental) please refer to [1].

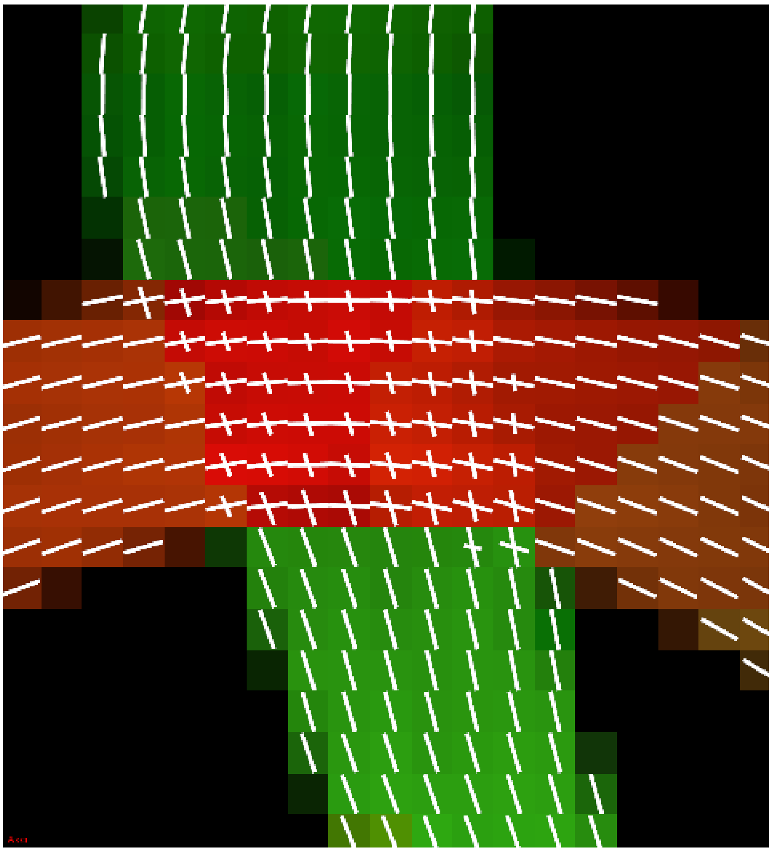


Figure 3.17: Peaks of a fiber crossing extracted using finite differences method.

3.11.4 Input Parameters

- Vector normalization method (no normalization, maximum normalization of the vecors of one voxel and independent normalization of each vecor).
- SH Order: Specify the order of the spherical harmonic coefficients.
- Maximum number of peaks to extract. If more peaks are found only the largest are kept.
- Threshold to discard small peaks. Value relative to the largest peak of the respective voxel.
- Absolute threshold on the peak size. To evaluate this threshold the peaks are additionally weighted by their GFA (low GFA voxels are more likely to be discarded). This threshold is only used for the finite differences extraction method.

3.11.5 References

[1] Aganj et al. Proceedings of the Thirteenth International Conference on Medical Image Computing and Computer Assisted Intervention 2010

3.12 DWI Denoising

This view provides the user interface to denoise a diffusion weighted image (DWI) with several methods.

Available sections:

- Input Data

- Output Data
- Non-local means filter
- Discrete gaussian filter
- References

3.12.1 Input Data

Mandatory Input:

- Diffusion weigthed image

Optional Input:

- Binary mask to define a denoising area.

3.12.2 Output Data

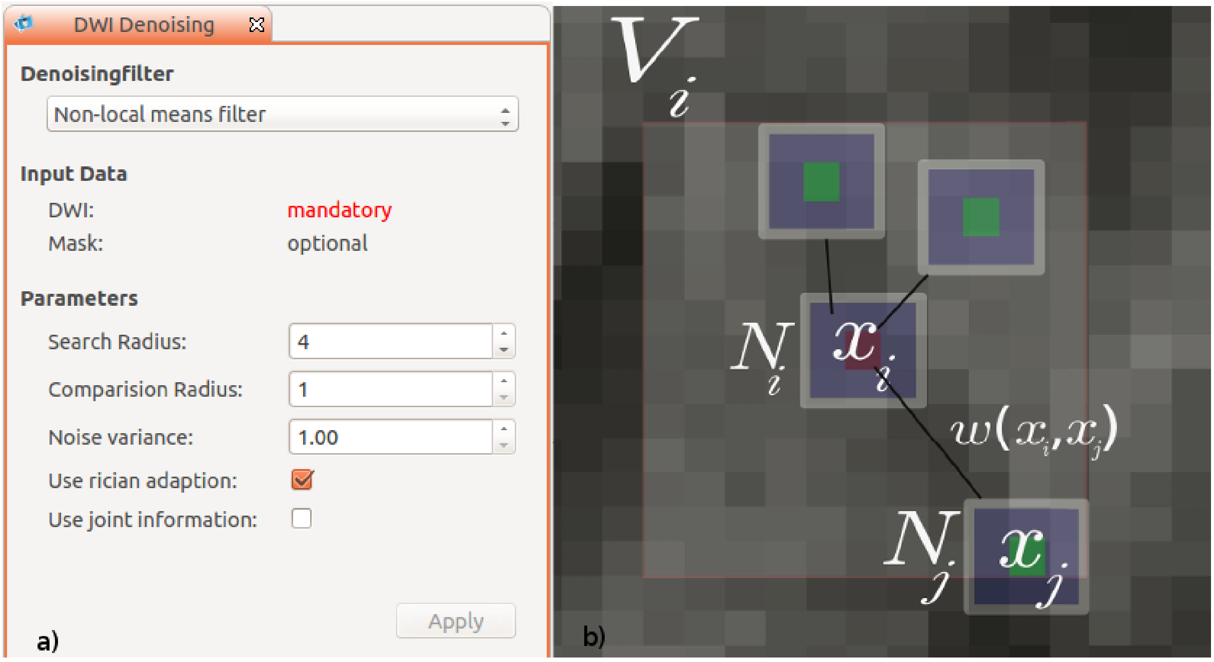
- Denoised DWI: if a mask is set for denoising, all voxel excluding the masked area are set to 0.

3.12.3 Non-local means filter

Denoise the input DWI using a non local means algorithm. For more details refer to [1] and [2].

Parameters:

- Searchradius: defines the size of the neighborhood V (Fig. 1 b)) in which the voxels will be weighted to reconstruct the center voxel. The resulting neighborhood size is defined as 2x searchradius + 1 (e.g. a searchradius of 1 generates a neighborhood cube with size 3x3x3).
- Comparisonradius: defines the size of the compared neighborhoods N (Fig. 1 b)) around each voxel. The resulting neighborhood size is defined as 2x comaprisonradius + 1 (e.g. a comparisonradius of 1 generates a neighborhood cube with size 3x3x3).
- Noise variance: the variance of the noise need to be set for filtering. An estimation of the noise varinace will be implemented soon.
- Rician adaption: if checked the non-local means uses an adaption for Rician distributed noise.
- Joint information: if checked the whole DWI is seen as a vector image, weighting each voxels complete vector, instead of weighting each channel seperate. (This might be a bit faster, but is less accurate)



Denoisingfilter

Non-local means filter

Input Data

DWI: mandatory

Mask: optional

Parameters

Search Radius: 4

Comparision Radius: 1

Noise variance: 1.00

Use rician adaption: ☑

Use joint information: ☐

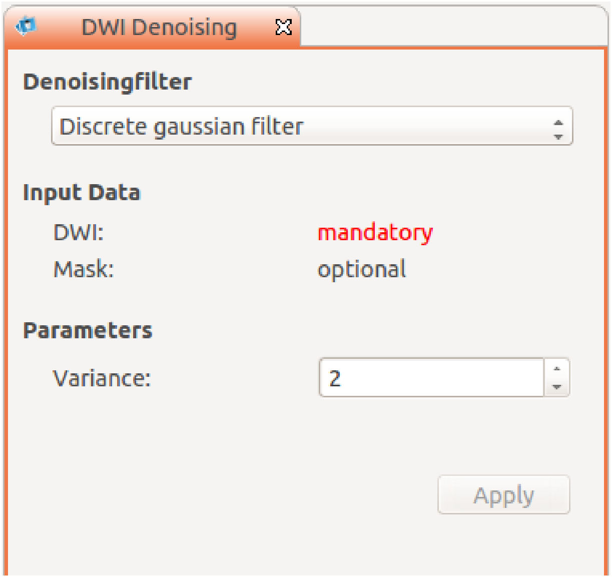
Figure 3.18: Fig. 1: a) View using the Non-local means filter b) 2D illustration of the Non-local means principle [1]

3.12.4 Discrete gaussian filter

Denoise the input DWI using a discrete gaussian filter.

Parameters:

- Variance: defines the varinance of the gaussian distribution to denoise the image.



Denoisingfilter

Discrete gaussian filter

Input Data

DWI: mandatory

Mask: optional

Parameters

Variance: 2

Figure 3.19: Fig. 2: View using the discrete gaussian filter

3.12.5 References

[1] Wiest-Daesslé et al. Non-Local Means Variants for Denoising of Diffusion-Weigthed and Diffusion Tensor MRI. MICCAI 2007, Part II, LNCS 4792, pp- 344-351.

[2] Wiest-Daesslé et al. Rician Noise Removal by Non-Local Means Filtering for Low Signal-to-Noise Ratio MRI: Applications to DT-MRI. MICCAI 2008, Part II, LNCS 5242, pp. 171-179.

3.13 Fiber Processing View

This view provides tools to modify and postprocess the selected fiber bundle as well as additional methods such as TDI.

Generation of additional data from fiber bundles:

- Tract density image: generate a 2D heatmap from a fiber bundle
- Binary envelope: generate a binary image from a fiber bundle
- Fiber bundle image: generate a 2D rgba image representation of the fiber bundle
- Fiber endings image: generate a 2D image showing the locations of fiber endpoints
- Fiber endings pointset: generate a poinset containing the locations of fiber endpoints

Fiber bundle postprocessing:

- The selected fiber bundle can be smoothed by interpolating the fiber points using Kochanek splines.
- The fiber bundle can be pruned using a length or curvature threshold.
- The fiber bundle can be mirrored in aqll three dimensions.
- If a float image with pixel values between 0 and 1 is selcted, the fiber bundle can be colored according to the pixel values (e.g. using an FA image).

Calculate main fiber directions:

Extracts the voxel-wise main fiber directions from a tractogram.

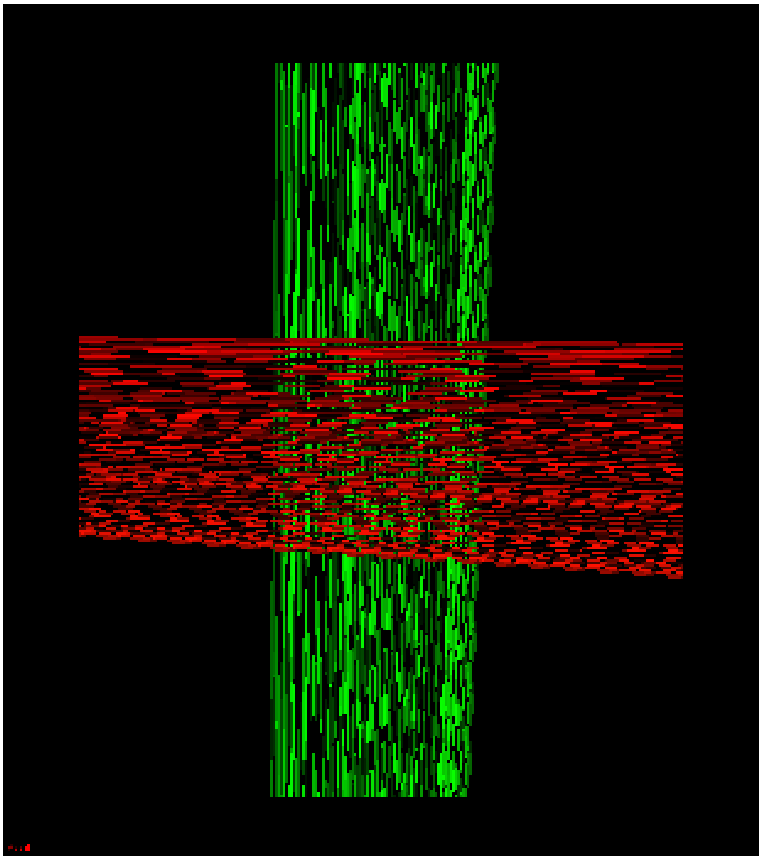


Figure 3.20: Input fiber bundle

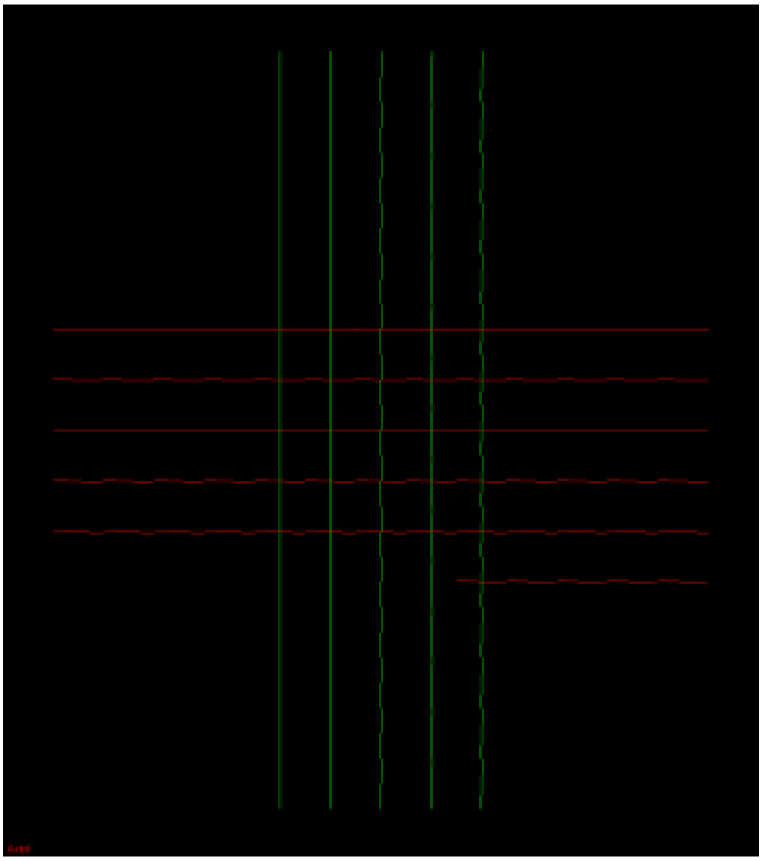
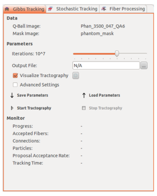


Figure 3.21: Output main fiber directions

3.14 Gibbs Tracking View

This view provides the user interface for the Gibbs Tracking algorithm, a global fiber tracking algorithm, originally proposed by Reisert et.al. [1].



Data

Q-Ball Image: Phan_3500_047_QA6
Mask Image: phantom_mask

Parameters

Iterations: 10^7
Output File: N/A
☑ Visualize Tractography
☐ Advanced Settings

Monitor

Progress: -
Accepted Fibers: -
Connections: -
Particles: -
Proposal Acceptance Rate: -
Tracking Time: -

Figure 3.22: The Gibbs Tracking View

3.14.1 Input Data

Mandatory Input:

- One Q-Ball or tensor image selected in the datamanager

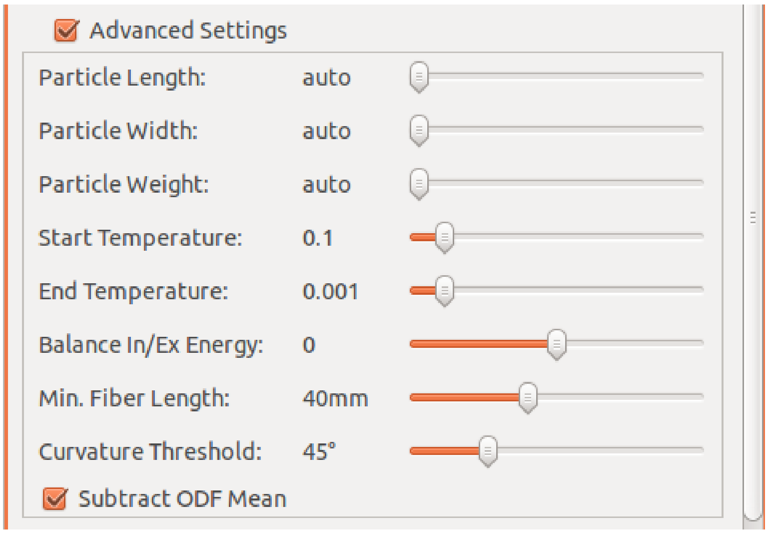
Optional Input:

- Mask Image: White matter probability mask. Corresponds to the probability to generate fiber segments in the respective voxel.

3.14.2 Q-Ball Reconstruction

- Number of iterations: More iterations causes the algorithm to be more stable but also to take longer to finish the tracking. Recommended: 10^8 to 5x10^8 iterations for full brain tractography.
- Particle length/width/weight controlling the contribution of each particle to the model M
- Start and end temperature controlling how fast the process reaches a stable state. (usually no change needed)
- Weighting between the internal (affinity of the model to long and straigt fibers) and external energy (affinity of the model towards the data). (usually no change needed).
- Minimum fiber length constraint (in mm). Shorter fibers are discarded after the tracking.

The automatic selection of parameters for the particle length/width and weight are determined directly from the input image using information about the image spacing and GFA.



☑ Advanced Settings

Particle Length: auto

Particle Width: auto

Particle Weight: auto

Start Temperature: 0.1

End Temperature: 0.001

Balance In/Ex Energy: 0

Min. Fiber Length: 40mm

Curvature Threshold: 45°

☑ Subtract ODF Mean

Figure 3.23: Advanced Tracking Parameters

3.14.3 Surveilance of the tracking process

Once started, the tracking can be monitored via the textual output that informs about the tracking progress and several stats of the current state of the algorithm. If enabled, the intermediate tracking results are displayed in the renderwindows each second. This live visualization should usually be disabled for performance reasons. It can be turned on and off during the tracking process via the according checkbox. The button next to this checkbox allows the visualization of only the next iteration step.

3.14.4 References

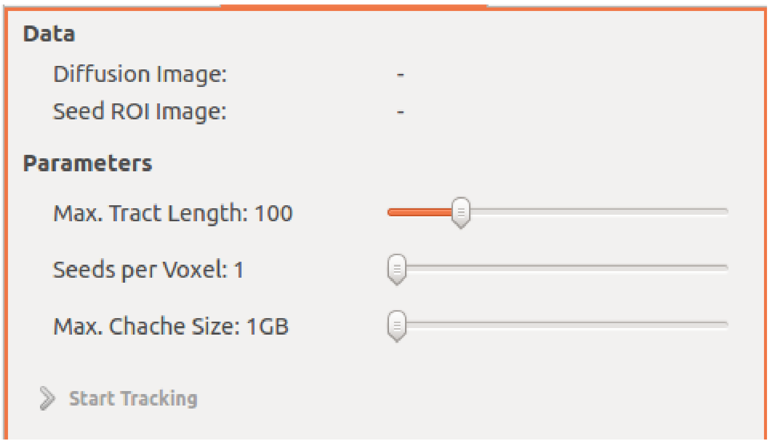
[1] Reisert, M., Mader, I., Anastasopoulos, C., Weigel, M., Schnell, S., Kiselev, V.: Global fiber reconstruction becomes practical. Neuroimage 54 (2011) 955-962

3.15 Stochastic Tracking View

This view provides the user interface for the Stochastic Fibertracking algorithm, proposed by Ngo [1].

Available sections:

- Input Data
- Input Parameters
- References


Data

Diffusion Image: -

Seed ROI Image: -

Parameters

Max. Tract Length: 100

Seeds per Voxel: 1

Max. Chache Size: 1GB

Figure 3.24: Stochastic Tracking View

3.15.1 Input Data

Mandatory Input:

- One DWI Image image selected in the datamanager
- One or more ROIs selected in the datamanager

For a successful execution of the stochastic tractography filter, a DWI input and a binary image defining the desired ROI are required. The ROI serves as the origin from where on the fibers are beeing tracked. To generate the seed ROI the segmentation view in the quantification perspective can be used or a binary image can be loaded.

3.15.2 Input Parameters

- Parameters such as max. tract length, number of seeds per voxel and likelihood cache size in MB can be controlled individually.
- After successfully setting necessary Input and Parameter, pressing the command button executes the algorithm.

3.15.3 References

[1] Tri M. Ngo, Polina Golland, and Tri M. Ngo. A stochastic tractography system and applications, 2007

3.16 Fiber Extraction View

This view provides tools to extract subbundles from a given fiber bundle.

Place ROIs in the 2D render widgets (cricles or polygons) and extract fibers from the bundle that pass through these ROIs by selecting the according ROI and fiber bundle in the datamanger and starting the extraction. The ROIs can be combined via logical operations. All fibers that pass through the thus generated composite ROI are extracted. The extraction can also be performed using 3D ROIs represented as binary mask images. In this extraction method, the logical operations are not implemented at the moment.

3.17 Fiber Processing View

This view provides tools to modify and postprocess the selected fiber bundle as well as additional methods such as TDI.

Generation of additional data from fiber bundles:

- Tract density image: generate a 2D heatmap from a fiber bundle
- Binary envelope: generate a binary image from a fiber bundle
- Fiber bundle image: generate a 2D rgba image representation of the fiber bundle
- Fiber endings image: generate a 2D image showing the locations of fiber endpoints
- Fiber endings pointset: generate a poinset containing the locations of fiber endpoints

Fiber bundle postprocessing:

- The selected fiber bundle can be smoothed by interpolating the fiber points using Kochanek splines.
- The fiber bundle can be pruned using a length or curvature threshold.
- The fiber bundle can be mirrored in aqll three dimensions.
- If a float image with pixel values between 0 and 1 is selcted, the fiber bundle can be colored according to the pixel values (e.g. using an FA image).

Calculate main fiber directions:

Extracts the voxel-wise main fiber directions from a tractogram.

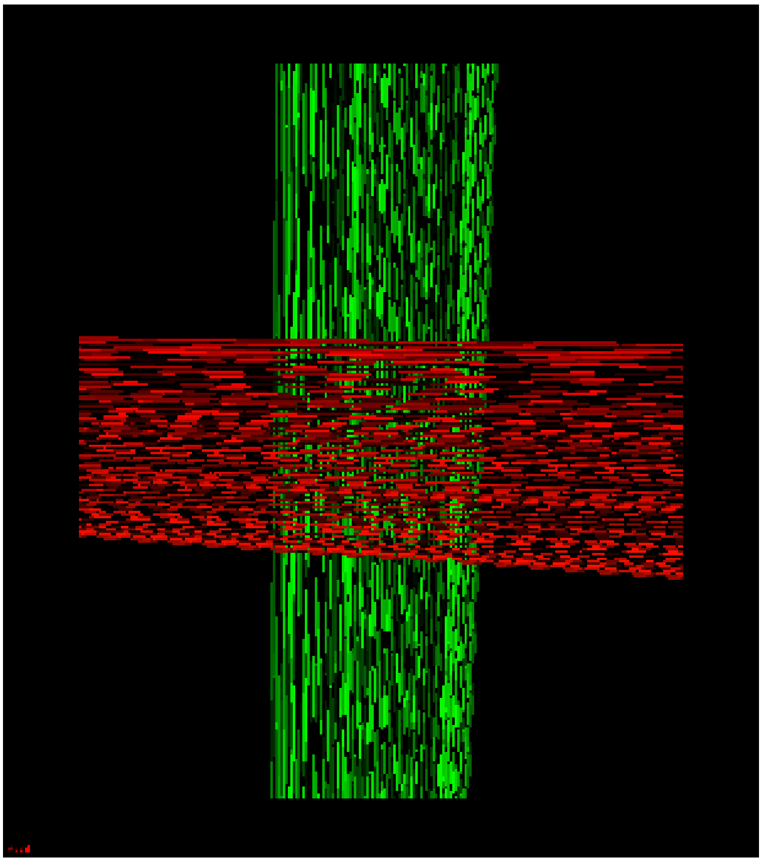


Figure 3.25: Input fiber bundle

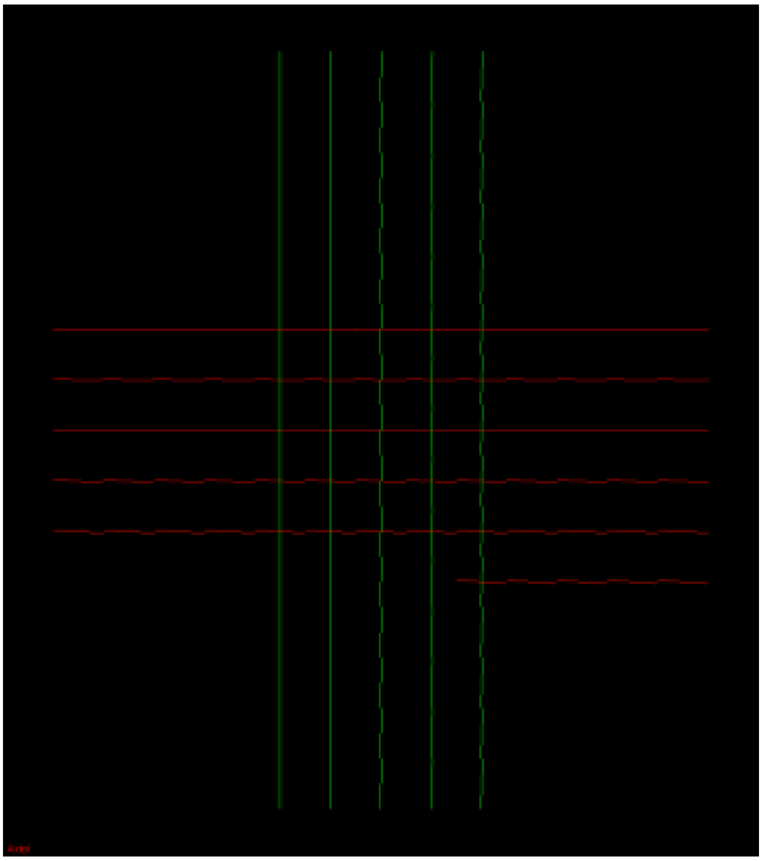


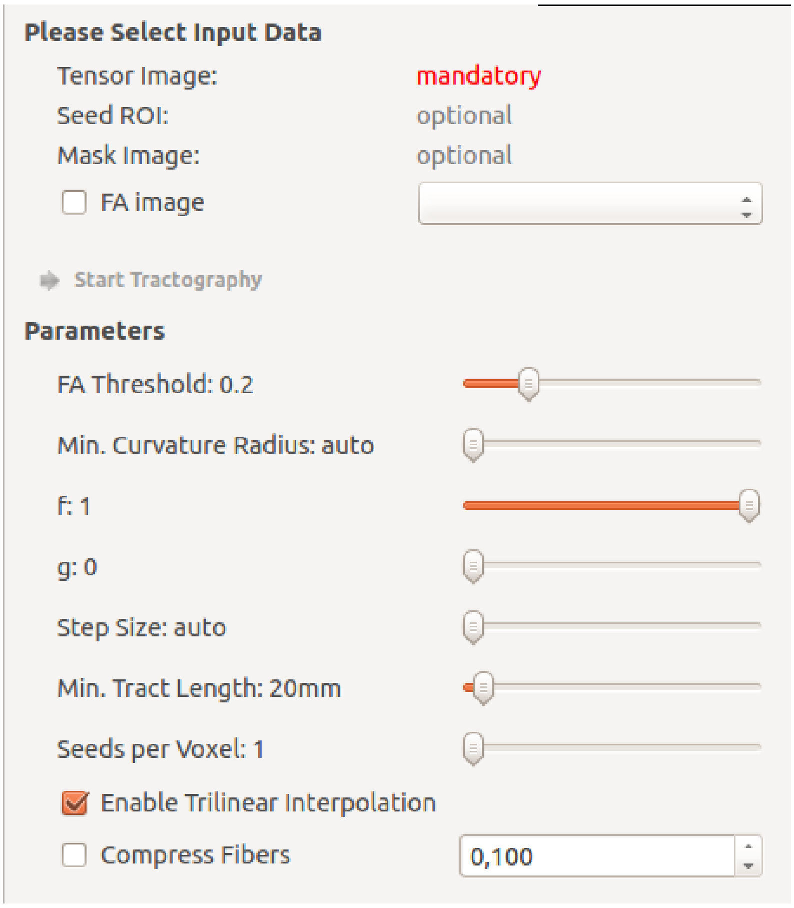
Figure 3.26: Output main fiber directions

3.18 Streamline Tracking View

This view provides the user interface for basic streamline fiber tractography on diffusion tensor images (single and multi-tensor tracking). FACT and TEND tracking methods are available.

Available sections:

- Input Data
- Input Parameters
- References



Please Select Input Data

Tensor Image: mandatory
Seed ROI: optional
Mask Image: optional
☐ FA image

Parameters

FA Threshold: 0.2
Min. Curvature Radius: auto
f: 1
g: 0
Step Size: auto
Min. Tract Length: 20mm
Seeds per Voxel: 1
☑ Enable Trilinear Interpolation
☐ Compress Fibers 0,100

Figure 3.27: Streamline Tracking View

3.18.1 Input Data

Mandatory Input:

- One or multiple DTI Image images selected in the datamanager.

Optional Input:

- FA image used to determine streamline termination. If no image is specified, the FA image is automatically calculated from the input tensor images. If multiple tensor images are used as input, it is recommended to provide such an FA image since the FA maps calculated from the individual input tensor images can not provide a suitable termination criterion.
- Binary mask used to define the seed voxels. If no seed mask is specified, the whole image volume is seeded.
- Binary mask used to constrain the generated streamlines. Streamlines can not leave the mask area.

3.18.2 Input Parameters

- FA Threshold: If the streamline reaches a position with an FA value lower than the speciefied threshold, it is not tracked any further.
- Min. Curvature Radius: If the streamline has a higher curvature than specified, it is not tracked any further. It is defined as the radius of the circle specified by three successive streamline positions.

- f and g values to balance between FACT [1] and TEND [2,3] tracking. For further information please refer to [2,3]
- Step Size: The algorithm proceeds along the streamline with a fixed stepsize. Default is $0.1 * \text{minSpacing}$.
- Min. Tract Length: Shorter fibers are discarded.
- Seeds per voxel: If set to 1, the seed is defined as the voxel center. If > 1 the seeds are distributet randomly inside the voxel.

By default the image values are not interpolated. Enable corresponding checkbox to use trilinear interpolation of the tensors as well as the FA values. Keep in mind that in the noninterpolated case, the TEND term is only applied once per voxel. In the interpolated case the TEND term is applied at each integration step which results in much higher curvatures and has to be compensated by an according choice of f and g.

Whole brain tractograms obtained with a small step size can contain billions of points. The tractograms can be compressed by removing points that do not really contribute to the fiber shape, such as many points on a straight line. An error threshold (in mm) can be defined to specify which points should be removed and which not.

3.18.3 References

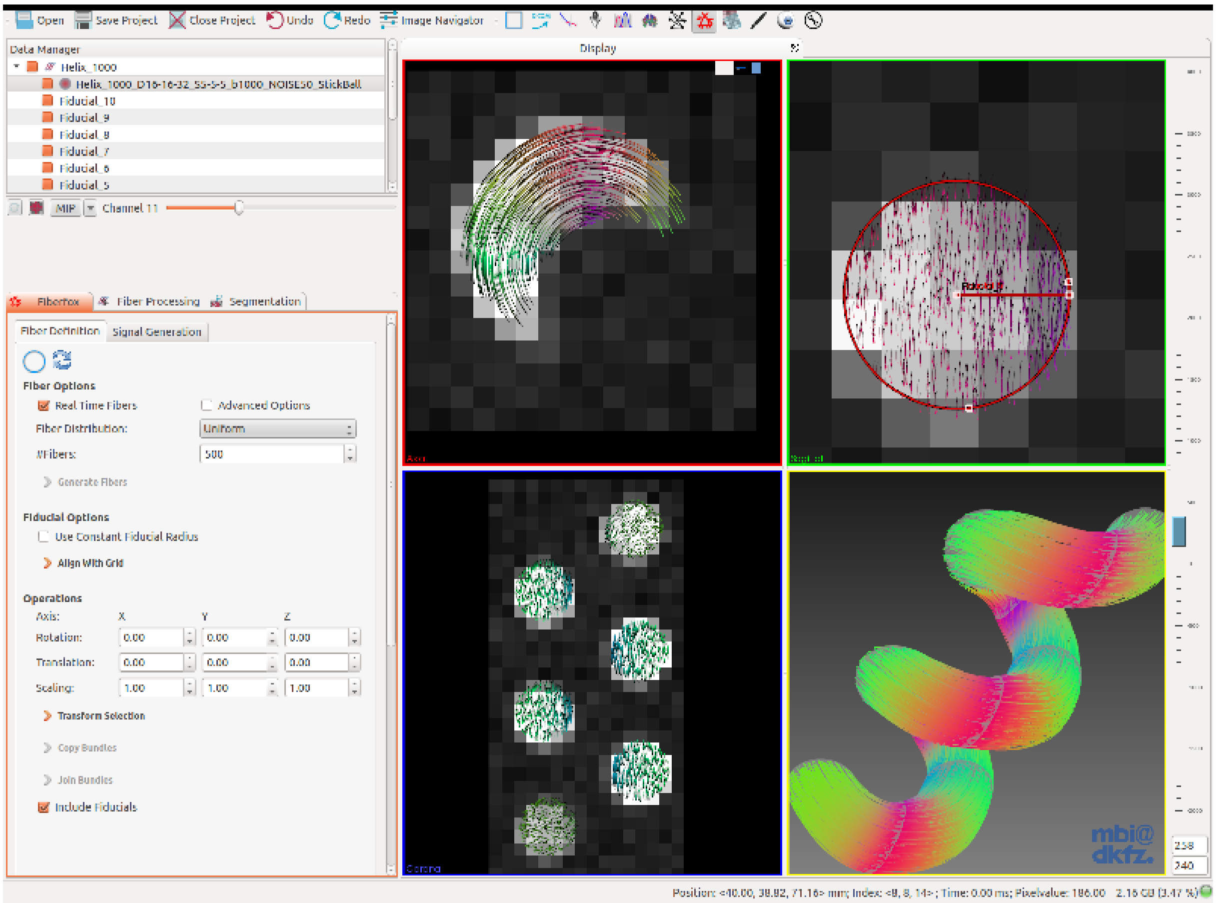
[1] Mori et al. Annals Neurology 1999
[2] Weinstein et al. Proceedings of IEEE Visualization 1999
[3] Lazar et al. Human Brain Mapping 2003

3.19 Fiberfox

This view provides the user interface for Fiberfox [1,2,3], an interactive simulation tool for defining artificial white matter fibers and generating corresponding diffusion weighted images. Arbitrary fiber configurations like bent, crossing, kissing, twisting, and fanning bundles can be intuitively defined by positioning only a few 3D waypoints to trigger the automated generation of synthetic fibers. From these fibers a diffusion weighted signal is simulated using a flexible combination of various diffusion models. It can be modified using specified acquisition settings such as gradient direction, b-value, signal-to-noise ratio, image size, and resolution. Additionally it enables the simulation of magnetic resonance artifacts including thermal noise, Gibbs ringing, N/2 ghosting, susceptibility distortions and motion artifacts. The employed parameters can be saved and loaded as xml file with the ending ".ffp" (Fiberfox parameters).

Available sections:

- Fiber Definition
- Signal Generation
- Known Issues
- References



Fiber Options

☑ Real Time Fibers ☐ Advanced Options

Fiber Distribution: Uniform

#Fibers: 500

Fiducial Options

☐ Use Constant Fiducial Radius

Operations

Axis:	X	Y	Z
Rotation:	0.00	0.00	0.00
Translation:	0.00	0.00	0.00
Scaling:	1.00	1.00	1.00

☑ Include Fiducials

Position: <40.00, 38.82, 71.16> mm; Index: <8, 8, 14> ; Time: 0.00 ms; Pixelvalue: 186.00 2.16 GB (3.47 %)

Figure 3.28: Fig. 1: Screenshot of the Fiberfox framework. The four render windows display an axial (top left), sagittal (top right) and coronal (bottom left) 2D cut as well as a 3D view of a synthetic fiber helix and the fiducials used to define its shape. In the 2D views the helix is superimposing the baseline volume of the corresponding diffusion weighted image. The sagittal render window shows a close-up view on one of the circular fiducials.

3.19.1 Fiber Definition

Fiber strands are defined simply by placing markers in a 3D image volume. The fibers are then interpolated between these fiducials.

Example:

- Chose an image volume to place the markers used to define the fiber pathway. If you don't have such an image available switch to the "Signal Generation" tab, define the size and spacing of the desired image and click "Generate Image". If no fiber bundle is selected, this will generate a dummy image that can be used to place the fiducials.
- Start placing fiducials at the desired positions to define the fiber pathway. To do that, click on the button with the circle pictogram, then click at the desired position and plane in the image volume and drag your mouse while keeping the button pressed to generate a circular shape. Adjust the shape using the control points (Fig. 2). The position of control point D introduces a twist of the fibers between two successive fiducials. The actual fiber generation is triggered automatically as soon as you place the second control point.
- In some cases the fibers are entangled in a way that can't be resolved by introducing an additional fiber twist. Fiberfox tries to avoid these situations, which arise from different normal orientations of succeeding fiducials, automatically. In rare cases this is not successful. Use the double-arrow button to flip the fiber positions of the selected fiducial in one dimension. Either the problem is resolved now or you can resolve it manually by adjusting the twist-control point.

- To create non elliptical fiber profile shapes switch to the Fiber Extraction View. This view provides tools to extract subesets of fibers from fiber bundles and enables to cut out arbitrary polygonal fiber shapes from existing bundles.

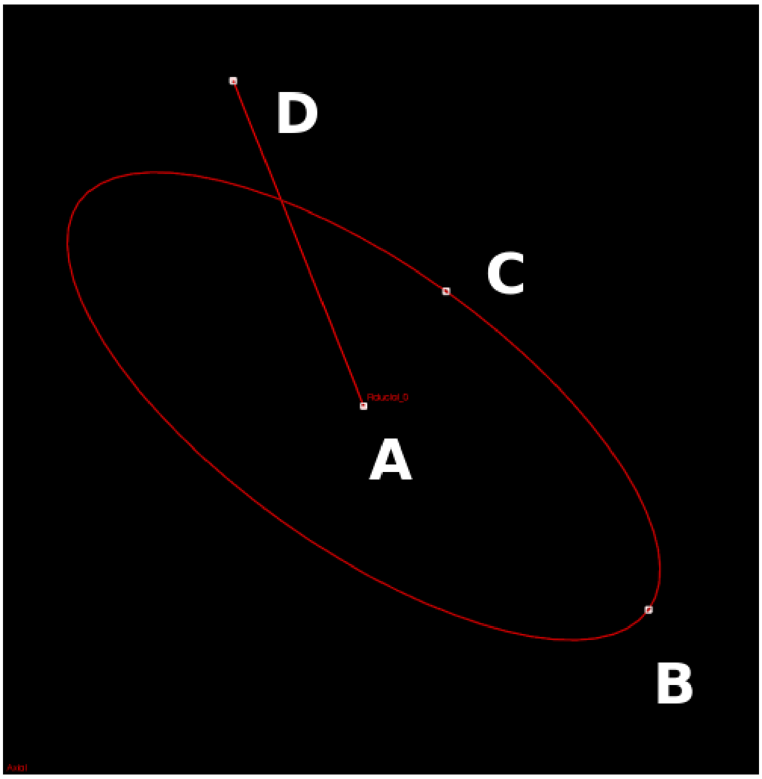


Figure 3.29: Fig. 2: Control points defining the actual shape of the fiducial. A specifies the fiducials position in space, B and C the two ellipse radii and D the twisting angle between two successive fiducials.

Fiber Options:

- **Real Time Fibers**: If checked, each parameter adjustment (fiducial position, number of fibers, ...) will be directly applied to the selected fiber bundle. If unchecked, the fibers will only be generated if the corresponding button "Generate Fibers" is clicked.
- **Advanced Options**: Show/hide advanced options
- **# Fibers**: Specifies the number of fibers that will be generated for the selected bundle.
- **Fiber Sampling**: Adjusts the distenace of the fiber sampling points (in mm). A higher sampling rate is needed if high curvatures are modeled.
- **Tension, Continuity, Bias**: Parameters controlling the shape of the splines interpolation the fiducials. See `Wikipedia` for details.

Fiducial Options:

- **Use Constant Fiducial Radius**: If checked, all fiducials are treated as circles with the same radius. The first fiducial of the bundle defines the radius of all other fiducials.
- **Align with grid**: Click to shift all fiducial center points to the next voxel center.

Operations:

- **Rotation**: Define the rotation of the selected fiber bundle around each axis (in degree).

- **Translation**: Define the translation of the selected fiber bundle along each axis (in mm).
- **Scaling**: Define a scaling factor for the selected fiber bundle in each dimension.
- **Transform Selection**: Apply specified rotation, translation and scaling to the selected Bundle/Fiducial
- **Copy Bundles**: Add copies of the selected fiber bundles to the datamanager.
- **Join Bundles**: Add new bundle to the datamanager that contains all fibers from the selected bundles.
- **Include Fiducials**: If checked, the specified transformation is also applied to the fiducials belonging to the selected fiber bundle and the fiducials are also copied.

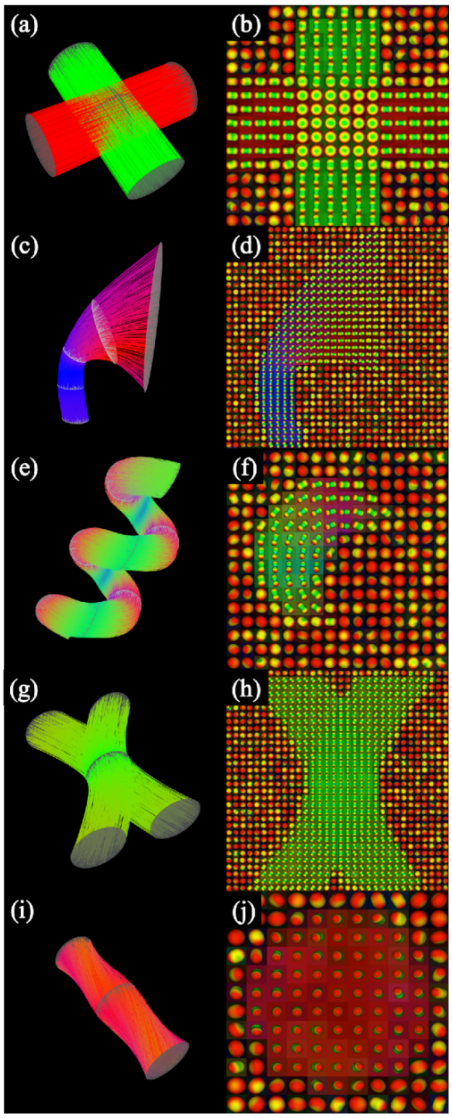


Figure 3.30: Fig. 3: Examples of artificial crossing (a,b), fanning (c,d), highly curved (e,f), kissing (g,h) and twisting (i,j) fibers as well as of the corresponding tensor images generated with Fiberfox.

3.19.2 Signal Generation

To generate an artificial signal from the input fibers we follow the concepts recently presented by Panagiotaki et al. in a review and taxonomy of different compartment models: a flexible model combining multiple compartments is used to simulate the anisotropic diffusion inside (intra-axonal compartment) and between axons (inter-axonal compartment), isotropic diffusion outside of the axons (extra-axonal compartment 1) and the restricted diffusion in other cell types (extra-axonal compartment 2) weighted according to their respective volume fraction.

A diffusion weighted image is generated from the fibers by selecting the according fiber bundle in the datamanager and clicking "Generate Image". If some other diffusion weighted image is selected together with the fiber bundle, Fiberfox directly uses the parameters of the selected image (size, spacing, gradient directions, b-values) for the signal generation process. Additionally a binary image can be selected that defines the tissue area. Voxels outside of this mask will contain no signal, only noise.

Basic Image Settings:

- **Image Dimensions**: Specifies actual image size (number of voxels in each dimension).
- **Image Spacing**: Specifies voxel size in mm. Beware that changing the voxel size also changes the signal strength, e.g. increasing the resolution from *2x2x2* mm to *1x1x1* mm decreases the signal obtained for each voxel by a factor 8.
- **Gradient Directions**: Number of gradients directions distributed equally over the half sphere. 10% baseline images are automatically added.
- **b-Value**: Diffusion weighting in s/mm^2. If an existing diffusion weighted image is used to set the basic parameters, the b-value is defined by the gradient direction magnitudes of this image, which also enables the use of multiple b-values.

Advanced Image Settings (activate checkbox "Advanced Options"):

- **Signal Scale**: Additional scaling factor for the signal in each voxel. The default value of 125 results in a maximum signal amplitude of 1000 for *2x2x2* mm voxels. Beware that changing this value without changing the noise variance results in a changed SNR. Adjustment of this value might be needed if the overall signal values are much too high or much too low (depends on a variety of factors like voxel size and relaxation times).
- **Echo Time *TE***: Time between the 90° excitation pulse and the first spin echo. Increasing this time results in a stronger *T2*-relaxation effect (Wikipedia).
- **Line Readout Time**: Time to read one line in k-space. Increasing this time results in a stronger $T2*$ effect which causes an attenuation of the higher frequencies in phase direction (here along y-axis) which again results in a blurring effect of sharp edges perpendicular to the phase direction.
- **T_{inhom} Relaxation**: Time constant specifying the signal decay due to magnetic field inhomogeneities (also called *T2'*). Together with the tissue specific relaxation time constant *T2* this defines the $T2*$ decay constant: $T2*=(T2\ T2')/(T2+T2')$
- **Fiber Radius (in µm)**: Used to calculate the volume fractions of the used compartments (fiber, water, etc.). If set to 0 (default) the fiber radius is set automatically so that the voxel containing the most fibers is filled completely. A realistic axon radius ranges from about 5 to 20 microns. Using the automatic estimation the resulting value might very well be much larger or smaller than this range.
- **Simulate Signal Relaxation**: If checked, the relaxation induced signal decay is simulated, other wise the parameters *TE*, Line Readout Time, T_{inhom}, and *T2* are ignored.
- **Disable Partial Volume Effects**: If checked, the actual volume fractions of the single compartments are ignored. A voxel will either be filled by the intra axonal compartment completely or will contain no fiber at all.
- **Output Volume Fractions**: Output a double image for each compartment. The voxel values correspond to the volume fraction of the respective compartment.

Compartment Settings:

The group-boxes "Intra-axonal Compartment", "Inter-axonal Compartment" and "Extra-axonal Compartments" allow the specification which model to use and the corresponding model parameters. Currently the following models are implemented:

- **Stick**: The “stick” model describes diffusion in an idealized cylinder with zero radius. Parameter: Diffusivity *d*
- **Zeppelin**: Cylindrically symmetric diffusion tensor. Parameters: Parallel diffusivity $d_{||}$ and perpendicular diffusivity *d*
- **Tensor**: Full diffusion tensor. Parameters: Parallel diffusivity $d_{||}$ and perpendicular diffusivity constants d_1 and d_2
- **Ball**: Isotropic compartment. Parameter: Diffusivity *d*
- **Astrosticks**: Consists of multiple stick models pointing in different directions. The single stick orientations can either be distributed equally over the sphere or are sampled randomly. The model represents signal coming from a type of glial cell called astrocytes, or populations of axons with arbitrary orientation. Parameters: randomization of the stick orientations and diffusivity of the sticks *d*.
- **Dot**: Isotropically restricted compartment. No parameter.
- **Prototype Signal**: The signal is not generated from a paranmetric model but a prototype signal is sampled from the selected diffusion-weighted image. Parameters: The number of prototype signals that are used for the signal generation (at each fiber position one is picked randomly) and the constraining diffusion parameters for a voxel signal to be included in the list. For a fiber signal one would for example probably select a high *FA* and for a CSF voxel a low *FA*.

For a detailed description of the single models, please refer to Panagiotaki *et al.* "Compartment models of the diffusion MR signal in brain white matter: A taxonomy and comparison". Additionally to the model parameters, each compartment has its own *T2* signal relaxation constant (in *ms*). This constant is not relevant if the prototype signal model is used, since in this case signal relaxation is disabled.

Noise and Artifacts:

- **Noise**: Add Rician or Chi-Square distributed noise with the specified variance to the signal.
- **Spikes**: Add signal spikes to the k-space signal resulting in stripe artifacts across the corresponding image slice.
- **Aliasing**: Aliasing artifacts occur if the FOV in phase direction is smaller than the imaged object. The parameter defines the percentage by which the FOV is shrunk.
- **N/2 Ghosts**: Specify the offset between successive lines in k-space. This offset causes ghost images in distance N/2 in phase direction due to the alternating EPI readout directions.
- **Distortions**: Simulate distortions due to magnetic field inhomogeneities. This is achieved by adding an additional phase during the readout process. The input is a frequency map specifying the inhomogeneities. The "Fieldmap Generator" view provides an interface to generate simple artificial frequency maps.
- **Motion Artifacts**: To simulate motion artifacts, the fiber configuration is moved between the signal simulation of the individual gradient volumes. The motion can be performed randomly, where the parameters are used to define the +/- maximum of the corresponding motion, or linearly, where the parameters define the maximum rotation/translation around/along the corresponding axis at the and of the simulated acquisition.
- **Eddy Currents**: EXPERIMENTAL! This feature is currently being tested and might not yet behave as expected!
- **Gibbs Ringing**: Ringing artifacts occurring on edges in the image due to the frequency low-pass filtering caused by the limited size of the k-space.

3.19.3 Known Issues

- If fiducials are created in one of the marginal slices of the underlying image, a position change of the fiducial can be observed upon selection/deselection. If the fiducial is created in any other slice this bug does not occur.
- If a scaling factor is applied to the selcted fiber bundle, the corresponding fiducials are not scaled accordingly.

- In some cases the automatic update of the selected fiber bundle is not triggered even if "Real Time Fibers" is checked, e.g. if a fiducial is deleted. If this happens on can always force an update by pressing the "Generate Fibers" button.

If any other issues or feature requests arises during the use of Fiberfox, please don't hesitate to send us an e-mail or directly report the issue in our bugtracker: `http://bugs.mitk.org/`

3.19.4 References

[1] Peter F. Neher, Frederik B. Laun, Bram Stieltjes, and Klaus H. Fritzsche: Fiberfox: Facilitating the creation of realistic white matter software phantoms, Magn Reson Med, DOI: 10.1002/mrm.25045.

[2] Peter F. Neher, Frederik B. Laun, Bram Stieltjes, and Klaus H. Fritzsche: Fiberfox: An extensible system for generating realistic white matter software phantoms, MICCAI CDMRI Workshop, Nagoya; 09/2013

[3] Peter F. Neher, Bram Stieltjes, Frederik B. Laun, Hans-Peter Meinzer, and Klaus H. Fritzsche: Fiberfox: Fiberfox: A novel tool to generate software phantoms of complex fiber geometries, ISMRM, Salt Lake City; 04/2013

3.20 Fieldmap Generator View

This view allows the creation of artificial frequency maps used by Fiberfox to introduce distortions into diffusion weighted images. The generated images can contain a linear frequency gradient and/or multiple 3D gaussian shaped field inhomogeneities.

Example:

- Select a reference image with the combo box. The generated fieldmap will feature the same geometry as the selected image.

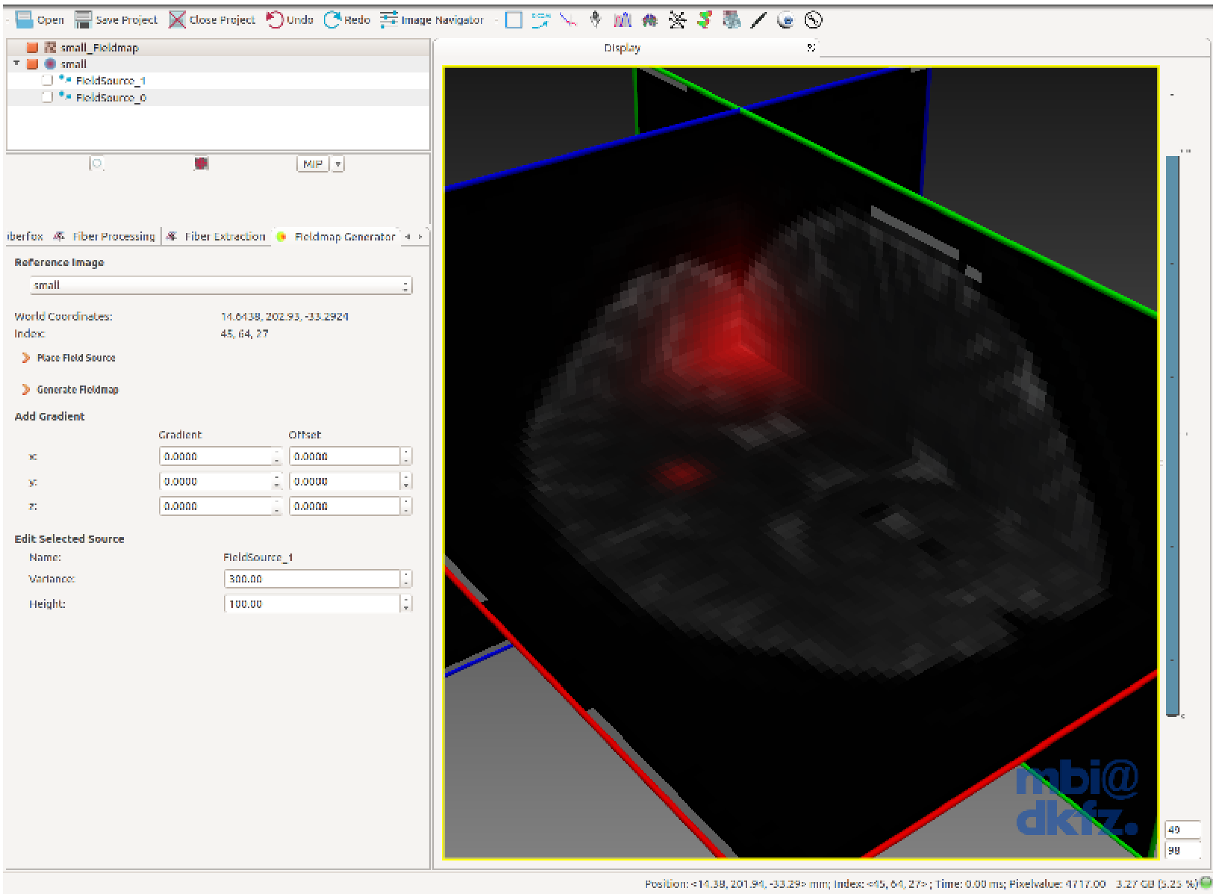
- Move the crosshair to the any position in the image and click "Place Field Source".

- A position marker will appear in the render windows and in the datamanager, which indicates the position of a 3D gaussian field distortion that will be introduced upon clicking "Generate Fieldmap".

- The strength and variance of the placed sources can be modified by selecting the corresponding data node in the data manager and adjusting the parameters in the lower part of the view (below "Edit Selected Source").

- To introduce an (additional) linear frequency gradient, specify the gradient below "Add Gradient".

- To finally generate the fieldmap, press "Generate Fieldmap".



Reference Image

small

World Coordinates: 14.6438, 202.93, -33.2924

Index: 45, 64, 27

Add Gradient

	Gradient	Offset
x:	0.0000	0.0000
y:	0.0000	0.0000
z:	0.0000	0.0000

Edit Selected Source

Name: FieldSource_1

Variance: 300.00

Height: 100.00

49

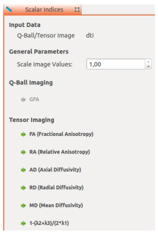
98

Position: <14.38, 201.94, -33.29> mm; Index: <45, 64, 27> ; Time: 0.00 ms; Pixelvalue: 4717.00 3.27 GB (5.25 %)

Figure 3.31: The Fieldmap Generator View. The render window shows a diffusion weighted image of the brain superimposed by a frequency map with two 3D gaussian field inhomogeneities (red).

3.21 Scalar Indices

The scalar indices view allows the derivation of different scalar anisotropy measures for the reconstructed tensors (Fractional Anisotropy, Relative Anisotropy, Axial Diffusivity, Radial Diffusivity) or q-balls (Generalized Fractional Anisotropy).



Input Data

Q-Ball/Tensor Image dti

General Parameters

Scale Image Values: 1,00

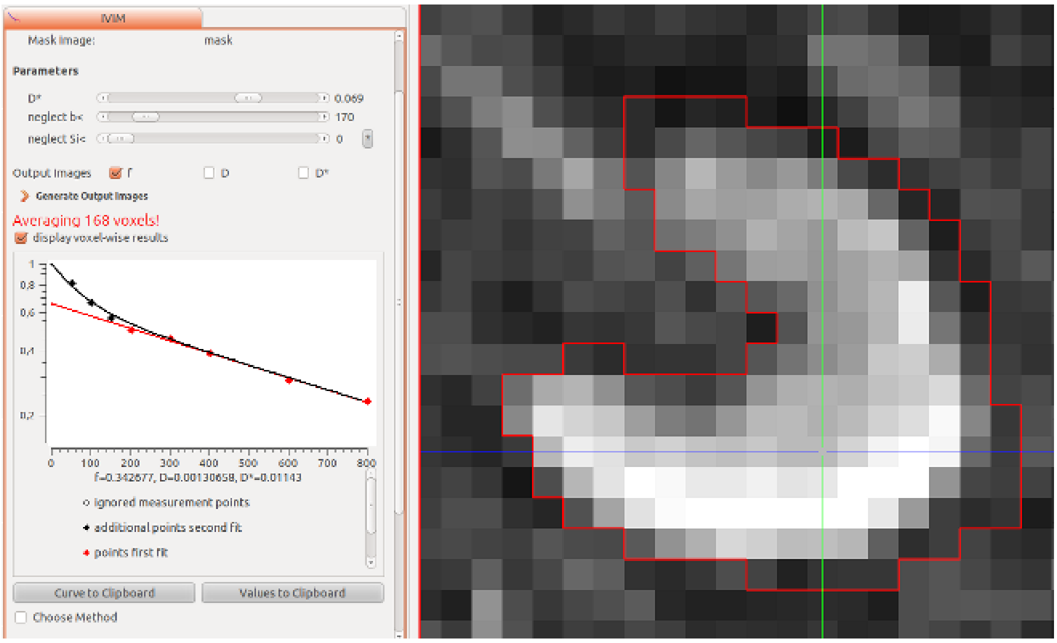
Q-Ball Imaging

Tensor Imaging

Figure 3.32: Anisotropy quantification

3.22 Intra-voxel incoherent motion estimation (IVIM)

The required input for the "Intra-voxel incoherent motion estimation" (IVIM) is a diffusion weighted image (.dwi or .hdwi) that was acquired with several different b-values.



Mask Image: mask

Parameters

D* 0.069

neglect b< 170

neglect Si< 0

Output Images ☑ f ☐ D ☐ D*

Averaging 168 voxels!

☑ display voxel-wise results

f=0.342677, D=0.00130658, D*=0.01143

Figure 3.33: The IVIM View

Once an input image is selected in the datamanager, the IVIM view allows for interactive exploration of the dataset (click around in the image and watch the estimated parameters in the figure of the view) as well as generation of f-, D-, and D$*$-maps (activate the checkmarks and press "Generate Output Images").

The "neglect b$<$" threshold allows you to ignore b-values smaller then a threshold for the initial fit of f and D. D$*$ is then estimated using all measurements.

The exact values of the current fit are always given in the legend underneath the figure.

3.22.1 Region of interest analysis

Create region of interest: To create a new segmentatin, open the "quantification" perspective, select the tab "↩ Segmentation", and create a segmentation of the structure of interest. Alternatively, of course, you may also load a binary image from file or generate your segmentation in any other possible way.

IVIM in region of interset: Go back to the "IVIM" perspective and select both, the diffusion image and the segmentation (holding the CTRL key). A red message should appear "Averaging N voxels".

3.22.2 Export

All model parameters and corresponding curves can be exported to clipboard using the buttons underneath the figure.

3.22.3 Advanced Settings

Advanced users, that know what they are doing, can change the method for the model-fit under "Advanced Settings" on the very bottom of the view. 3-param fit, linear fit of f/D, and fix D$*$ are among the options.

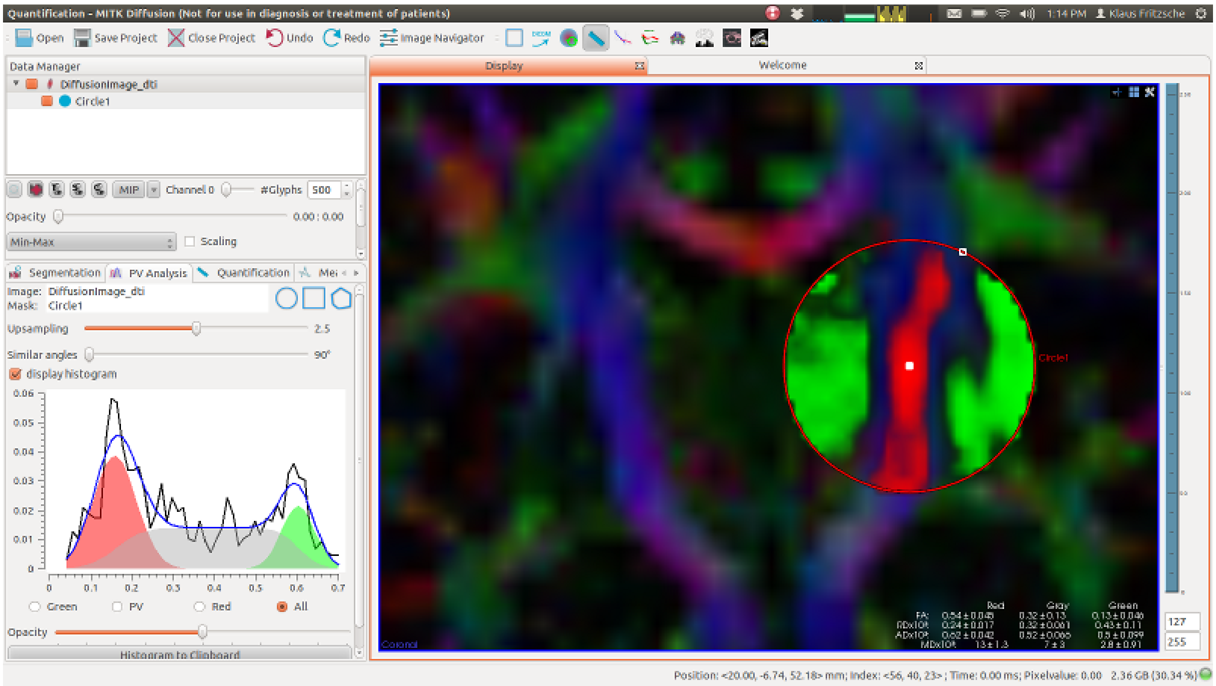
3.22.4 Suggested Readings

Toward an optimal distribution of b values for intravoxel incoherent motion imaging. Lemke A, Stieltjes B, Schad LR, Laun FB. Magn Reson Imaging. 2011 Jul;29(6):766-76. Epub 2011 May 5. PMID: 21549538

Differentiation of pancreas carcinoma from healthy pancreatic tissue using multiple b-values: comparison of apparent diffusion coefficient and intravoxel incoherent motion derived parameters. Lemke A, Laun FB, Klauss M, Re TJ, Simon D, Delorme S, Schad LR, Stieltjes B. Invest Radiol. 2009 Dec;44(12):769-75. PMID: 19838121

3.23 Partial Volume Analysis

The "Partial Volume Analysis" view can be found in the "Quantification" perspective. It allows for robust quantification of diffusion or other scalar measures in the presents of two classes (e.g. fiber vs. non-fiber) and partial volume between them. The algorithm estimates a probabilistic segmentation of the three classes and returns a weighted average of the measure of interest within the each class.



Quantification - MITK Diffusion (Not for use in diagnosis or treatment of patients)

Image: DiffusionImage_dti
Mask: Circle1

Upsampling 2.5

Similar angles 90°

☑ display histogram

○ Green ○ PV ○ Red ● All

	Red	Gray	Green
FA:	0.54 ± 0.045	0.32 ± 0.13	0.13 ± 0.046
RDx10^3:	0.24 ± 0.017	0.32 ± 0.061	0.43 ± 0.11
ADx10^3:	0.62 ± 0.042	0.52 ± 0.066	0.5 ± 0.099
MDx10^3:	13 ± 1.3	7 ± 3	2.8 ± 0.91

Position: <20.00, -6.74, 52.18> mm; Index: <56, 40, 23> ; Time: 0.00 ms; Pixelvalue: 0.00 2.36 GB (30.34 %)

Figure 3.34: The Partial Volume Analysis View

3.23.1 Export

All measures are automatically written to the clipboard once the estimation is updated. For scalar images, the mean and the variance of the gaussian selected in the radio-box above is copied out. If 'All' is selected, then all means and variances are carried out as a tab-separated list. For tensor images, the values for all scalar measures (FA, MD, RD and AD) are carried out to the clipboard.

The histogram export is provided by the button underneath the histogram. The values can be pasted to excel or any text editor.

3.23.2 Advanced Settings

Are not recommended for use yet.

3.23.3 Suggested Readings

Diffusion tensor imaging in primary brain tumors: reproducible quantitative analysis of corpus callosum infiltration and contralateral involvement using a probabilistic mixture model. Stieltjes B, Schlüter M, Didinger B, Weber MA, Hahn HK, Parzer P, Rexilius J, Konrad-Verse O, Peitgen HO, Essig M. Neuroimage. 2006 Jun;31(2):531-42. Epub 2006 Feb 14. PMID: 16478665

3.24 The TBSS View

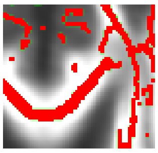


Figure 3.35: Icon of the TBSS View

3.24.1 Summary

This view can be used to locally explore data resulting from preprocessing with the TBSS view of FSL

This document will tell you how to use this view, but it is assumed that you already know how to use MITK in general and how to work with the TBSS view of FSL.

If you encounter problems using the view, please have a look at the Troubleshooting page.

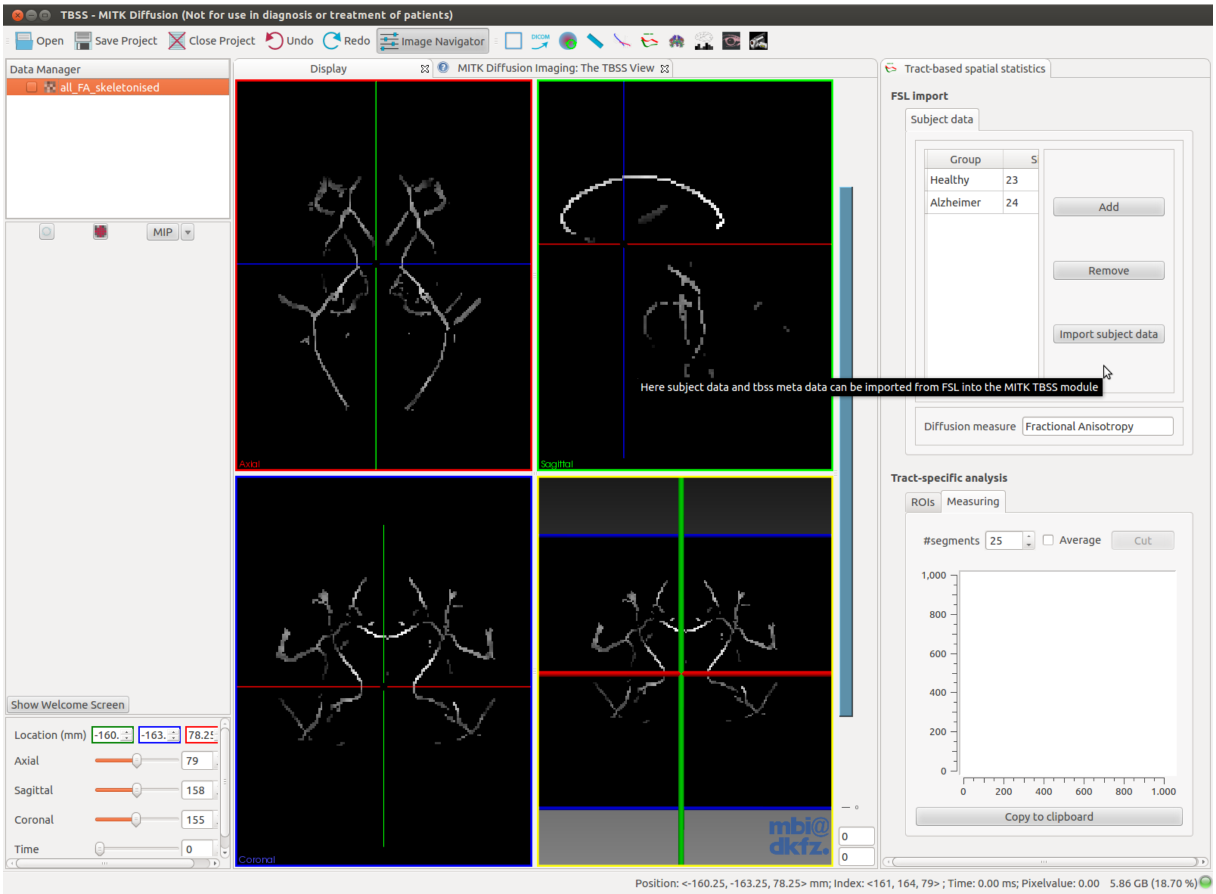
3.24.2 Overview

This view is currently under development and as such the interface as well as the capabilities are likely to change significantly between different versions.

This documentation describes the features of this current version.

3.24.3 FSL Import

The FSL import allows to import data that has been preprocessed by FSL. FSL creates output images that typically have names like all_FA_skeletonized.nii.gz that are 4-dimensional images that contain registered images of all subjects. By loading this 4D image into the datamanager and listing the groups with the correct number of subjects, in the order of occurrence in the 4D image, in the TBSS-View using the Add button and clicking the import subject data a TBSS file is created that contains all the information needed for tract analysis. The diffusion measure of the image can be set as well.



TBSS - MITK Diffusion (Not for use in diagnosis or treatment of patients)

Group	S
Healthy	23
Alzheimer	24

Here subject data and tbss meta data can be imported from FSL into the MITK TBSS module

Diffusion measure: Fractional Anisotropy

#segments: 25

Location (mm): -160. -163. 78.2!

Axial: 79

Sagittal: 158

Coronal: 155

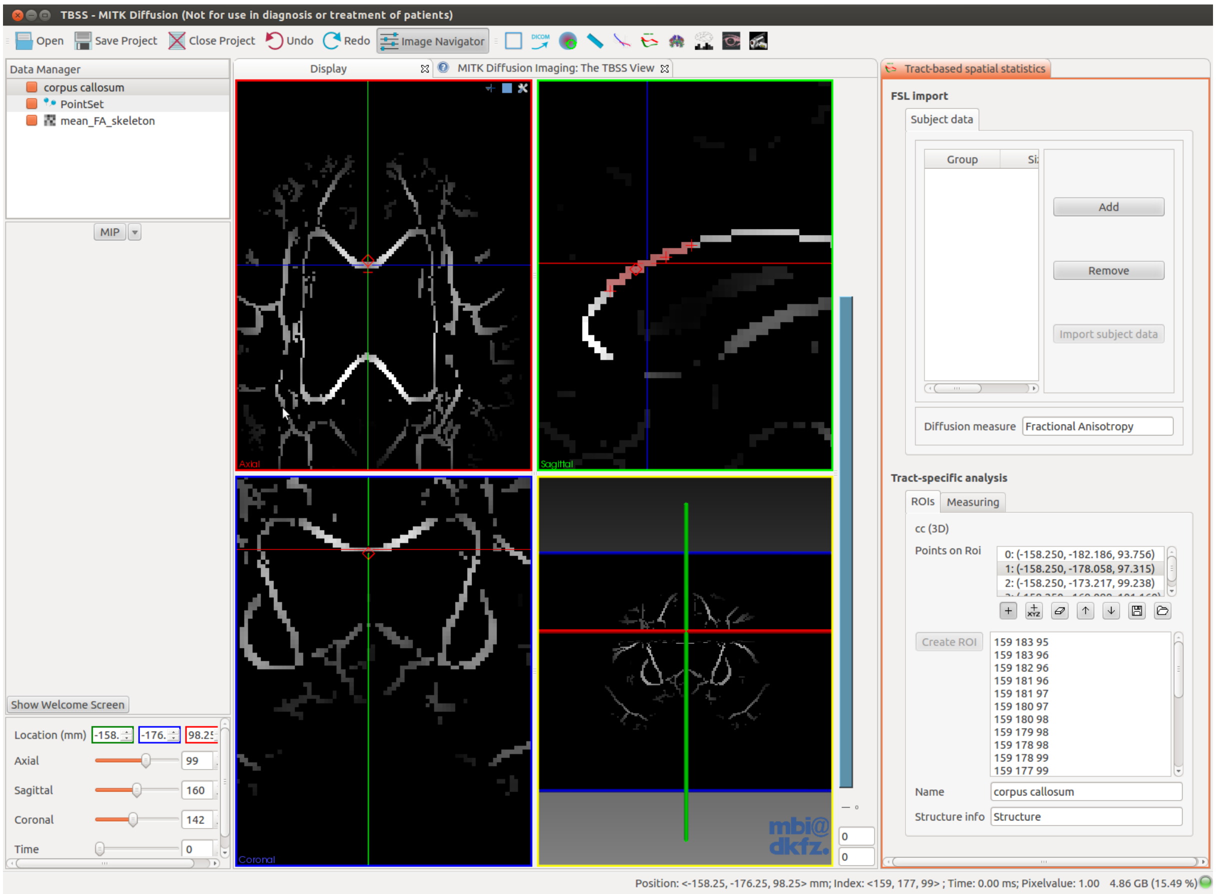
Time: 0

Position: <-160.25, -163.25, 78.25> mm; Index: <161, 164, 79> ; Time: 0.00 ms; Pixelvalue: 0.00 5.86 GB (18.70 %)

Figure 3.36: FSL Import

3.24.4 Regions of Interest (ROIs)

To create a ROI the mean FA skeleton (typically called mean_FA_skeleton.nii.gz) that is created by FSL should be loaded in to the datamanager and selected. By using the Pointlistwidget points should be set on the skeleton (make sure to select points with relatively high FA values). Points are set by first selecting the button with the '+' and than shift-leftclick in the image. When the correct image is selected in the datamanager the Create ROI button is enabled. Clicking this will create a region of interest that passes through the previously selected points. The roi appears in the datamanager. Before doing so, the name of the roi and the information on the structure on which the ROI lies can be set. This will be saved as extra information in the roi-image. Before the ROI is calculated, a pop-up window will ask the user to provide a threshold value. This should be the same threshold that was previously used in FSL to create a binary mask of the FA skeleton. When this is not done correctly, the region of interest will possible contain zero-valued voxels.



Location (mm) -158. -176. 98.25

Axial 99

Sagittal 160

Coronal 142

Time 0

Diffusion measure: Fractional Anisotropy

cc (3D)

Points on Roi

0: (-158.250, -182.186, 93.756)
1: (-158.250, -178.058, 97.315)
2: (-158.250, -173.217, 99.238)

```
159 183 95
159 183 96
159 182 96
159 181 96
159 181 97
159 180 97
159 180 98
159 179 98
159 178 98
159 178 99
159 177 99
```

Name: corpus callosum

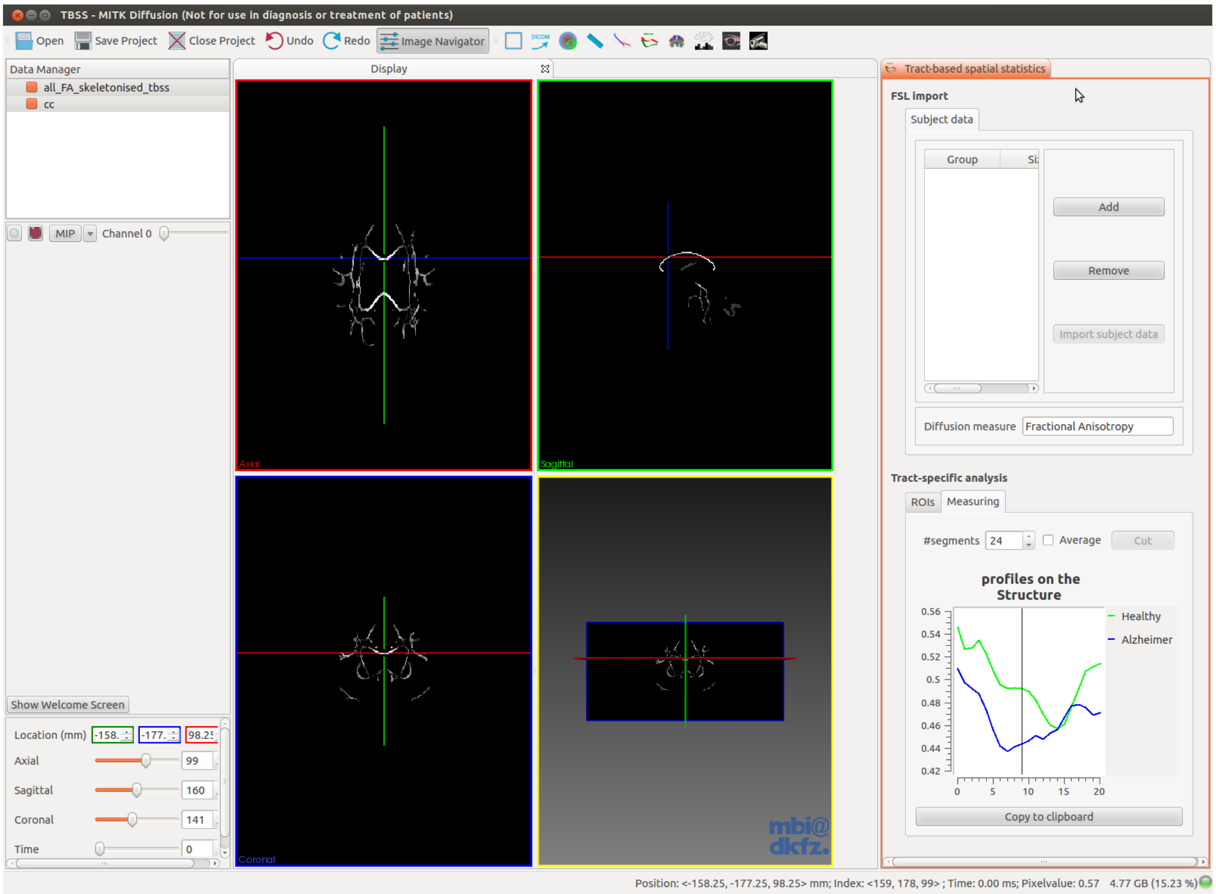
Structure info: Structure

Position: <-158.25, -176.25, 98.25> mm; Index: <159, 177, 99> ; Time: 0.00 ms; Pixelvalue: 1.00 4.86 GB (15.49 %)

Figure 3.37: Regions of Interest (ROIs)

3.24.5 QmitkTractbasedSpatialStatisticsUserManualProfiles

y selecting a tbss image with group information and a region of interest image (as was created in a previous stap). A profile plot is drawn in the plot canvas. By clicking in the graph the crosshairs jump to the corresponding location in the image.



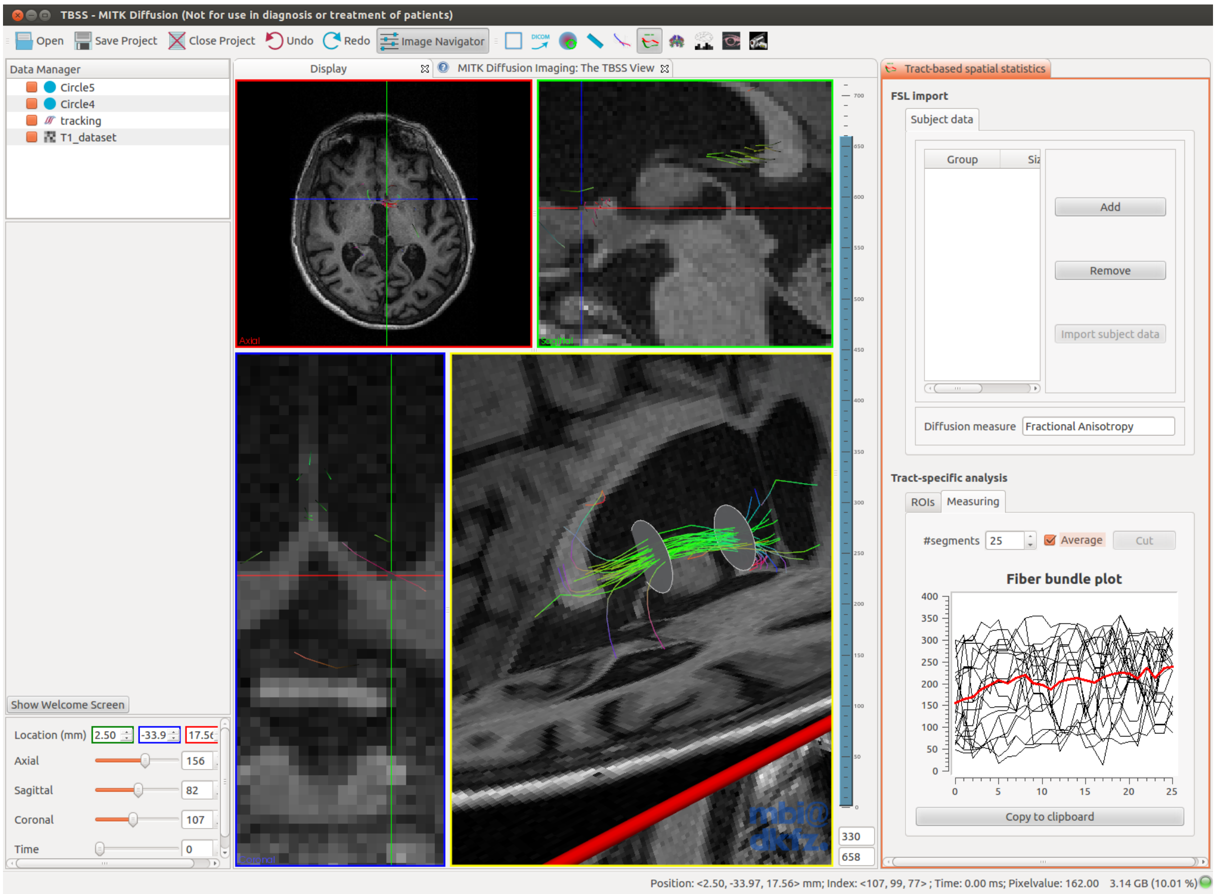
Location (mm) -158. -177. 98.25
Axial 99
Sagittal 160
Coronal 141
Time 0
Diffusion measure Fractional Anisotropy
#segments 24

Position: <-158.25, -177.25, 98.25> mm; Index: <159, 178, 99> ; Time: 0.00 ms; Pixelvalue: 0.57 4.77 GB (15.23 %)

Figure 3.38: Profile plots

3.24.6 QmitkTractbasedSpatialStatisticsUserManualFiberPlotting

It is also possible to use fiber bundles to plot values in images. However, unlike TBSS this only works on 3D volumes (one subject only). To plot the values of a 3D volume using fiber tracking results, we need, fiber data and two planar circles as Regions of Interest that define the region that must be plotted. For every fiber that passes through both ROIs the values at the corresponding positions in the volume are plotted. Every fiber is resampled to the number of segments specified in the GUI. The average value can also be plotted. The subset of the fibers between the ROIs can be cut out by pressing the Cut button.



Location (mm) 2.50 -33.9 17.56
Axial 156
Sagittal 82
Coronal 107
Time 0

Diffusion measure Fractional Anisotropy

#segments 25 ☑ Average

330
658

Position: <2.50, -33.97, 17.56> mm; Index: <107, 99, 77> ; Time: 0.00 ms; Pixelvalue: 162.00 3.14 GB (10.01 %)

Figure 3.39: Plot of fiber tracts

3.24.7 Troubleshooting

Please report to the MITK mailing list. See http://www.mitk.org/wiki/Mailinglist on how to do this.

3.24.8 References

1. S.M. Smith, M. Jenkinson, H. Johansen-Berg, D. Rueckert, T.E. Nichols, C.E. Mackay, K.E. Watkins, O. Ciccarelli, M.Z. Cader, P.M. Matthews, and T.E.J. Behrens. Tract-based spatial statistics: Voxelwise analysis of multi-subject diffusion data. NeuroImage, 31:1487-1505, 2006.

2. S.M. Smith, M. Jenkinson, M.W. Woolrich, C.F. Beckmann, T.E.J. Behrens, H. Johansen-Berg, P.R. Bannister, M. De Luca, I. Drobnjak, D.E. Flitney, R. Niazy, J. Saunders, J. Vickers, Y. Zhang, N. De Stefano, J.M. Brady, and P.M. Matthews. Advances in functional and structural MR image analysis and implementation as FSL. NeuroImage, 23(S1):208-219, 2004.

3.25 The Connectomics Perspective

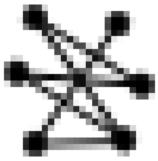


Figure 3.40: Icon of the Connectomics Perspective

The connectomics perspective is a collection of views which provide functionality for the work with brain connectivity networks. Currently there exist the following views:

The Connectomics Network Data View provides network generation either from data or synthetically.

The Connectomics Network Operations View provides functionalies to operate and process on networks and other data.

The Connectomics Statistics View provides statistical measures for networks.

The Random Parcellation View provides a method to randomly parcellate a segmentation using different criteria

3.25.1 Network Rendering Customization

The rendering of the connectomics networks can be customized by changing the associated properties using the property list. A selection of possible options are:

- Connectomics.Rendering.Edges.Filtering - Only render edges above a certain threshold
- Connectomics.Rendering.Edges.Gradient.Parameter - Color the edges according to certain parameters
- Connectomics.Rendering.Edges.Radius.Parameter - Change the radius of the edges according to certain parameters
- Connectomics.Rendering.Nodes.Filtering - Only render nodes above a certain threshold
- Connectomics.Rendering.Nodes.Gradient.Parameter - Color the nodes according to certain parameters
- Connectomics.Rendering.Nodes.Radius.Parameter - Change the radius of the nodes according to certain parameters
- Connectomics.Rendering.Scheme - Switch between the MITK rendering scheme using above properties and the very fast, but less customizable rendering scheme for VTK graphs

3.25.2 Troubleshooting

No known problems.

All other problems.
Please report to the MITK mailing list. See `http://www.mitk.org/wiki/Mailinglist` on how to do this.

3.25.3 The Connectomics Network Data View

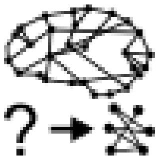
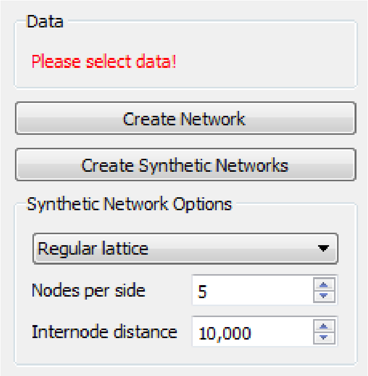


Figure 3.41: Icon of the Connectomics Network Data View

This view can be used to create a network from a parcellation and a fiber image as well as to create synthetic networks.



Please select data!

Synthetic Network Options: Regular lattice

Nodes per side: 5

Internode distance: 10,000

Figure 3.42: The user interface

To create a network select first a parcellation of the brain (e.g. as provided by `freesurfer`) by CTRL+Leftclick and secondly a fiber image (as created using a tractography view). Then click on the "Create Network" button.

- "Use label of end position of fibers" will create a network containing a node for every label a fiber ends in
- "Extrapolate label" will avoid creating nodes using FreeSurfer white matter labels and instead extrapolate in which grey matter label the fiber would end

Additionally you have the option to create artificial networks, for testing purposes. Currently choices are:

- A regular lattice, where each node is placed in a cubic pattern and only connected to its neighbours
- A heterogenic sphere, where one node is placed in the center and connected to all nodes on the shell
- A random network, where nodes are randomly placed on a spherical shell and randomly connected

3.25.4 The Connectomics Network Operations View

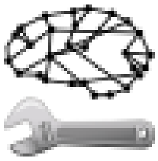


Figure 3.43: Icon of the Connectomics Network Operations View

This view can be used modify networks and related data.

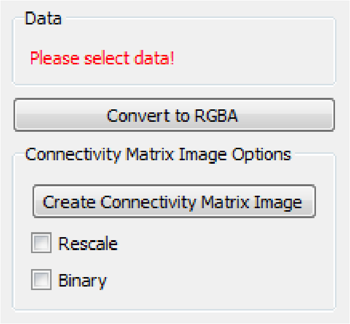


Figure 3.44: The user interface

Select a parcellation and press "Convert to RGBA" to create a RGBA image. By doing this conversion it is much easier to discern the different parcels. Furthermore MITK supports 3D visualization of an RGBA image.

Select a network and press "Create Connectivity Matrix Image" to create a 2D image of the connectivity matrix. By default the weight of a connection is used as grey value. Using the "Rescale" option will rescale the weights so highest one is 255. Using the "Binary" option will result in a binary connectivity matrix.

3.25.5 The Connectomics Statistics View

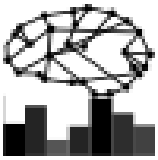
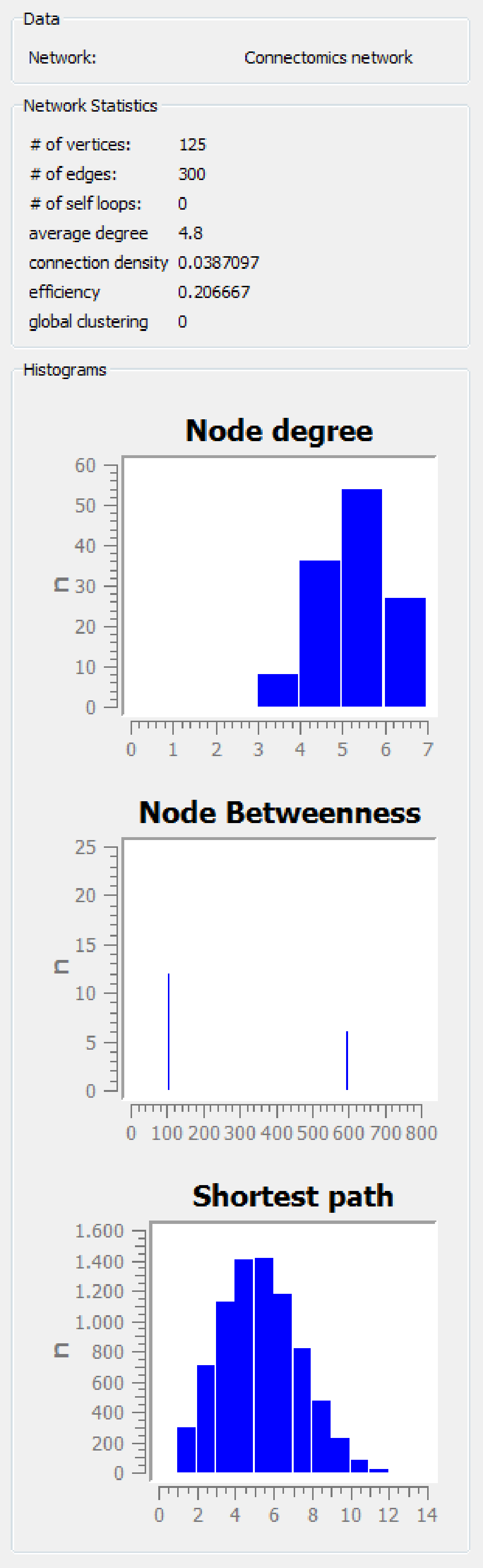


Figure 3.45: Icon of the Connectomics Statistics View

This view can be used to show statistical analysis of a network.



Data

Network: Connectomics network

Network Statistics

# of vertices:	125
# of edges:	300
# of self loops:	0
average degree	4.8
connection density	0.0387097
efficiency	0.206667
global clustering	0

Histograms

Node degree

Node Betweenness

Shortest path

Figure 3.46: The user interface

To calculate network statistics select a network in the datamanager. At this time the following statistics are calculated for the entire network:

- The number of vertices in the network
- The number of edges in the network
- The number of edges which have the same vertex as beginning and end point
- The average degree of the nodes in the network
- The connection density the network (the number of edges divided by the number of possible edges)
- The unweighted efficiency of the network (1 divided by average path length, this is zero for disconnected graphs)
- The global clustering

Furthermore some statistics are calculated on a per node basis and displayed as histograms:

- The degree of each node
- The (unweighted) betweenness centrality of each node
- The spread of shortest paths between each pair of nodes (For disconnected graphs the shortest paths with infinite length are omitted for readability)

3.25.6 The Random Parcellation View

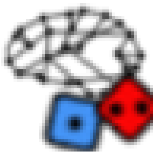


Figure 3.47: Icon of the Random Parcellation View

3.25.6.1 Summary

This view is used to create a random parcellation of a segmented image. It was designed to parcellate grey matter to acquire random nodes for network analysis. This document describes how to use this view.

3.25.6.2 Overview

When parcellating an image you can either do a parcellation and use the generated parcellation. In which case it will have as many parcellation as seed points were specified. Instead you can choose to merge some of the original parcels untill either none below a threshold size remain, or until a specified minimum number of parcels has been reached. A further option is to "Just Merge Small Parcels". This will result in only neighbouring parcels below a threshold size being merged. Using this option may result in final parcels below the given threshold size if they are not adjacent to another parcel below the threshold size.

In all there are four possibilities to get a random parcellation

- Without merging
- Merging with respect to the number of nodes afterwards
- Merging with respect to the number of voxels of the smallest parcel
- Merging with respect to the number of voxels of the smallest parcel, where just small parcels are merged

3.25.6.3 Usage

First you have to get a segmentation of the target structure. You can use the Segmentation View to create one. Select it in the Data Manager. Now you can choose which of the possible ways of parcellation you would like. See Overview for these possibilities. Start the parcellation process using the "Select Random Nodes" button. You will get a new image in the Data Manager. Every voxel of a given parcel will have the same value (0 for background).

3.25.6.4 Methods

This Plug-In uses a kind of region growing algorithm. Random seed voxels are set on the segmentation and each is an own region at the beginning. Then voxels are added iteratively to each smallest region, first just the 6-connected ones and later the 26-connected ones and isolated voxels. Specific constraints have to be fulfilled before a voxel is added to a region. The algorithm stops when all voxels are contained in the regions. For merging it is checked which regions are neighbors of the smallest parcel and then a suitable one, according to a cost function, is merged to this parcel. When the merging algorithm stops depends on which stop-mechanism is chosen.

Part III

Other plugins

This chapter describes plugins not directly related to diffusion imaging.

Chapter 4

The Command Line Modules View

Figure 4.1: Icon of the Command Line Modules View

4.1 Contribution

This plugin was developed at the Centre For Medical Image Computing (CMIC), part of University College London (UCL) and contributed back to the MITK community with thanks.

4.2 Introduction

This view provides the facility to run third party command line programs, and load the data back into the Data Manager for immediate visualisation. All that is required is that the command line application can be called with an argument of –xml and respond with a valid XML description of the necessary parameters, and currently, that if the program requires images, they must be NifTI images. This view can then generate a Graphical User Interface (GUI) dynamically from the XML to enable the user to interact with the command line application. This provides an easy to use, and potentially very flexible way to integrate almost any third party, medical imaging, command line application.

As a high level introduction, this view performs the following steps:

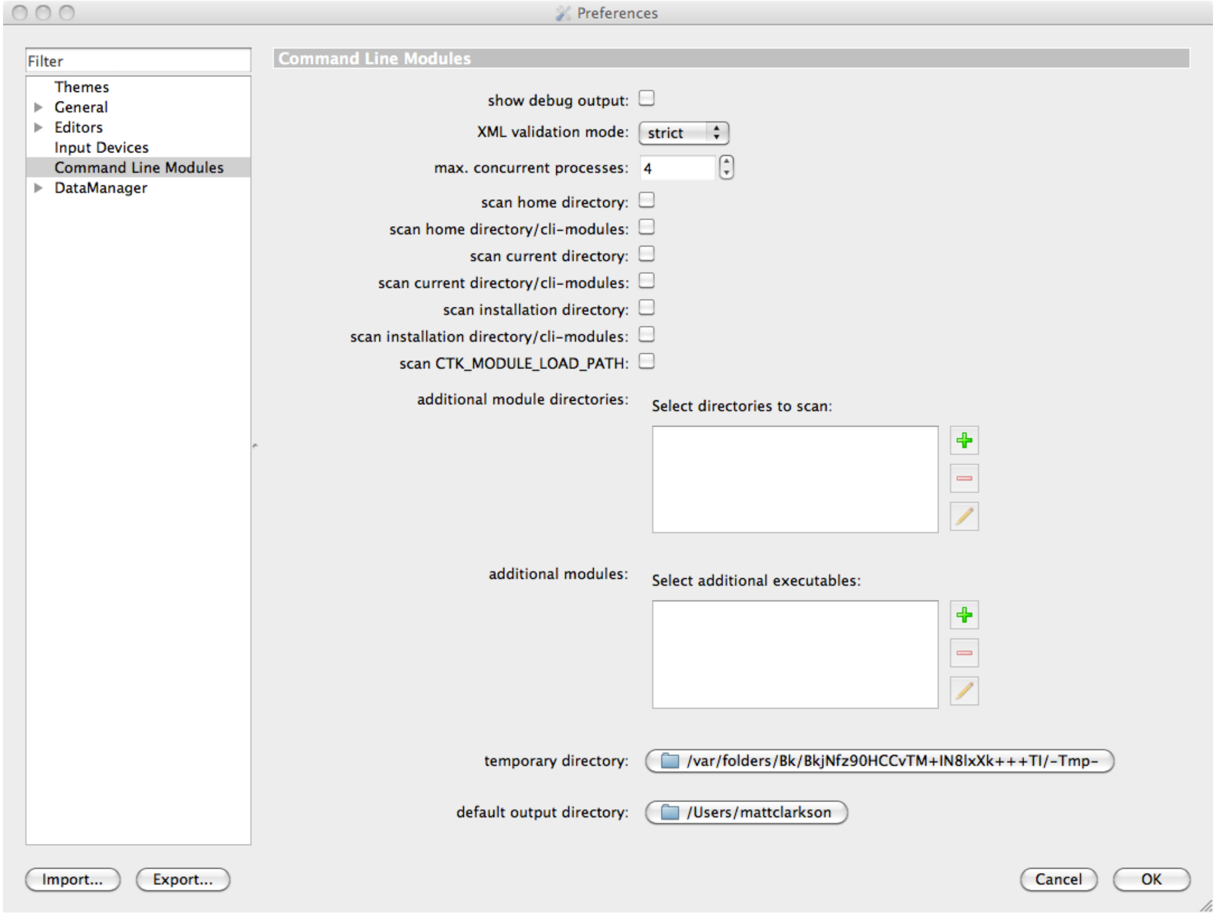
- The view searches for available programs to run, and for each valid module, stores the XML document describing the interface, and populates a searchable list of available programs.
- When a program is selected, the GUI is generated.
- The user can then set the necessary parameters and run the program.
- Multiple programs can be launched in succession and run simultaneously, and where available on the host platform, the user can pause, resume or cancel running jobs and see console output for each job.

As a consequence of the very flexible nature of this plugin, these instructions can only describe how to launch command line modules in a general sense. The examples shown have been constructed by downloading the latest version (subversion commit 329) of the NiftyReg package, available here, and described further here. NiftyReg provides valid XML descriptors to enable the integration of the NiftyReg affine (RegAladin) and and non-rigid

(RegF3D) image registration algorithms, as well as utility programs to resample an image, and calculate a Jacobian image. These same XML descriptors work within `Slicer` and `MITK` based applications.

4.3 Preferences

The first time that the Command Line Modules View is launched, it is advisable to set the user preferences for the view. Please refer to Figure 1.



show debug output: ☐
XML validation mode: strict
max. concurrent processes: 4
scan home directory: ☐
scan home directory/cli-modules: ☐
scan current directory: ☐
scan current directory/cli-modules: ☐
scan installation directory: ☐
scan installation directory/cli-modules: ☐
scan CTK_MODULE_LOAD_PATH: ☐
additional module directories: Select directories to scan:
additional modules: Select additional executables:
temporary directory: /var/folders/Bk/BkjNfz90HCCvTM+IN8lxXk+++TI/-Tmp-
default output directory: /Users/mattclarkson

Figure 4.2: Figure 1. The Command Line Modules Preferences Page

Each of these preferences is now explained in some detail.

- show debug output: If checked will output more messages to the console for debugging purposes.
- XML validation mode: The user may select a different mode for XML validation. If this is changed, the application will need to be restarted. There are 3 modes available. If the user selects "strict" mode, the XML schema produced by the command line application must exactly conform to `this definition`. For "none", there will be no validation. For "weak" validation, the application will report errors, but try to carry on and load as many modules as possible. The XML validation errors are available as tool-tips on the tab widget when the module is launched. Many third party modules included with Slicer currently have incorrect XML (typically, mis-ordered XML tags), and so the "weak" or "none" mode may assist in loading them. By default the "strict" mode is chosen so that only valid modules are loaded.
- max concurrent processes: Sets the maximum number of concurrent jobs that can be run via this interface. The default is 4. When the maximum number is reached, the green "Run" button is disabled until a job finishes.

The next 7 preferences are to control where the view will search for valid command line programs. By default these are off as the searching process can take a long time and slow down the startup time of the GUI. The options provided are:

- scan home directory: Scan the users home directory. (See QDir::homePath().)
- scan home directory/cli-modules: Scans the sub-directory called cli-modules under the users home directory.
- scan current directory: Scan the current working directory. (See QDir::homePath().)
- scan current directory/cli-modules: Scans the sub-directory called cli-modules under the current working directory.
- scan installation directory: This is the directory where the actual application is stored.
- scan installation directory/cli-modules: Scans the sub-directory called cli-modules under the application installation directory.
- scan CTK_MODULE_LOAD_PATH: Scans the directory or list of directories defined by the environment variable CTK_MODULE_LOAD_PATH. A list is colon separated on Linux/Mac, and semi-colon separated on Windows.

In most cases, it is suggested that the user will leave these options unchecked, as the user can also specify custom directories, and even cherry-pick specific command line programs to load. Figure 2 shows a selection box that enables the user to specify custom directories to scan, and Figure 3. shows a selection box that enables the user to select specific modules. Picking specific directories, and specific executables will most likely make the application quicker to launch.

." _linebr \image latex cmdlinemodules_PreferencesAdditionalDirectories.png "Figure 2. The User can specify specific directories to scan"." width=7.90cm ." _linebr \image latex cmdlinemodules_PreferencesAdditionalModules.↩
png "Figure 3. The User can specify specific command line programs to load"." width=7.92cm

These directory and file selection boxes enable directories or files to be added, removed and updated in a similar fashion.

The user must make sure that the list of files selected in the "additional modules" section are not already contained within the directories specified in the "additional module directories" section.

In addition, the preferences page provides:

- temporary directory: Images stored in the DataManager are first written to a temporary folder as `Nifti` images before being passed to each command line program. This temporary directory will default to a platform specific temporary folder, but the user may select their preferred choice of temporary workspace.

4.4 Usage

When the view is launched, a simple interface is presented, as shown in Figure 4.

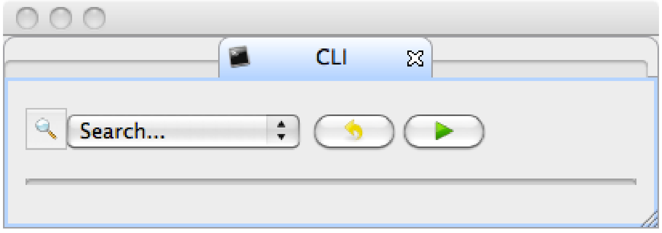


Figure 4.3: Figure 4. The initial interface, with no command line programs available.

In this example, all the above check-box preferences were off, and the "additional module directories" was empty, and the "additional modules" list was empty so no command line applications were found. The "Search" box displays zero entries, and there is nothing to search.

If the available search paths contain programs that are compatible (i.e. runnable) with this view, the name of the programs are displayed in the "Search" box in a nested menu, shown in Figure 5.

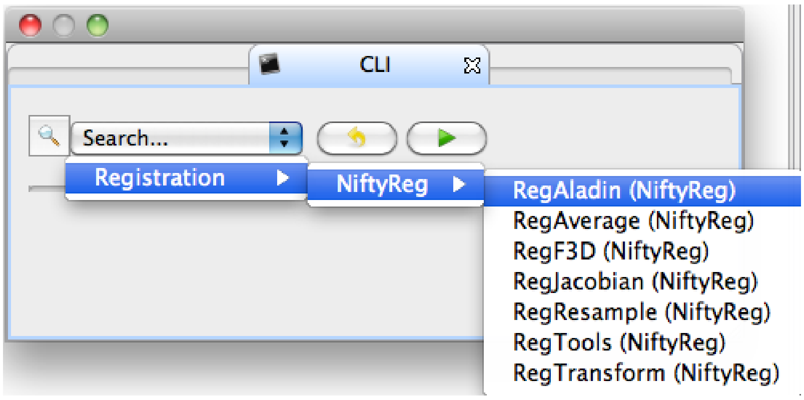


Figure 4.4: Figure 5. When valid paths are set, and programs are discovered, the menu is recalculated to show available programs.

When a program is selected, the relevant interface is displayed, by default as collapsed group boxes to save space. Each section can be individually expanded if necessary to see the parameters.

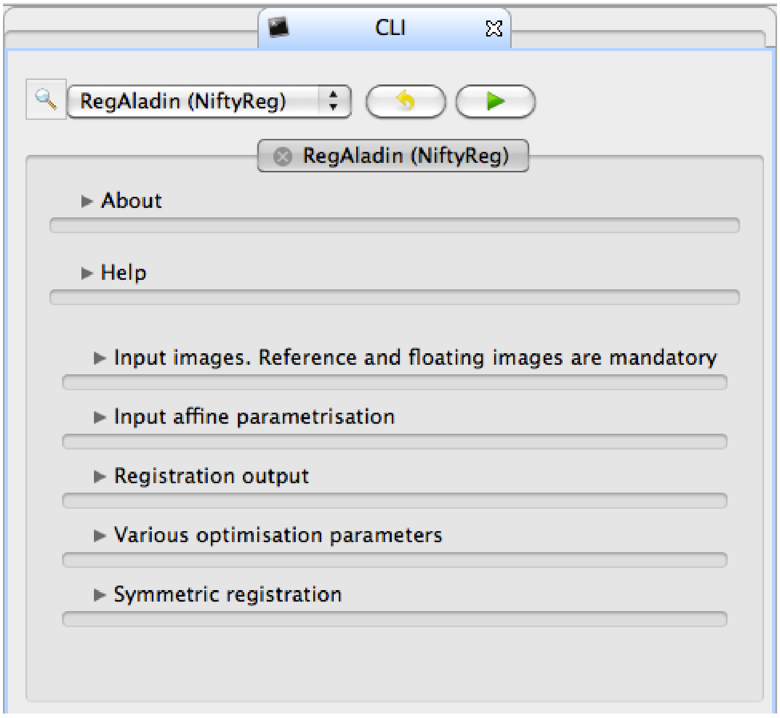


Figure 4.5: Figure 6. An example program, showing parameters for NiftyReg's program RegAladin.

In this example, the parameters are displayed for `NiftyReg` produced at `UCL`, and more specifically for the affine registration program called RegAladin. The interface can contain a wide variety of controls. If a parameter for a command line program is an input image, then the widget displayed is linked to the DataManager, so that as new images are loaded, the correct image can be easily selected from the combo box.

At this stage, multiple tabs can be opened, with one tab for each command line program. Figure 7 shows 2 tabs, for the RegAladin and RegF3D programs.

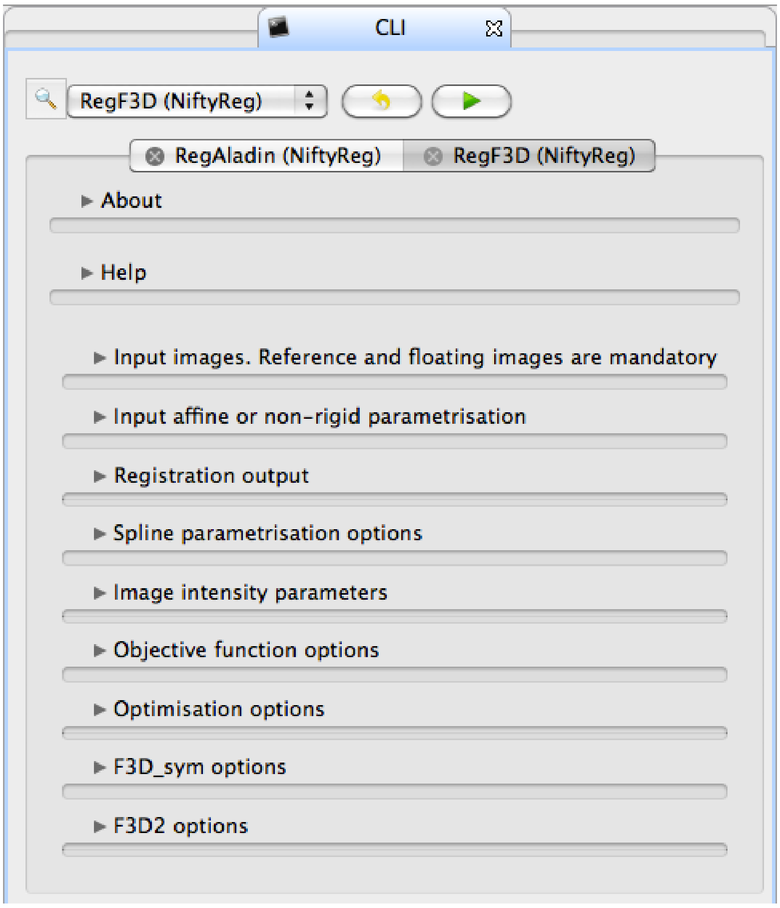


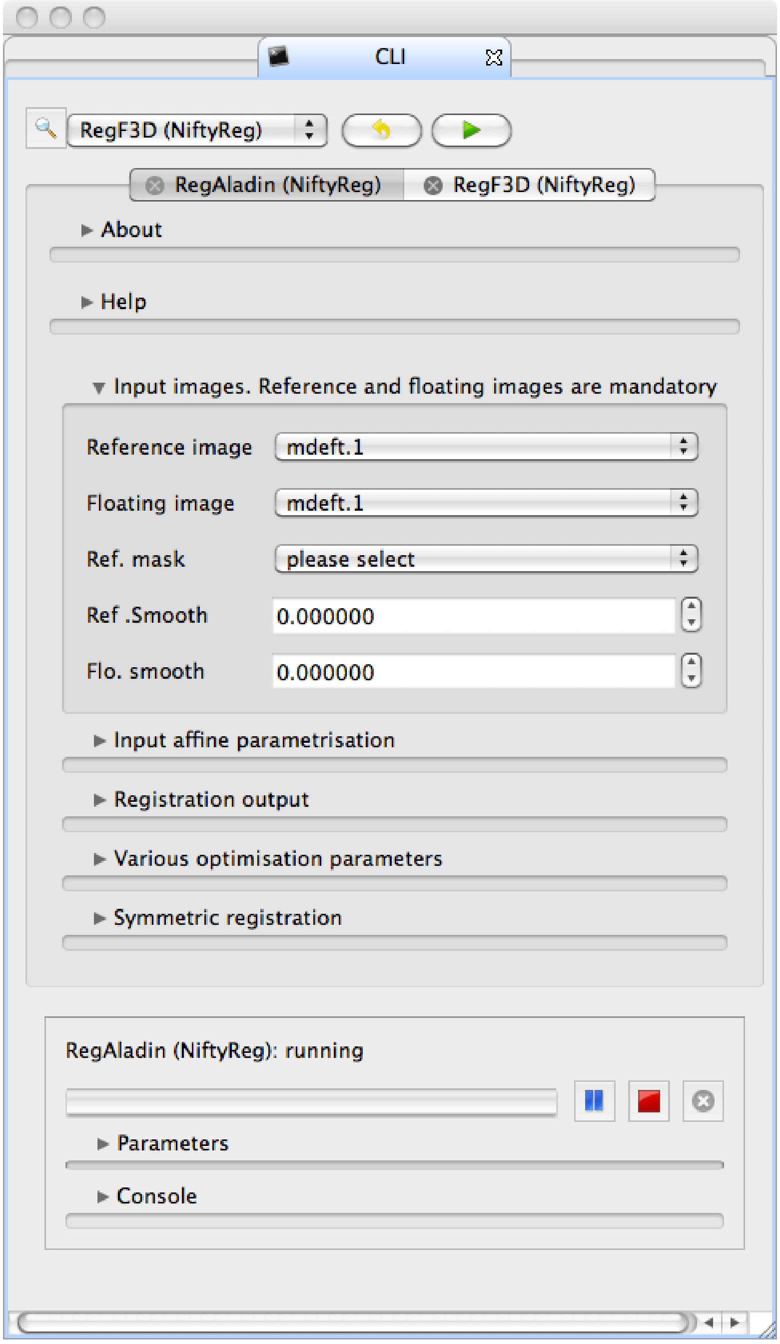
Figure 4.6: Figure 7. Multiple tabs can be opened, one for each command line program.

The main view provides some simple controls:

- Green arrow: Launch (run) the command line executable of the currently selected tab.

- Yellow undo arrow: Resets the GUI controls of the currently selected tab to default values, if and only if the original XML specified a default value.

At this stage, nothing has been launched. When the user hits the green arrow button, a job is launched. Each running job is shown as a new progress reporting widget under the main tabbed widget, as shown in Figure 8.



Reference image: mdeft.1
Floating image: mdeft.1
Ref. mask: please select
Ref .Smooth: 0.000000
Flo. smooth: 0.000000

RegAladin (NiftyReg): running

Figure 4.7: Figure 8. Multiple programs can be run, each with individual controls and console output.

The controls for each running job are:

- Blue pause button: If supported on the host platform, this button will be enabled and can be toggled off (pause) or on (resume).
- Red square: If supported on the host platform, this button will kill the command line program.
- Black cross: Will remove the progress reporting widget from the GUI.

When the user hits the green arrow in the main view:

- The currently selected tab is designated the "current" job, and contains the "current" set of parameters.

- A new progress reporting widget is created.
- The current parameters are copied to the progress reporting widget. In Figure 8. a parameters section is visible, and by default is collapsed, as they are simply for referring back to.
- All the output for the command line program is shown in the console widget, with a separate console for each job.
- Each new progress reporting widget is simply stacked vertically (newest is top-most), and it is up to the user to delete them when they are finished.

It is easy to run multiple jobs. The green button simply launches the job corresponding to the current tab repeatedly. It is up to the user to make sure that any output file names are changed between successive invocations of the same command line module to avoid overwritting output data.

In addition, each set of parameters contains an "About" section containing details of the contributors, the licence and acknowledgements and also a "Help" section containing a description and a link to any on-line documentation.

These documentation features are provided by the developers of the third party plugin, and not by the host program. If information is missing, the user must contact the third party developers.

4.5 Technical Notes

From a technical perspective, the Command Line Modules View is a simple view, harnessing the power of the CTK command line modules framework. For technical information see:

- The doxygen generated `manual page`.
- The `wiki page`.

and obviously the CTK code base.

Chapter 5

The Logging Plugin

Logging.png

Figure 5.1: Icon of the Logging Plugin

This plug-in records all logging output of events and progress as specified in the source code with time of occurence, level of importance (Info, Warning, Error, Fatal, Debug), the message given and where it happens. The logging starts once the plug-is started. A screenshot of the provided Logging view is shown next.

LogView.png

Figure 5.2: Screenshot of the Logging Module

There are different features available in the view. The filter text field allows for searching all log events containing a certain substring. Using the button "Copy to clipboard" on the bottom right you can copy the current content of the logging view to your clipboard. This enables you to insert the logging information to any text processing application.

You can also show more information on every logging message by activating the two checkboxes. In the simple view, leaving both checkboxes unchecked, you'll see logging messages and logging levels. A brief description of the logging levels can be found in the logging concept documentation. The checkbox "Category" adds a column for the category. The checkbox "Show Advanced Field" shows method, filename and linenumber where the logging message was emitted as well as the running time of the application. The next figure shows all information which can be shown in the Logging Module.

`LogViewExplain.png`

Figure 5.3: Details on the Vizualized Logging Information

Chapter 6

The DataManager

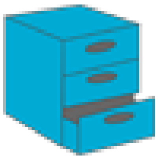


Figure 6.1: Icon of the Data Manager

6.1 Introduction

The Datamanager is the central componenent to manage medical data like images, surfaces, etc.. After loading one or more data into the Datamanager the data are shown in the four-view window, the so called Standard View. The user can now start working on the data by just clicking into the standard view or by using the MITK-modules such as "Segmentation" or "Basic Image Processing".

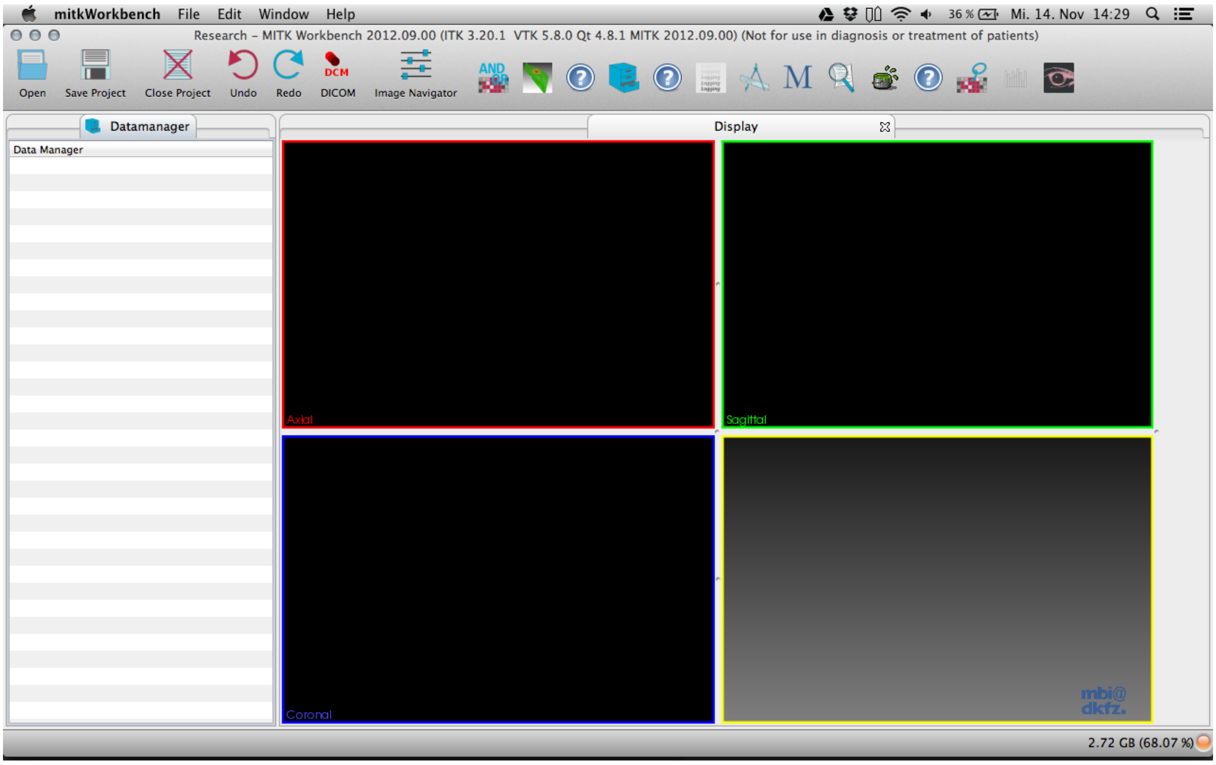


Figure 6.2: How MITK looks when started

6.2 Loading Data

There are three ways of loading data into the Datamanager as so called Data-Elements.

The user can just drag and drop data into the Datamanager or directly into one of the four parts of the Standard View. He can as well use the Open-Button in the right upper corner. Or he can use the standard "File->Open"-Dialog on the top.

A lot of file-formats can be loaded into MITK, for example

- 2D-images/3D-volumes with or without several timesteps (∗.dcm, ∗.ima, ∗.pic, ...)
- Surfaces (∗.stl, ∗.vtk, ...)
- Pointsets (∗.mps)
- ...

The user can also load a series of 2D images (e.g. image001.png, image002.png ...) to a MITK 3D volume. To do this, just drag and drop one of those 2D data files into the Datamanager by holding the ALT key.

After loading one or more data into the Datamanager they appear as Data-Elements in a sorted list inside the Datamanager. Data-Elements can also be sorted hierarchically as a parent-child-relation. For example after using the Segmentation-Module on Data-Element1 the result is created as Data-Element2, which is a child of Data-Element1 (see Screenshot1). The order can be changed by drag and drop.

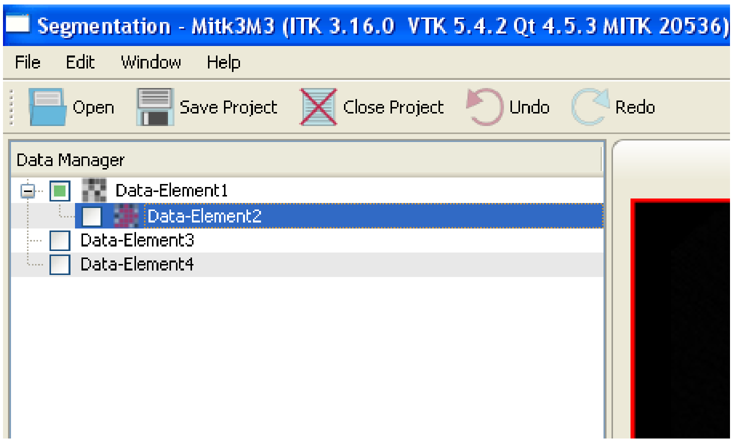


Figure 6.3: Screenshot1

The listed Data-Elements are shown in the standard view. Here the user can scale or rotate the medical objects or he can change the cutting planes of the object by just using the mouse inside this view.

6.3 Saving Data

There are two ways of saving data from the Datamanger. The user can either save the whole project with all Data-Elements by clicking on "File"->"Save Project" or he can save single Data-Elements by right-clicking->"↩
Save", directly on a Data-Element. When saving the whole project, the sorting of Data-Elements is saved as well. By contrast the sorting is lost, when saving a single Data-Element.

6.4 Working with the Datamanager

6.4.1 List of Data-Elements

The Data-Elements are listed in the Datamanager. As described above the elements can be sorted hierarchically as a parent-child-relation. For example after using the Segmentation-Module on Data-Element1 the result is created as Data-Element2, which is a child of Data-Element1 (see Screenshot1). By drag and drop the sorting of Data-↩
Elements and their hierarchical relation can be changed.

6.4.2 Visibility of Data-Elements

By default all loaded Data-Elements are visible in the standard view. The visibility can be changed by right-clicking on the Data-Element and then choosing "Toogle visibility". The box in front of the Data-Element in the Datamanager shows the visibility. A green-filled box means a visible Data-Element, an empty box means an invisible Data-Element (see Screenshot1).

6.4.3 Representation of Data-Elements

There are different types of representations how to show the Data-Element inside the standard view. By right-clicking on the Data-Element all options are listed (see Screenshot2 and Screenshot 3).

- An arbitrary color can be chosen
- The opacity can be changed with a slide control

- In case of images a texture interpolation can be switched on or off. The texture interpolation smoothes the image, so that no single pixels are visible anymore.
- In case of surfaces the surface representation can be changed between points, wireframe or surface.
- Global reinit updates all windows to contain all the current data. Reinit updates a single data item fits the windows to contain this data item.

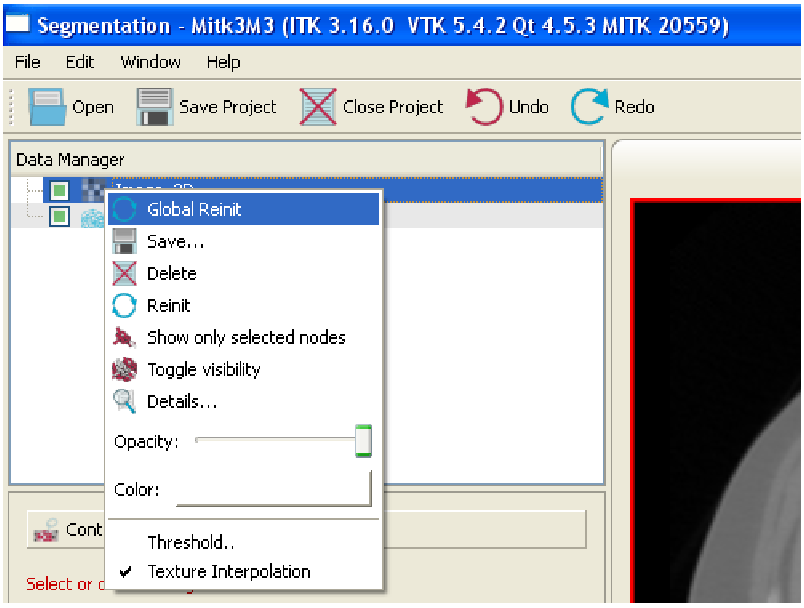


Figure 6.4: Screenshot2: Properties for images

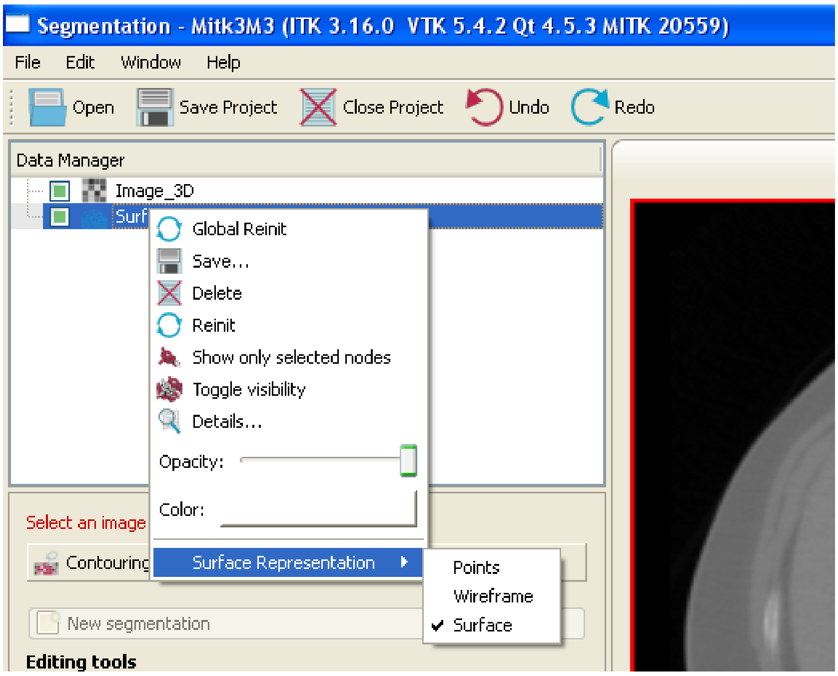
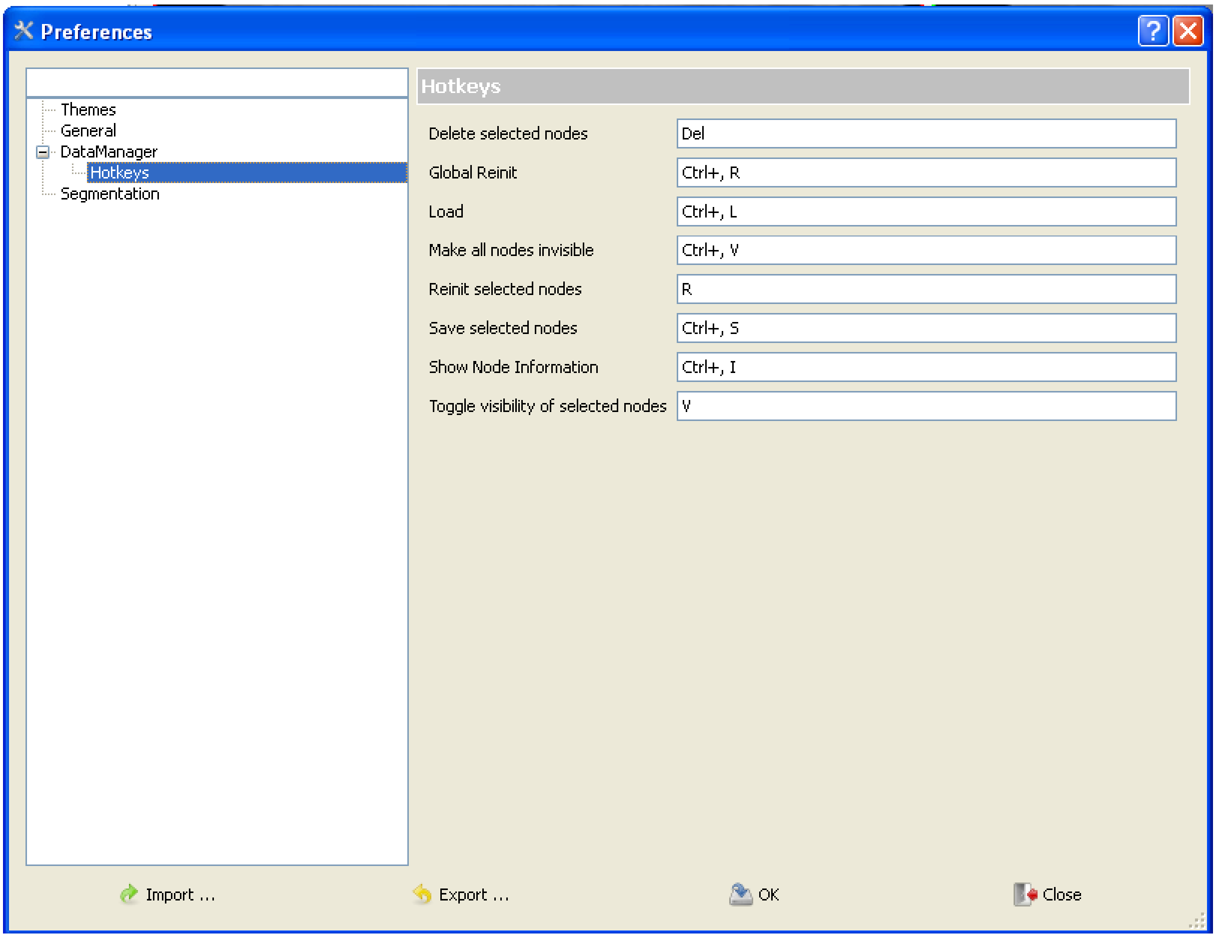


Figure 6.5: Screenshot3: Properties for surfaces

6.4.4 Preferences

For the datamanager there are already some default hotkeys like the del-key for deleting a Data-Element. The whole list is seen in Screenshot4. From here the Hotkeys can also be changed. The preference page is found in "Window"->"Preferences".

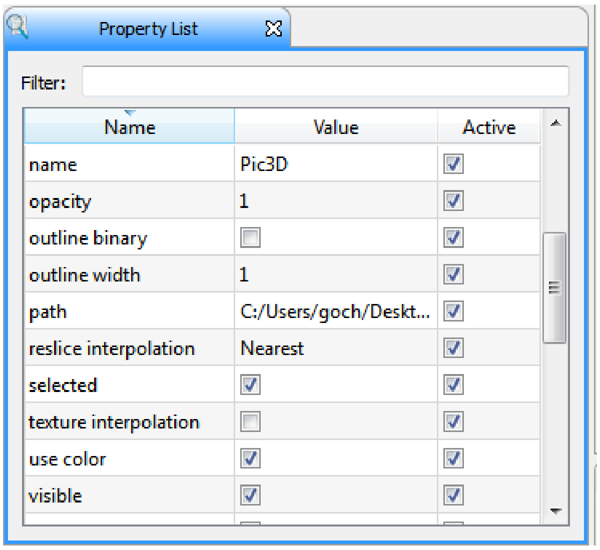


Delete selected nodes	Del
Global Reinit	Ctrl+, R
Load	Ctrl+, L
Make all nodes invisible	Ctrl+, V
Reinit selected nodes	R
Save selected nodes	Ctrl+, S
Show Node Information	Ctrl+, I
Toggle visibility of selected nodes	V

Figure 6.6: Screenshot4

6.5 Property List

The Property List displays all the properties the currently selected Data-Element has. Which properties these are depends on the Data-Element. Examples are opacity, shader, visibility. These properties can be changed by clicking on the appropriate field in the "value" column.



Filter:

Name	Value	Active
name	Pic3D	☑
opacity	1	☑
outline binary	☐	☑
outline width	1	☑
path	C:/Users/goch/Deskt...	☑
reslice interpolation	Nearest	☑
selected	☑	☑
texture interpolation	☐	☑
use color	☑	☑
visible	☑	☑

Figure 6.7: Screenshot5: Property List

Chapter 7

The Basic Image Processing Plugin

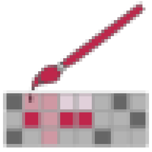


Figure 7.1: Icon of the Basic Image Processing Plugin

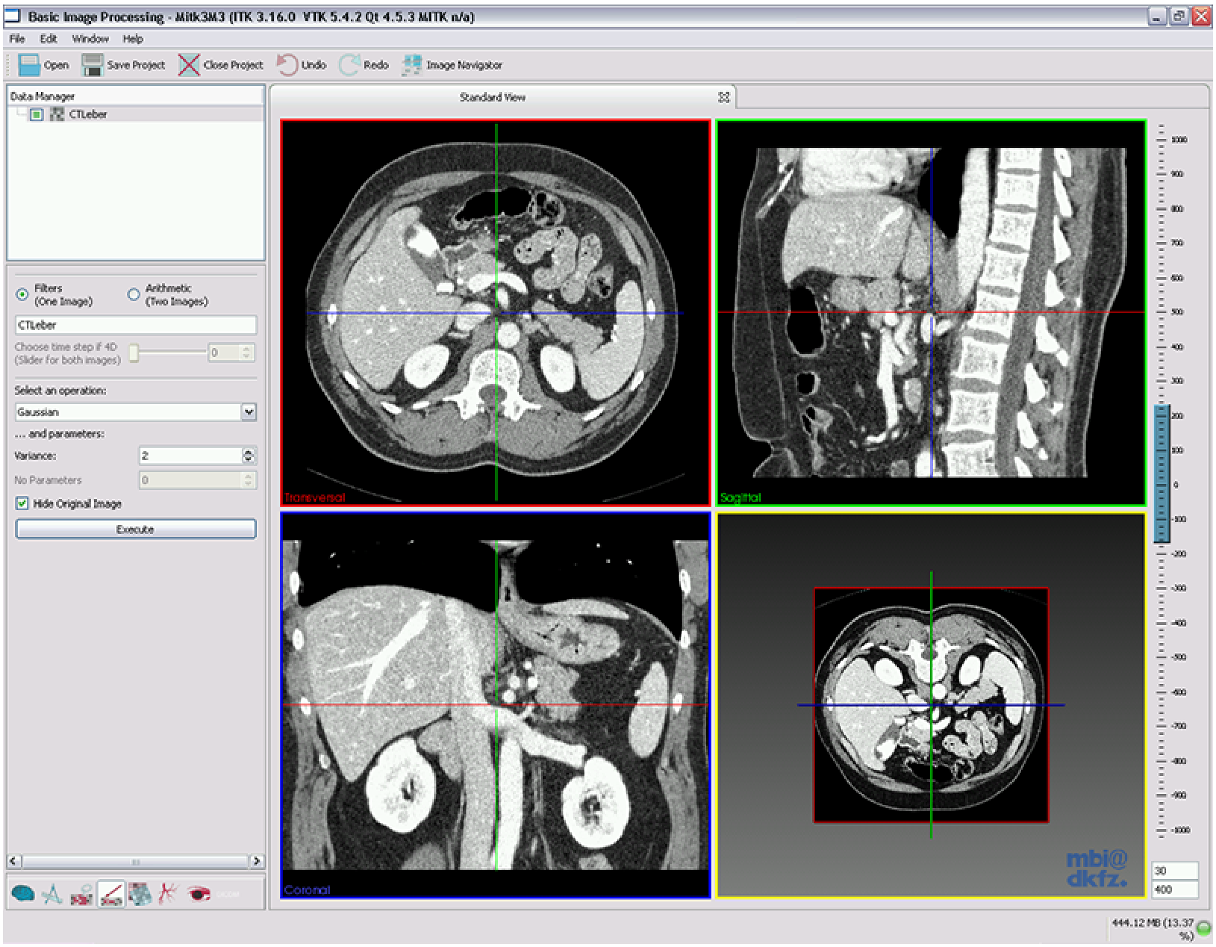
7.1 Summary

This view provides an easy interface to fundamental image preprocessing and enhancement filters. It offers filter operations on 3D and 4D images in the areas of noise suppression, morphological operations, edge detection and image arithmetics, as well as image inversion and downsampling.

Please see Overview for more detailed information on usage and supported filters. If you encounter problems using the view, please have a look at the Troubleshooting page.

7.2 Overview

This view provides an easy interface to fundamental image preprocessing and image enhancement filters. It offers a variety of filter operations in the areas of noise suppression, morphological operations, edge detection and image arithmetics. Currently the view can be used with all 3D and 4D image types loadable by MITK. 2D image support will be added in the future. All filters are encapsulated from the Insight Segmentation and Registration Toolkit (ITK, www.itk.org).



○ Filters (One Image) ○ Arithmetic (Two Images)

CTLeber

Choose time step if 4D (Slider for both images): 0

Select an operation: Gaussian

... and parameters:

Variance: 2

No Parameters: 0

☑ Hide Original Image

Figure 7.2: MITK with the Basic Image Processing view

This document will tell you how to use this view, but it is assumed that you already know how to use MITK in general.

7.3 Filters

This section will not describe the fundamental functioning of the single filters in detail, though. If you want to know more about a single filter, please have a look at `http://www.itk.org/Doxygen316/html/classes.html` or in any good digital image processing book. For total denoising filter, please see Tony F. Chan et al., "The digital TV filter and nonlinear denoising".

Available filters are:

Single image operations

- **Noise Suppression**
  - Gaussian Denoising
  - Median Filtering
  - Total Variation Denoising
- **Morphological Operations**
  - Dilation
  - Erosion

  - Opening
  - Closing
- **Edge Detection**
  - Gradient Image
  - Laplacian Operator (Second Derivative)
  - Sobel Operator
- **Misc**
  - Threshold
  - Image Inversion
  - Downsampling (isotropic)

Dual image operations

- **Image Arithmetics**
  - Add two images
  - Subtract two images
  - Multiply two images
  - Divide two images
- **Binary Operations**
  - Logical AND
  - Logical OR
  - Logical XOR

7.4 Usage

All you have to do to use a filter is to:

- Load an image into MITK
- Select it in data manager
- Select which filter you want to use via the drop down list
- Press the execute button

A busy cursor appeares; when it vanishes, the operation is completed. Your filtered image is displayed and selected for further processing. (If the checkbox "Hide original image" is not selected, you will maybe not see the filter result imideately, because your filtered image is possibly hidden by the original.)

For two image operations, please make sure that the correct second image is selected in the drop down menu, and the image order is correct. For sure, image order only plays a role for image subtraction and division. These are conducted (Image1 - Image2) or (Image1 / Image2), respectively.

Please Note: When you select a 4D image, you can select the time step for the filter to work on via the time slider at the top of the GUI. The 3D image at this time step is extracted and processed. The result will also be a 3D image. This means, a true 4D filtering is not yet supported.

7.5 Troubleshooting

I get an error when using a filter on a 2D image.
2D images are not yet supported...

I use a filter on a 4D image, and the output is 3D.
When you select a 4D image, you can select the time step for the filter to work on via the time slider at the top of the GUI. The 3D image at this time step is extracted and processed. The result will also be a 3D image. This means, a true 4D filtering is not supported by now.

A filter crashes during execution.
Maybe your image is too large. Some filter operations, like derivatives, take a lot of memory. Try downsampling your image first.

All other problems.
Please report to the MITK mailing list. See `http://www.mitk.org/wiki/Mailinglist` on how to do this.

Chapter 8

The Image Navigator

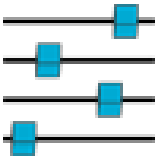
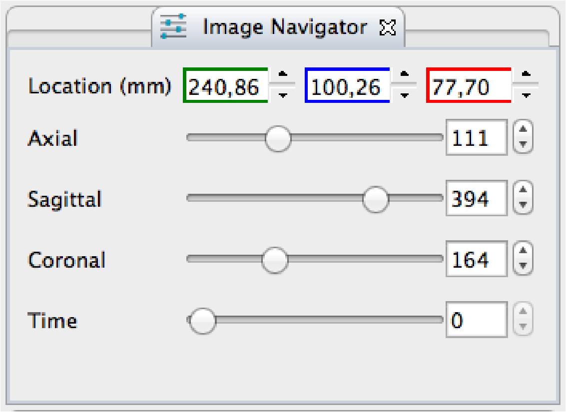


Figure 8.1: Icon of the Image Navigator



Location (mm) 240,86 100,26 77,70

Axial 111

Sagittal 394

Coronal 164

Time 0

Figure 8.2: Image Navigator

Fast movement through the available data can be achieved by using the Image Navigator. By moving the sliders around you can scroll quickly through the slides and timesteps. By entering numbers in the relevant fields you can jump directly to your point of interest.

The "Show detail" checkbox enables you to see the world coordinates in millimetres and the index/voxel coordinates. These may be edited to jump to a specific location.

Chapter 9

The View Navigator

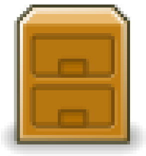


Figure 9.1: Icon of the view navigator

9.1 Overview

This view allows for the easy navigation of the available views.

You can select which view to open by double-clicking the name of the view in the view navigator. It provides a keyworded, grouped and searchable list of views.

9.2 Usage

You can toggle the View Navigator on and off by clicking on its icon in the menu bar. Alternatively it is also available via the Window->Show Views dialog.

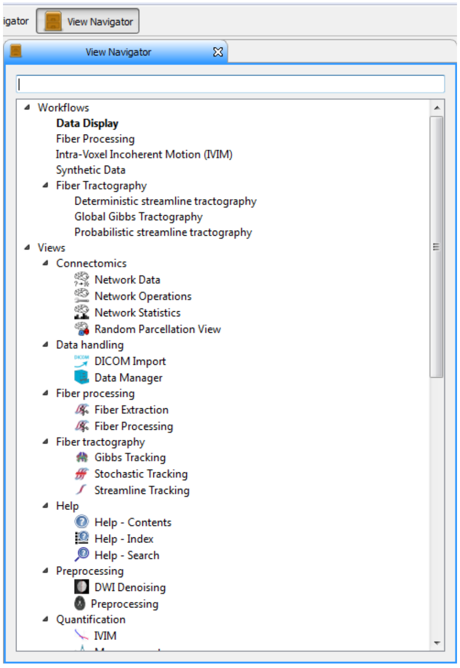


Figure 9.2: The View Navigator GUI

Once the View Navigator has been opened you will see a list divided in workflows/perspectives and views. Via this list you can access any view or perspective provided by the application. They are further organized based on their associated category.

An entry on the list is opened by a double left-click.

9.2.1 Search

You can search the lists for view/workflow names, keywords or categories.

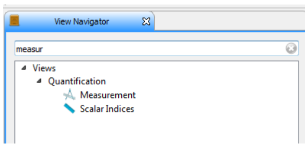


Figure 9.3: Search the View Navigator

9.2.2 Custom Workflows

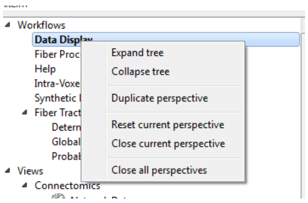


Figure 9.4: The workflow context menu

A right click on a workflow opens a context menu that allows you to copy that workflow to create your own custom one.

Note

The duplicated workflow will look like the basic state of the original workflow. Any changes to the original workflow will not be copied.

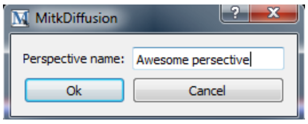


Figure 9.5: Custom workflow creation dialog

Your new workflow will appear in the list and you can modify it independently of the original workflow. Any changes will be stored for future sessions unless the workflow is deleted.

Note

Resetting a custom perspective will return it to the original state after duplicating.

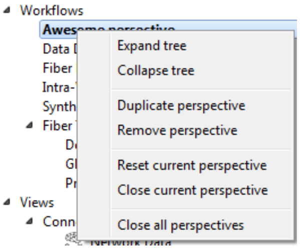


Figure 9.6: Custom workflow context menu

Chapter 10

The Measurement Toolbox Plugin

10.1 Manual

This plugin contains all views that provide measurement and statistics functionality.

- The Measurement View
- The Image Statistics View

10.2 The Measurement View

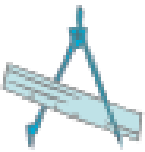
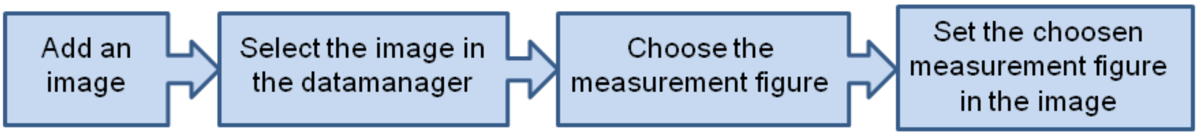


Figure 10.1: Icon of the Measurement View

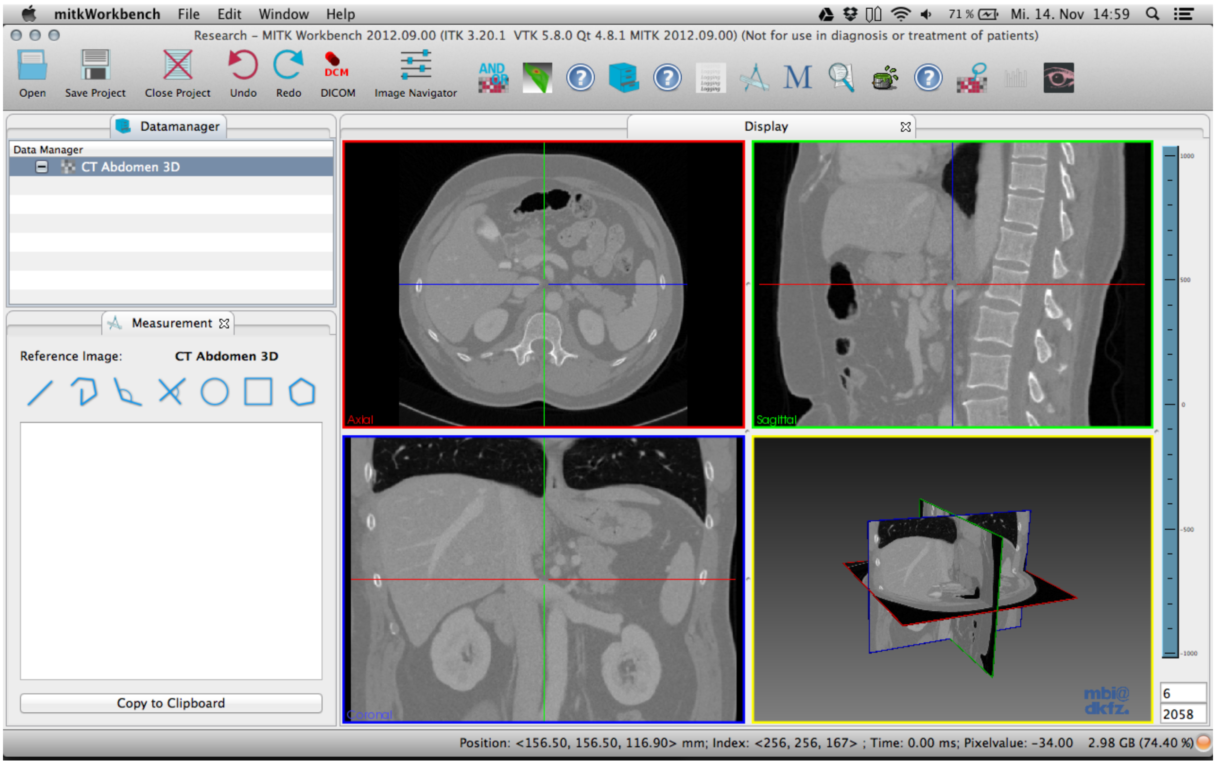
10.2.1 Overview

The Measurement view enables the user to interact with 2D images or single slices of 3D image stacks and planar figure data types. It allows to measure distances, angels, pathes and several geometric figures on a dataset.

The workflow to use this view is:



The workflow is repeatedly useable with the same or different measurement figures, which are correlated to the choosen image and can be saved together with it for future use. On pressing the Measurement icon (see picture below the page title) in the view button line the basic appearance of the view is as follws.



Research – MITK Workbench 2012.09.00 (ITK 3.20.1 VTK 5.8.0 Qt 4.8.1 MITK 2012.09.00) (Not for use in diagnosis or treatment of patients)

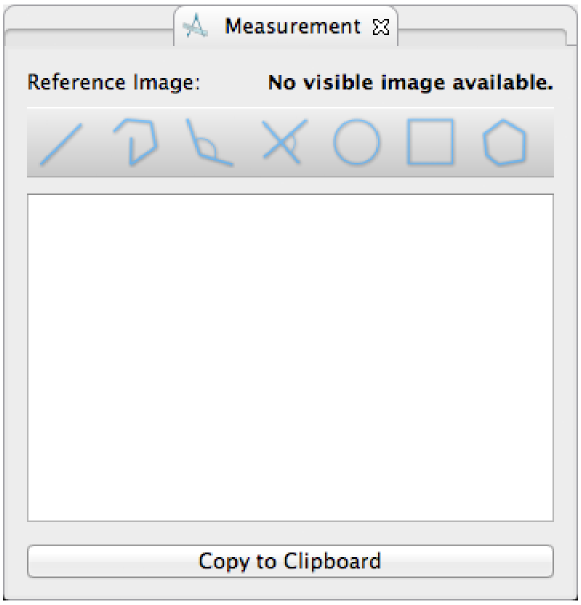
Reference Image: CT Abdomen 3D

Position: <156.50, 156.50, 116.90> mm; Index: <256, 256, 167> ; Time: 0.00 ms; Pixelvalue: -34.00 2.98 GB (74.40 %)

The standard working plane is "Axial" but the other standard viewplanes ("Saggital" and "Coronal") are also valid for measurements. To swap between the view planes refer to the application user manual.

10.2.2 Features

The view as it is depicted below offers the following features in the order of apperance on the image from top to bottom:



Reference Image: No visible image available.

The first information is the selected image's name (here: DICOM-MRI-Image) followed by the measurement figures button line with the seven measurement figures. From left to right the buttons are connected with the following functions:

10.2.2.1 Draw Line

Draws a line between two set points and returns the distance between these points.

10.2.2.2 Draw Path

Draws a path between several set points (two and more) and calculates the circumference, that is all line's length summed up. Add the final point by double left click.

10.2.2.3 Draw Angle

Draws two lines from three set points connected in the second set point and returns the inner angle at the second point.

10.2.2.4 Draw Four Point Angle

Draws two lines that may but must not intersect from four set points. The returned angle is the one depicted in the icon.

10.2.2.5 Draw Circle

Draws a circle by setting two points, whereas the first set point is the center and the second the radius of the circle. The measured values are the radius and the included area.

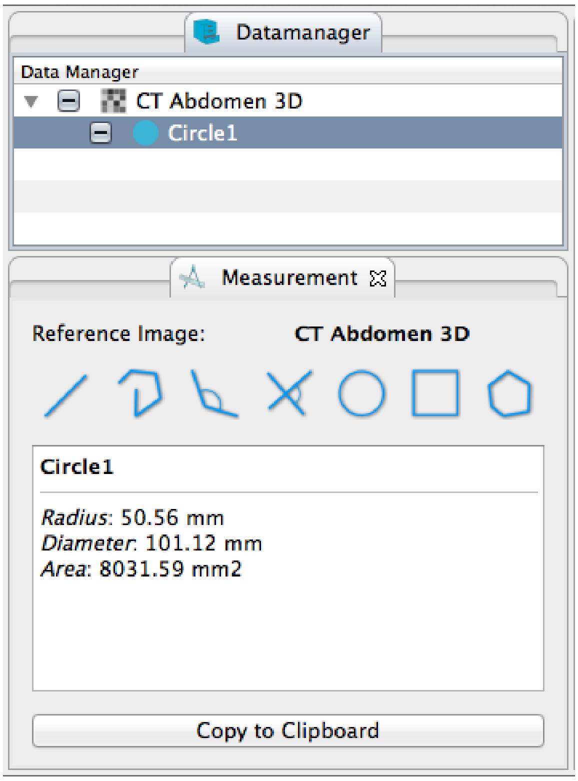
10.2.2.6 Draw Rectangle

Draws a rectangle by setting two points at the opposing edges of the rectangle starting with the upper left edge. The measured values are the circumference and the included area.

10.2.2.7 Draw Polygon

Draws a polygon by setting three or more points. The measured values are the circumference and the included area. Add the final point by double left click.

Below the buttonline the statistics window is situated, it displays the results of the actual measurements from the selected measurement figures. The content of the statistics window can be copied to the clipboard with the correspondig button for further use in a table calculation programm (e.g. Open Office Calc etc.).



Reference Image: CT Abdomen 3D

Circle1

Radius: 50.56 mm
Diameter: 101.12 mm
Area: 8031.59 mm2

The last row contains again a button line to swap from the measurement perspective (activated in the image) to other supported MITK perspectives.

10.2.3 Usage

This Section is subdivided into four subsections:

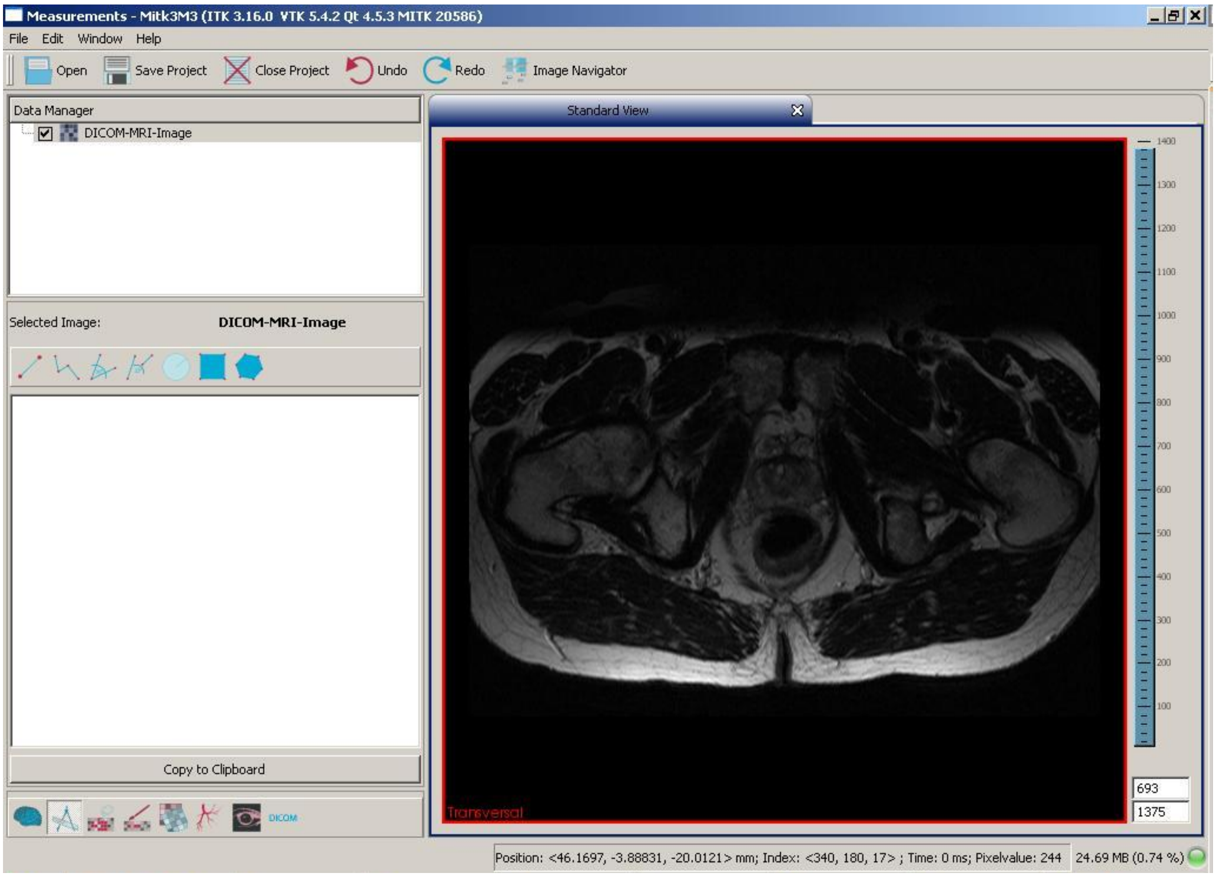
1. **Add an image**
2. **Work with measurement figures**
3. **Save the image with measurement information**
4. **Remove measurement figures or image**

Let's start with subsection 1

10.2.3.1 Add an image

There are two possible ways to add an image to the programm. One is to grap the image with left mouse click from your prefered file browser and simply drag&drop it to the View Plane field. The other way is to use the Open button in the upper left corner of the application. A dialog window appears showing the file tree of the computer. Navigate to the wanted file and select it with the left mouse click. Afterwards just use the dialog's open button.

The wanted image appears in the View Plane and in the Data Manager the images name appears as a new tree node. Now the image is loaded it can be adjusted in the usual way (zoom in/out: right mouse button + moving the mouse up and down, moving the image: press mouse wheel and move the mouse to the wished direction, scroll through the slices(only on 3D images): scroll mouse wheel up and down).



Selected Image: **DICOM-MRI-Image**

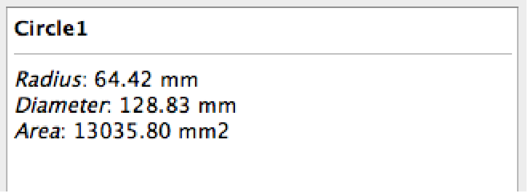
Position: <46.1697, -3.88831, -20.0121> mm; Index: <340, 180, 17> ; Time: 0 ms; Pixelvalue: 244 24.69 MB (0.74 %)

After the image is loaded the image's name appears in the Data Manager. By left-clicking on the image name the buttonline becomes activated.

10.2.3.2 Work with measurement figures

The measurement view comes with seven measurement figures(see picture below), that can be applied to the images.

The results of the measurement with each of these figures is shown in the statistics window and in the lower right corner of the view plane.



Circle1

Radius: 64.42 mm
Diameter: 128.83 mm
Area: 13035.80 mm2

When applying more then one measurement figure to the image the actual measurement figure is depicted in red and the displayed values belong to this measurement figure. All measurement figures become part of the Data Manager as a node of the image tree.

10.2.3.3 Save the image with measurement information

After applying the wanted measurement figures the entire scene consisting of the image and the measurement figures can be saved for future use. Therefore just click the right mouse button when over the image item in the

Data Manager and choose the item "Save" in the opening item list. Following to that a save dialog appears where the path to the save folder can be set. Afterwards just accept your choice with the save button.

10.2.3.4 Remove measurement figures or image

If the single measurement figures or the image is not needed any longer, it can be removed solely or as an entire group. The image can't be removed without simultaneously removing all the dependent measurement figures that belong to the image tree in the Data Manager. To remove just select the wanted items in the data manager list by left-click on it or if several items wanted to be removed left click on all wanted by simultaneously holding the ctrl-button pressed.

For more detailed usage of the save/remove functionality refer to the Data Manager User Manual.

" _linebr \image latex QmitkMeasurementToolbox_MeasurementGUI.png "Graphical User Interface of Measurement –>" width=16.00cm

10.3 The Image Statistics View

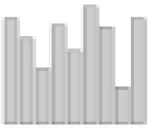


Figure 10.2: Icon of the Image Statistics View

10.3.1 Summary

This view provides an easy interface to quickly compute some features of a whole image or a region of interest.

This document will tell you how to use this view, but it is assumed that you already know how to use MITK in general.

Please see Details for more detailed information on usage and supported filters. If you encounter problems using the view, please have a look at the Troubleshooting page.

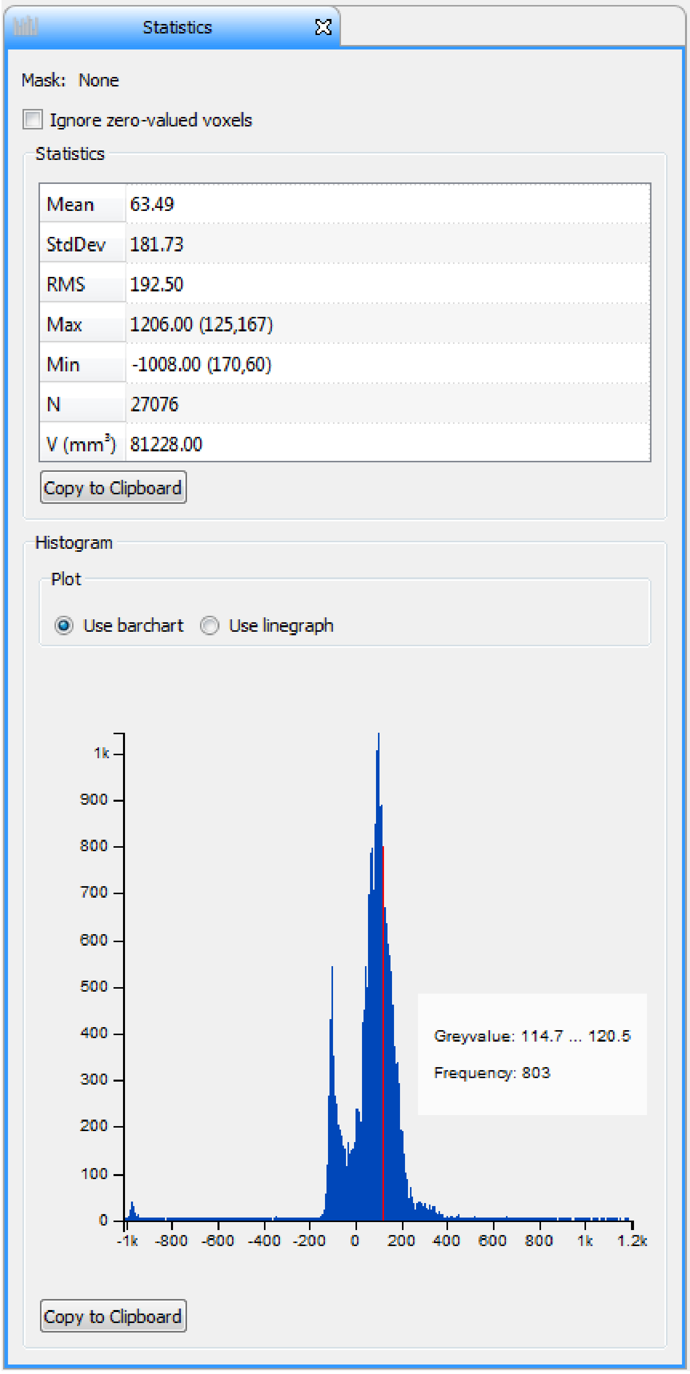
10.3.2 Details

Manual sections:

- Overview
- Usage
- Troubleshooting

10.3.3 Overview

This view provides an easy interface to quickly compute some features of a whole image or a region of interest.



Mask: None

☐ Ignore zero-valued voxels

Statistics

Mean	63.49
StdDev	181.73
RMS	192.50
Max	1206.00 (125,167)
Min	-1008.00 (170,60)
N	27076
V (mm^3)	81228.00

● Use barchart ○ Use linegraph

Greyvalue: 114.7 ... 120.5

Frequency: 803

Figure 10.3: The interface

10.3.4 Usage

After selection of an image or a binary mask of an image in the datamanager, the Image Statistics view shows some statistical information. If a mask is selected, the name of the mask and the name of the image, to which the mask is applied, are shown at the top. For time data the current time step is used for the selected mask and the selected image. If the total number of time steps on the selected mask is less than the current time step, the last time step of the mask is used. If a mask is selected, its used time step will be displayed next to its name like this: (t=0).

Check "Ignore zero-valued voxels" to hide voxels with grayvalue zero.

Below it is the statistics window which displays the calculated statistical features (such as mean, standard deviation...).

Beneath the statistics window is the histogram window, which shows the histogram of the current selection.

At top of the histogram window are two radiobuttons. Toggle one of them to either show the histogram as a barchart or as a lineplot.

Use mousewheel to zoom in and out the histogram. With the left mouse button the histogram is pannable in zoomed state.

If the histogram is displayed as a barchart a tooltip is available by hovering over one of the bins. A tooltip is also available, if an intesity profile is created for a path element as mask.

At the bottom of each view is one button. They copy their respective data in csv format to the clipboard.

10.3.5 Troubleshooting

No known problems.

All other problems.
Please report to the MITK mailing list. See `http://www.mitk.org/wiki/Mailinglist` on how to do this.

Chapter 11

The Movie Maker Plugin

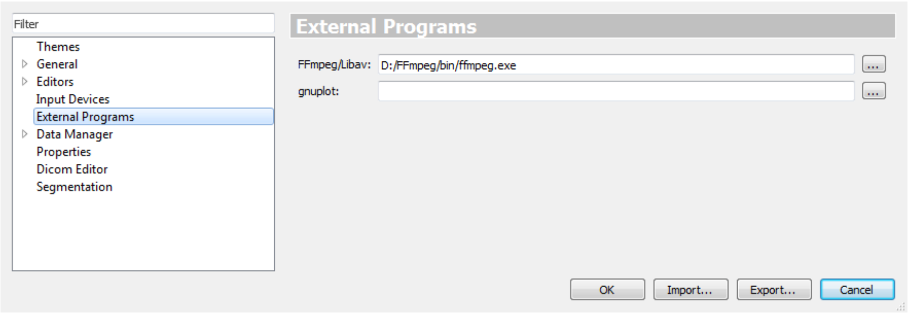
Figure 11.1: Icon of the Movie Maker Plugin.

11.1 Overview

The Movie Maker View allows you to create basic animations of your scene and to record them to video files. Individual animations are arranged in a timeline and can be played back sequential or in parallel.

The Movie Maker View uses external FFmpeg/Libav command line utilities to write compressed video files.

You have to manually install either FFmpeg or Libav and set the corresponding path in "External Programs" in the MITK Workbench Preferences (Ctrl+P) in order to record your movies to video files.

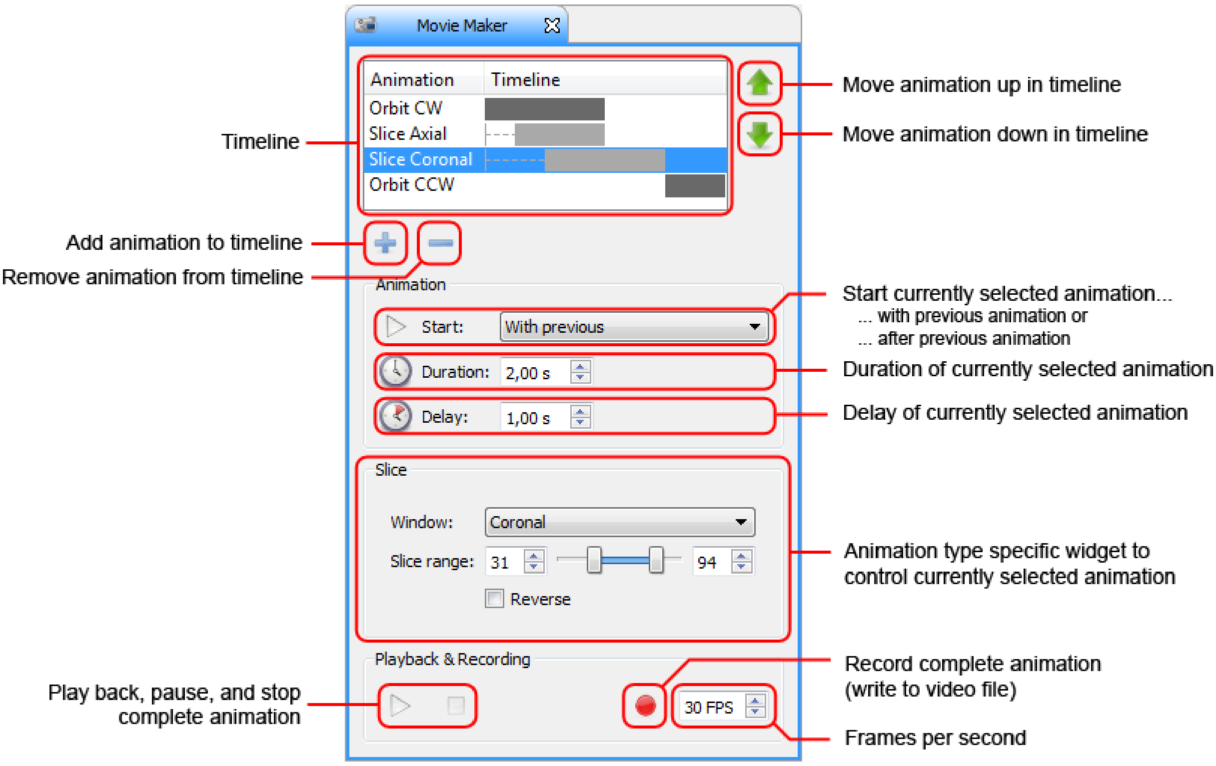


FFmpeg/Libav: D:/FFmpeg/bin/ffmpeg.exe

gnuplot:

Figure 11.2: The External Programs preferences page.

11.2 Usage



Start: With previous

Duration: 2,00 s

Delay: 1,00 s

Window: Coronal

Slice range: 31 94

☐ Reverse

30 FPS

Figure 11.3: The Movie Maker View.

To create a movie you have to add an animation to the timeline by clicking the "Add animation" button. You can choose between the available types of animations, e.g., Orbit or Slice.

The timeline surrounding bottons allow you to arrange, remove, or add further animations to your movie.

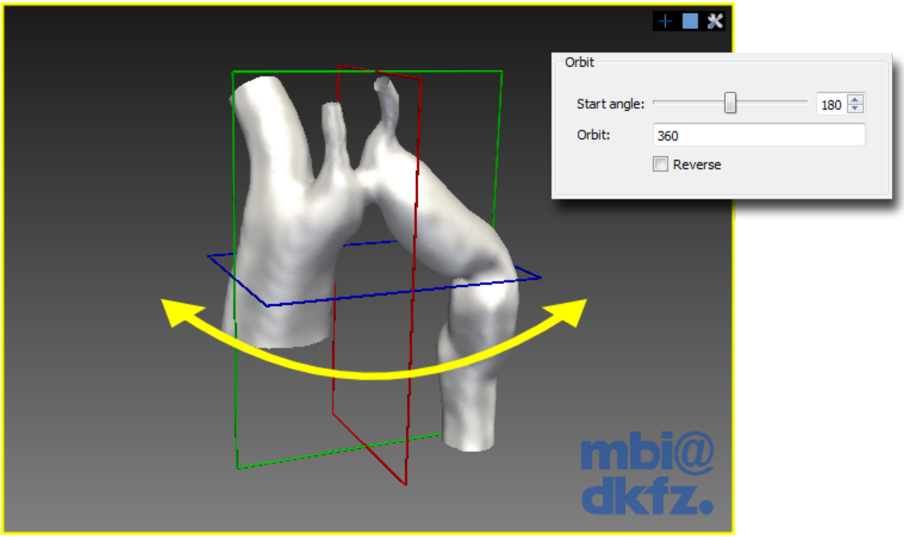
Each animation can be set to either begin with the previous animation, i.e., run in parallel, or to start after the previous animation, i.e., run sequential. In combination with delays, rather complex animation arrangements are possible.

To set animation specific parameters, select the corresponding animation in the timeline first.

You can play back, pause and stop your movie with the according controls at the bottom of the Movie Maker View. Click the "Record" button to finally record your movie to a video file with the specified number of frames per second. You have to choose the render window which you want to record.

11.2.1 Orbit Animation

The Orbit animation rotates the camera in the 3D window around the scene. Align the camera directly in the 3D window and enter the number of degrees for the orbitting. You usually do not need to adjust the "Start angle" setting - leave it at 180 degrees. If you are planning to have a specific view in the middle of your movie you can play the movie and pause it at the specific frame of interest. Adjust the camera in the 3D window and restart the animation.



Orbit

Start angle: 180

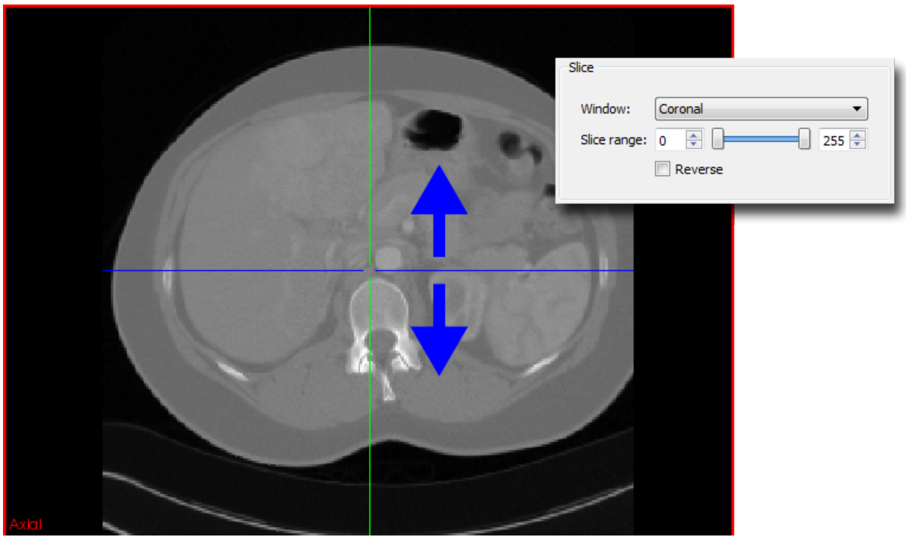
Orbit: 360

☐ Reverse

Figure 11.4: The Orbit animation.

11.2.2 Slice Animation

The Slice animation slices through an image. You can choose the image plane (axial, sagittal, or coronal), as well as the start and end points of the slicing. Use the image navigator in the bottom left of the Workbench to get an idea of the desired values. Check "Reverse" in order to slice from the higher slice number to the lower slice number.



Slice

Window: Coronal

Slice range: 0 255

☐ Reverse

Figure 11.5: The Slice animation.

11.2.3 Time Animation

The Time animation steps through the individual time steps of the current scene. You can specify the range of the animated time steps. Use the image navigator in the bottom left of the Workbench to get an idea of the desired values. Check "Reverse" in order to step from later time steps to previous time steps.

Figure 11.6: The Time animation.

Chapter 12

The Registration Plugin

12.1 Overview

MITK provides several views for the registration of images.

12.2 List of Views

- The Point Based Registration View
- The Rigid Registration View

12.3 The Point Based Registration View

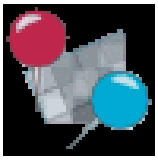


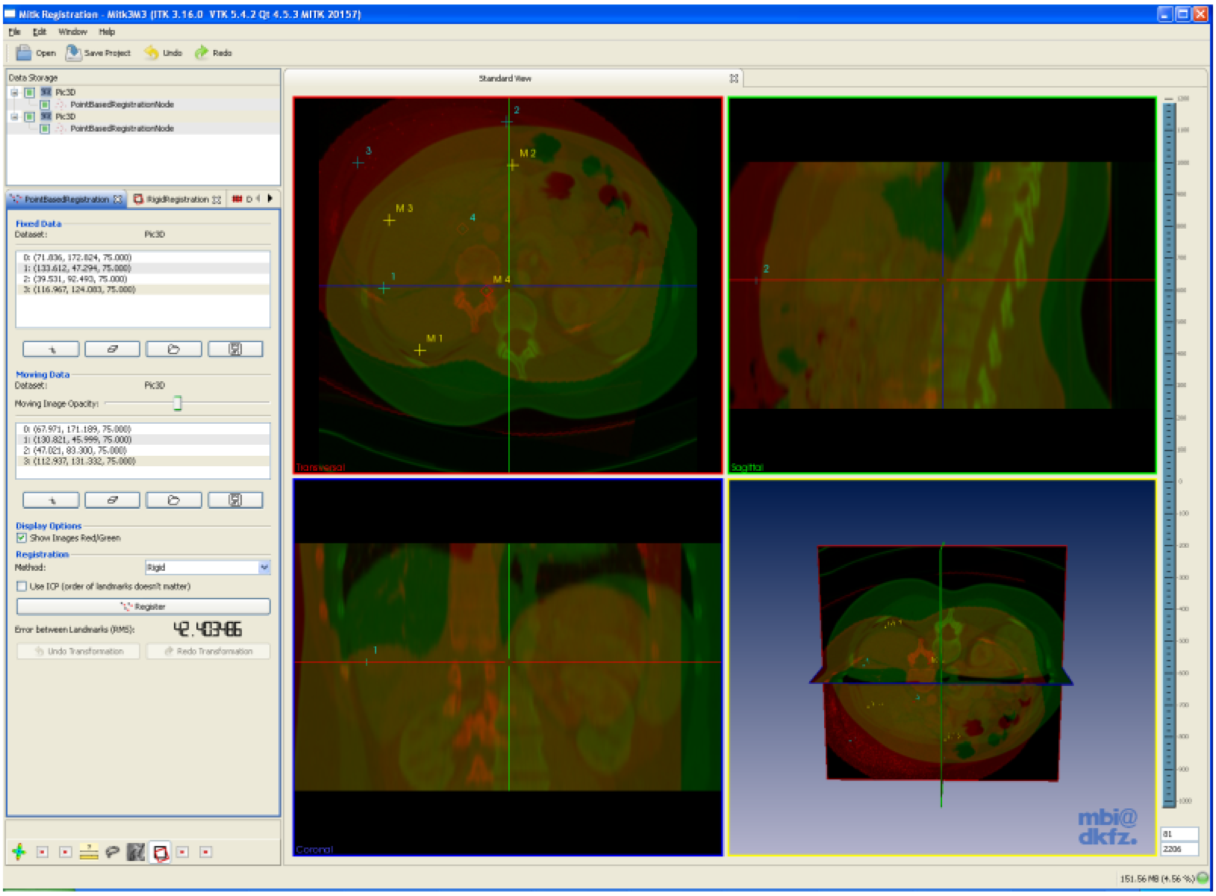
Figure 12.1: Icon of the Point Based Registration View

Available sections:

- Overview
- Details

12.3.1 Overview

This view allows you to register two datasets in a rigid and deformable manner via corresponding PointSets. Register means to align two datasets, so that they become as similar as possible. Therefore you have to set corresponding points in both datasets, which will be matched. The movement, which has to be performed on the points to align them, will be performed on the moving data as well. The result is shown in the multi-widget.



Fixed Data
Dataset: Pic3D

0: (71.836, 172.824, 75.000)
1: (133.612, 47.294, 75.000)
2: (39.531, 92.493, 75.000)
3: (116.967, 124.083, 75.000)

Moving Data
Dataset: Pic3D
Moving Image Opacity:

0: (67.971, 171.189, 75.000)
1: (130.821, 45.999, 75.000)
2: (47.021, 83.300, 75.000)
3: (112.937, 131.332, 75.000)

Display Options
☑ Show Images Red/Green

Registration
Method: Rigid
☐ Use ICP (order of landmarks doesn't matter)

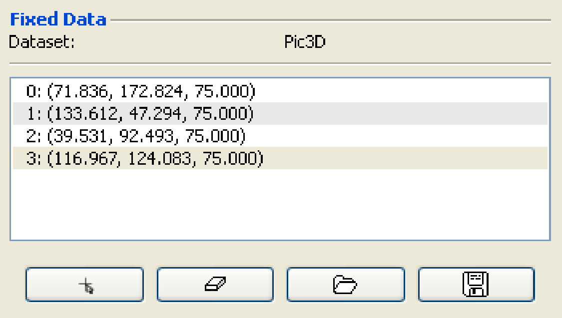
Error between Landmarks (RMS): 42.403466

Figure 12.2: MITK with the PointBasedRegistration view

This document will tell you how to use this view, but it is assumed that you already know how to navigate through the slices of a dataset using the multi-widget.

12.3.2 Details

First of all you have to open the data sets which you want to register and select them in the Data Manager. You have to select exactly 2 images for registration. The image which was selected first will become the fixed image, the other one the moving image. The two selected images will remain for registration until exactly two images were selected in the Data Manager again. While there aren't two images for registration a message is viewed on top of the view saying that registration needs two images. If two images are selected the message disappears and the interaction areas for the fixed and moving data appears. The upper area is for interaction with the fixed data. Beneath this area is the interaction area for the moving data. On default only the fixed and moving image with their corresponding pointsets are shown in the render windows. If you want to have other images visible you have to set the visibility via the Data Manager. Also if you want to perform a reinit on a specific node or a global reinit for all nodes you have to use the Data Manager.

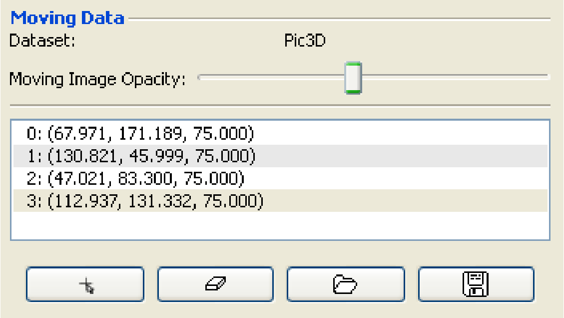


Dataset: Pic3D

0: (71.836, 172.824, 75.000)
1: (133.612, 47.294, 75.000)
2: (39.531, 92.493, 75.000)
3: (116.967, 124.083, 75.000)

Figure 12.3: The Fixed Data area

The "Fixed Data" area contains a QmitkPointListWidget. Within this widget, all points for the fixed data are listed. The label above this list shows the number of points that are already set. To set points you have to toggle the "Set Points" button, the leftmost under the QmitkPointListWidget. The views in the QmitkStdMultiWidget were reinitialized to the fixed data. Points can be defined by performing a left click while holding the "Shift"-key pressed in the QmitkStdMultiWidget. You can remove the interactor which listens for left clicks while holding the "Shift"-key pressed by detoggle the "Set Points" button. The next button, "Clear Point Set", is for deleting all specified points from this dataset. The user is prompted to confirm the decision. With the most right button, a previously saved point set can be loaded and all of its points are shown in the QmitkPointListWidget and in the QmitkStdMultiWidget. The user is prompted to select the file to be loaded. The file extension is ".mps". On the left of this button is the save button. With this function all points specified for this dataset and shown in the QmitkPointListWidget are saved to harddisk. The user is prompted to select a filename. Pointsets were saved in XML fileformat but have to have a ".mps" file extension. You can select landmarks in the render window with a left mouse button click on them. If you keep the mouse button pressed you can move the landmark to an other position by moving the mouse and then release the mouse button. With the delete key you can remove the selected landmarks. You can also select landmarks by a double click on a landmark within the QmitkPointListWidget. Using the "Up-Arrow"-button or the "F2" key you can easily move a landmark upwards and bring it further downwards by pressing "F3" or using the "Down-Arrow"-button. Thus the landmark number can be changed. The QmitkStdMultiWidget changes its view to show the position of the landmark.



Dataset: Pic3D

Moving Image Opacity:

0: (67.971, 171.189, 75.000)
1: (130.821, 45.999, 75.000)
2: (47.021, 83.300, 75.000)
3: (112.937, 131.332, 75.000)

Figure 12.4: The Moving Data area

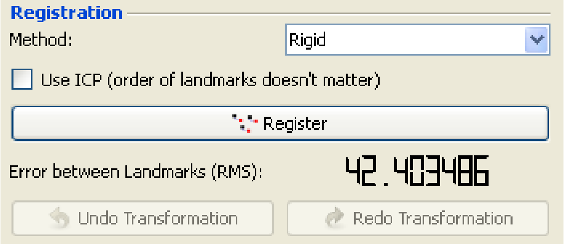
The "Moving Data" area contains a QmitkPointListWidget. Within this widget, all points for the moving data are listed. The label above this list shows the number of points that are already set. To set points you have to toggle the "Set Points" button, the leftmost under the QmitkPointListWidget. The views in the QmitkStdMultiWidget were reinitialized to the moving data. With the "Opacity:" slider you can change the opacity of the moving dataset. If the slider is leftmost the moving dataset is totally transparent, whereas if it is rightmost the moving dataset is totally opaque. Points can be defined by performing a left click while holding the "Shift"-key pressed in the QmitkStdMultiWidget. You can remove the interactor which listens for left mousebutton click while holding the "Shift"-key pressed by detoggle the "Set Points" button. The next button, "Clear Point Set", is for deleting all specified points from this dataset. The user is prompted to confirm the decision. With the button on your right hand side, a previously saved point set can be loaded and all of its points are shown in the QmitkPointListWidget and in the QmitkStdMultiWidget. The user is prompted to select the file to be loaded. The file extension is ".mps". On the left of this button is the save button. With this function all points specified for this dataset and shown in the QmitkPointListWidget are saved to harddisk. The user is prompted to select a filename. Pointsets were saved in XML fileformat but have to have a ".mps" file extension. You can select landmarks in the render window with a left click on them. If you keep the mouse button pressed you can move the landmark to an other position by moving the mouse and then release the mouse button. With the delete key you can remove the selected landmarks. You can also select landmarks by a double click on a landmark within the QmitkPointListWidget. Using the "Up-Arrow"-button or the "F2" key you can easily move a landmark upwards and bring it further downwards by pressing "F3" or using the "Down-Arrow"-button. Thus the landmark number can be changed.The QmitkStdMultiWidget changes its view to show the position of the landmark.

☑ Show Images Red/Green

Figure 12.5: The Display Options area

In this area you can find the "Show Images Red/Green" checkbox. Here you can switch the color from both datasets. If you check the box, the fixed dataset will be displayed in redvalues and the moving dataset in greenvalues to improve visibility of differences in the datasets. If you uncheck the "Show Images Red/Green" checkbox, both datasets will be displayed in greyvalues.

Before you perform your transformation it is useful to see both images again. Therefore detoggle the "Set Points" button for the fixed data as well as for the moving data.



Method: Rigid

☐ Use ICP (order of landmarks doesn't matter)

Error between Landmarks (RMS): 42.403486

Figure 12.6: The Registration area

The functions concerning the registration are displayed in the "Registration" area. It not only contains the registration method selection and the registration itself but also offers the possibility to save, undo or redo the results. Furthermore a display is implemented, which shows you how good the landmarks correspond. Those features will be explained in following paragraphs.

Using the "Method"-selector, you can pick one of those transformations: **Rigid**, **Similarity**, **Affine** and **Landmark Warping**. Depending on which one you chose, an additional specifier, "Use ICP" can be set, which leads to the following possibilities for registration:

- **Rigid with ICP** means only translation and rotation. The order of your landmarks will not be taken into account. E. g. landmark one in the fixed data can be mapped on landmark three in the moving data. You have to set at least one landmark in each dataset to enable the Register button which performs the transformation.

- **Similarity with ICP** means only translation, scaling and rotation. The order of your landmarks will not be taken into account. E. g. landmark one in the fixed data can be mapped on landmark three in the moving data. You have to set at least one landmark in each dataset to enable the Register button which performs the transformation.

- **Affine with ICP** means only translation, scaling, rotation and shearing. The order of your landmarks will not be taken into account. E. g. landmark one in the fixed data can be mapped on landmark three in the moving data. You have to set at least one landmark in each dataset to enable the Register button which performs the transformation.

- **Rigid** means only translation and rotation. The order of your landmarks will be taken into account. E. g. landmark one in the fixed data will be mapped on landmark one in the moving data. You have to set at least one landmark and the same number of landmarks in each dataset to enable the Register button which performs the transformation.

- **Similarity** means only translation, scaling and rotation. The order of your landmarks will be taken into account. E. g. landmark one in the fixed data will be mapped on landmark one in the moving data. You have to set at least one landmark and the same number of landmarks in each dataset to enable the Register button which performs the transformation.

- **Affine** means only translation, scaling, rotation and shearing. The order of your landmarks will be taken into account. E. g. landmark one in the fixed data will be mapped on landmark one in the moving data. You have to set at least one landmark and the same number of landmarks in each dataset to enable the Register button which performs the transformation.

- **LandmarkWarping** means a freeform deformation of the moving image, so that afterwards the landmarks are exactly aligned. The order of your landmarks will be taken into account. E. g. landmark one in the fixed data will be mapped on landmark one in the moving data. You have to set at least one landmark and the same number of landmarks in each dataset to enable the Register button which performs the transformation.

The root mean squares difference between the landmarks will be displayed as number, so that you can check how good the landmarks correspond.

The "Undo Transformation" button becomes enabled after performing a transformation and allows you to undo it. After doing this, the "Redo Transformation" button is enabled and lets you redo, the just undone transformation(no calculation needed)

Saving of the transformed image can be done via the Data Manager.

12.4 The Rigid Registration View

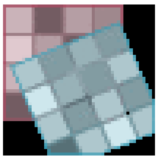


Figure 12.7: Icon of the View

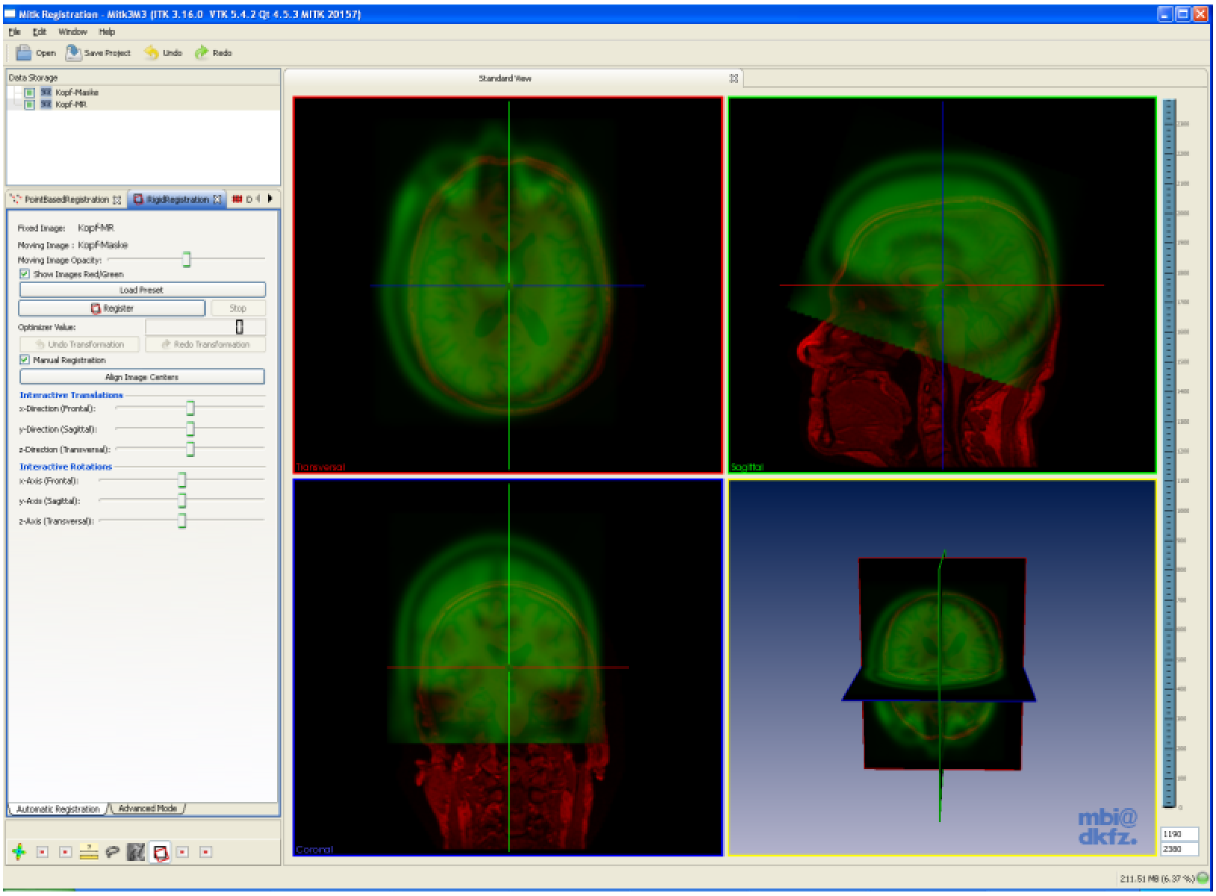
Available sections:

- Overview
- Known Issues
- Details
- References:

12.4.1 Overview

This view allows you to register 2D as well as 3D images in a rigid manner. If the Moving Image is an image with multiple timesteps you can select one timestep for registration. Register means to align two images, so that they become as similar as possible. Therefore you can select from different transforms, metrics and optimizers.

Registration results will directly be applied to the Moving Image. Also binary images as image masks can be used to restrict the metric evaluation only to the masked area.



Fixed Image: Kopf-MR
Moving Image : Kopf-Maske
Moving Image Opacity:
☑ Show Images Red/Green
Optimizer Value:
☑ Manual Registration

Figure 12.8: MITK with the QmitkRigidRegistration view

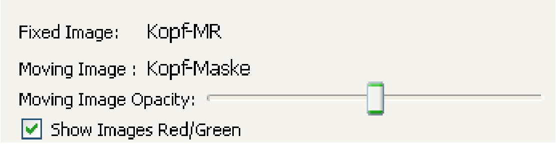
This document will tell you how to use this view, but it is assumed that you already know how to navigate through the slices of an image using the multi-widget.

12.4.2 Known Issues

Depending on your system the registration can fail to allocate memory for calculating the gradient image for registration. In this case you can try to select another optimizer which is not based on a gradient image and uncheck the checkbox for "Compute Gradient".

12.4.3 Details

First of all you have to open the data sets which you want to register and select them in the Data Manager. You have to select exactly 2 images for registration. The image which was selected first will become the fixed image, the other one the moving image. The two selected images will remain for registration until exactly two images were selected in the Data Manager again.



Fixed Image: Kopf-MR

Moving Image : Kopf-Maske

Moving Image Opacity:

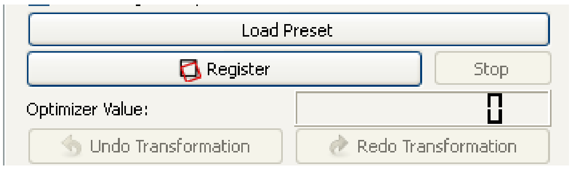
☑ Show Images Red/Green

Figure 12.9: The Image area

While there aren't two images for registration a message is viewed on top of the view saying that registration needs two images. If two images are selected the message disappears and the interaction areas for the fixed and moving data appears. If both selected images have a binary image as childnode a selection box appears which allows, when checked, to use the binary images as image mask to restrict the registration on this certain area. If an image has more than one binary image as child, the upper one from the DataManager list is used. If the Moving Image is a dynamic images with several timesteps a slider appears to select a specific timestep for registration.

On default only the fixed and moving image are shown in the render windows. If you want to have other images visible you have to set the visibility via the Data Manager. Also if you want to perform a reinit on a specific node or a global reinit for all nodes you have to use the Data Manager.

The colour of the images can be changed between grey values and red/green and the opacity of the moving image can be changed. With the "Moving Image Opacity:" slider you can change the opacity of the moving dataset. In the "Show Images Red/Green" you can switch the color from both datasets. If you check the box, the fixed dataset will be displayed in red-values and the moving dataset in green-values to improve visibility of differences in the datasets. If you uncheck the "Show Images Red/Green" checkbox, both datasets will be displayed in grey-values.



Optimizer Value: 0

Figure 12.10: The Registration area

In the "Register" area you can start the registration by clicking the "Calculate Transform" button. The optimizer value for every iteration step is diplayed as LCD number next to the "Optimizer Value:" label. Many of the registration methods can be canceled during their iteration steps by clicking the "Stop Optimization" button. During the calculation, a progress bar indicates the progress of the registration process. The render widgets are updated for every single iteration step, so that the user has the chance to supervise how good the registration process works with the selected methods and parameters. If the registration process does not lead to a sufficient result, it is possible to undo the transformation and restart the registration process with some changes in parameters. The differences in transformation due to the changed parameters can be seen in every iteration step and help the user understand the parameters. Also the optimizer value is updated for every single iteration step and shown in the GUI. The optimizer value is an indicator for the misalignment between the two images. The real time visualization of the registration as well as the optimizer value provides the user with information to trace the improvement through the optimization process. The "Undo Transformation" button becomes enabled when you have performed an transformation and you can undo the performed transformations. The "Redo Transformation" button becomes enabled when you have performed an undo to redo the transformation without to recalculate it.

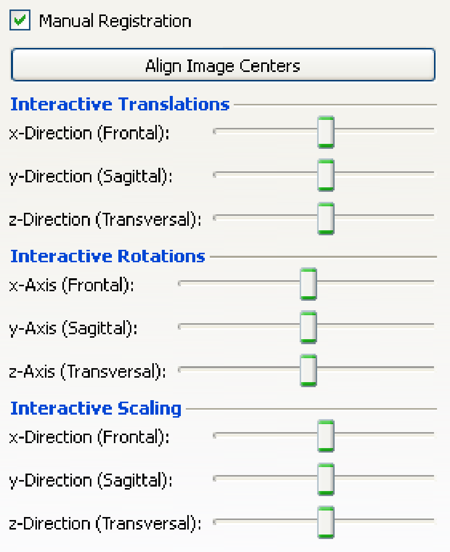
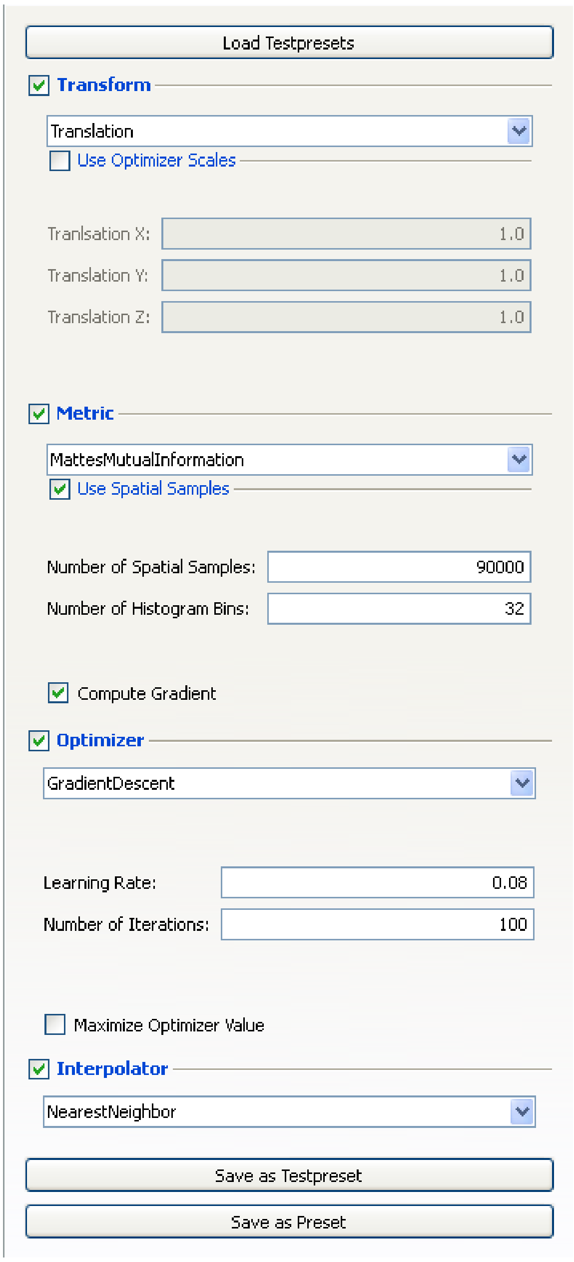


Figure 12.11: The Manual Registration area

In the "Manual Registration" area, shown by checking the checkbox Manual Registration, you can manually allign the images by moving sliders for translation and scaling in x-, y- and z-axis as well as for rotation around the x-, y- and z-Axis. Additionally you can automatically allign the image centers with the button "Automatic Allign Image Centers".



☑ Transform
Translation
☐ Use Optimizer Scales
Tranlsation X: 1.0
Translation Y: 1.0
Translation Z: 1.0

☑ Metric
MattesMutualInformation
☑ Use Spatial Samples
Number of Spatial Samples: 90000
Number of Histogram Bins: 32
☑ Compute Gradient

☑ Optimizer
GradientDescent
Learning Rate: 0.08
Number of Iterations: 100
☐ Maximize Optimizer Value

☑ Interpolator
NearestNeighbor

Figure 12.12: The Advanced Mode tab

In the "Advanced Mode" tab you can choose a transform, a metric, an optimizer and an interpolator and you have to set the corresponding parameters to specify the registration method you want to perform. With the topmost button you can also load testpresets. These presets contain all parametersets which were saved using the "Save as Testpreset" button. The "Save as Preset" button makes the preset available from the "Automatic Registration" tab. This button should be used when a preset is not intended for finding good parameters anymore but can be used as standard preset.

To show the current transform and its parameters for the registration process, the Transform checkbox has to be checked. Currently, the following transforms are implemented (for detailed information see [1] and [2]):

- **Translation:** Transformation by a simple translation for every dimension.
- **Scale:** Transformation by a certain scale factor for each dimension.
- **ScaleLogarithmic:** Transformation by a certain scale factor for each dimension. The parameter factors are passed as logarithms.

- **Affine:** Represents an affine transform composed of rotation, scaling, shearing and translation.
- **FixedCenterOfRotationAffine:** Represents an affine transform composed of rotation around a user provided center, scaling, shearing and translation.
- **Rigid3D:** Represents a 3D rotation followed by a 3D translation.
- **Euler3D:** Represents a rigid rotation in 3D space. That is, a rotation followed by a 3D translation.
- **CenteredEuler3D:** Represents a rigid rotation in 3D space around a user provided center. That is, a rotation followed by a 3D translation.
- **QuaternionRigid:** Represents a 3D rotation and a 3D translation. The rotation is specified as a quaternion.
- **Versor:** Represents a 3D rotation. The rotation is specified by a versor or unit quaternion.
- **VersorRigid3D:** Represents a 3D rotation and a 3D translation. The rotation is specified by a versor or unit quaternion.
- **ScaleSkewVersor3D:** Represents a 3D translation, scaling, shearing and rotation. The rotation is specified by a versor or unit quaternion.
- **Similarity3D:** Represents a 3D rotation, a 3D translation and homogeneous scaling.
- **Rigid2D:** Represents a 2D rotation followed by a 2D translation.
- **CenteredRigid2D:** Represents a 2D rotation around a user provided center followed by a 2D translation.
- **Euler2D:** Represents a 2D rotation and a 2D translation.
- **Similarity2D:** Represents a 2D rotation, homogeneous scaling and a 2D translation.
- **CenteredSimilarity2D:** Represents a 2D rotation around a user provided center, homogeneous scaling and a 2D translation.

The desired transform can be chosen from a combo box. All parameters defining the selected transform have to be specified within the line edits and checkboxes underneath the transform combo box. To show the current metric and its parameters for the registration process, the Metric checkbox has to be checked. Currently, the following metrics are implemented (for detailed information see [1] and [2]):

- **MeanSquares:** Computes the mean squared pixel-wise difference in intensity between image A and B.
- **NormalizedCorrelation:** Computes pixel-wise cross correlation and normalizes it by the square root of the autocorrelation of the images.
- **GradientDifference:** Evaluates the difference in the derivatives of the moving and fixed images.
- **KullbackLeiblerCompareHistogram**[3]**:** Measures the relative entropy between two discrete probability distributions.
- **CorrelationCoefficientHistogram:** Computes the cross correlation coefficient between the intensities.
- **MeanSquaresHistogram:** The joint histogram of the fixed and the mapped moving image is built first. Then the mean squared pixel-wise difference in intensity between image A and B is calculated.

- **MutualInformationHistogram:** Computes the mutual information between image A and image B.
- **NormalizedMutualInformationHistogram:** Computes the mutual information between image A and image B.
- **MattesMutualInformation**[4, 5]**:** The method of Mattes et al. is used to compute the mutual information between two images to be registered.
- **MeanReciprocalSquareDifference:** Computes pixel-wise differences and adds them after passing them through a bell-shaped function 1 / (1+x^2).
- **MutualInformation**[6]**:** Computes the mutual information between image A and image B.
- **MatchCardinality:** Computes cardinality of the set of pixels that match exactly between the moving and fixed images.
- **KappaStatistic**[7]**:** Computes spatial intersection of two binary images.

The desired metric can be chosen from a combo box. All parameters defining the selected metric have to be specified within the line edits and checkboxes underneath the metric combo box. To show the current optimizer and its parameters for the registration process, the Optimizer checkbox has to be checked. Currently, the following optimizers are implemented (for detailed information see [1] and [2]):

- **Exhaustive:** Fully samples a grid on the parametric space.
- **GradientDescent:** A simple gradient descent optimizer.
- **QuaternionRigidTransformGradientDescent:** Variant of a gradient descent optimizer.
- **LBFGSB**[8, 9]**:** Limited memory Broyden Fletcher Goldfarb Shannon minimization with simple bounds.
- **OnePlusOneEvolutionary**[10]**:** 1+1 evolutionary strategy.
- **Powell:** Implements Powell optimization using Brent line search.
- **FRPR:** Fletch-Reeves & Polak-Ribiere optimization using dBrent line search.
- **RegularStepGradientDescent:** Variant of a gradient descent optimizer.
- **VersorTransform:** Variant of a gradient descent optimizer.
- **Amoeba:** Implementation of the Nelder-Meade downhill simplex algorithm.
- **ConjugateGradient:** Used to solve unconstrained optimization problems.
- **LBFGS:** Limited memory Broyden Fletcher Goldfarb Shannon minimization.
- **SPSA**[11]**:** Based on simultaneous perturbation.
- **VersorRigid3DTransform:** Variant of a gradient descent optimizer for the VersorRigid3DTransform parameter space.

The desired optimizer can be chosen from a combo box. All parameters defining the selected optimizer have to be specified within the line edits and checkboxes underneath the optimizer combo box. To show the current interpolator for the registration process, just check the Interpolator checkbox. Currently, the following interpolators are implemented (for detailed information see [1] and [2]):

- **Linear:** Intensity varies linearly between grid positions.
- **NearestNeighbor:** Uses the intensity of the nearest grid position.

You can show and hide the parameters for the selection by checking or unchecking the corresponding area. You can save the current sets of parameters with the "Save as Testpreset" or "Save as Preset" buttons.

12.4.4 References:

1. L. Ibanez, W. Schroeder and K. Ng, The ITK Software Guide, Kitware Inc, New York, 2005.
2. `http://www.itk.org/Doxygen/`
3. Albert C.S. Chung, William M. Wells III, Alexander Norbash, and W. Eric L. Grimson, Multi-modal Image Registration by Minimising Kullback-Leibler Distance, In Medical Image Computing and Computer-Assisted Intervention - MICCAI 2002, LNCS 2489, pp. 525 - 532.
4. D. Mattes, D. R. Haynor, H. Vesselle, T. Lewellen and W. Eubank, "Nonrigid multimodality image registration", Medical Imaging 2001: Image Processing, 2001, pp. 1609-1620.
5. D. Mattes, D. R. Haynor, H. Vesselle, T. Lewellen and W. Eubank, "PET-CT Image Registration in the Chest Using Free-form Deformations", IEEE Transactions in Medical Imaging. Vol.22, No.1, January 2003, pp.120-128.
6. Viola, P. and Wells III, W. (1997). "Alignment by Maximization of Mutual Information" International Journal of Computer Vision, 24(2):137-154.
7. AP Zijdenbos, BM Dawant, RA Margolin , AC Palmer, Morphometric analysis of white matter lesions in MR images: Method and validation, IEEE Transactions on Medical Imaging, 13(4):716-724, Dec. 1994.
8. R. H. Byrd, P. Lu and J. Nocedal. A Limited Memory Algorithm for Bound Constrained Optimization, (1995), SIAM Journal on Scientific and Statistical Computing , 16, 5, pp. 1190-1208.
9. C. Zhu, R. H. Byrd and J. Nocedal. L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization (1997), ACM Transactions on Mathematical Software, Vol 23, Num. 4, pp. 550 - 560.
10. Martin Styner, G. Gerig, Christian Brechbuehler, Gabor Szekely, "Parametric estimate of intensity inhomogeneities applied to MRI", IEEE TRANSACTIONS ON MEDICAL IMAGING; 19(3), pp. 153-165, 2000.
11. Spall, J.C. (1998), "An Overview of the Simultaneous Perturbation Method for Efficient Optimization," Johns Hopkins APL Technical Digest, vol. 19, pp. 482-492.

Chapter 13

The Segmentation Plugin

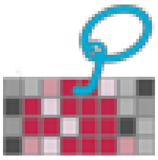


Figure 13.1: Icon of the Segmentation Plugin

Some of the features described below are closed source additions to the open source toolkit MITK and are not available in every application.

13.1 Overview

The **Segmentation plugin** allows you to create segmentations of anatomical and pathological structures in medical images of the human body. The plugin consists of a number of view which can be used for:

- manual and (semi-)automatic segmentation of organs on CT or MR image volumes via the **Segmentation View**
- segmentation postprocessing via the The Segmentation Utilities View
- clipping of existing segmentations using a resection plane via the The Clipping Plane

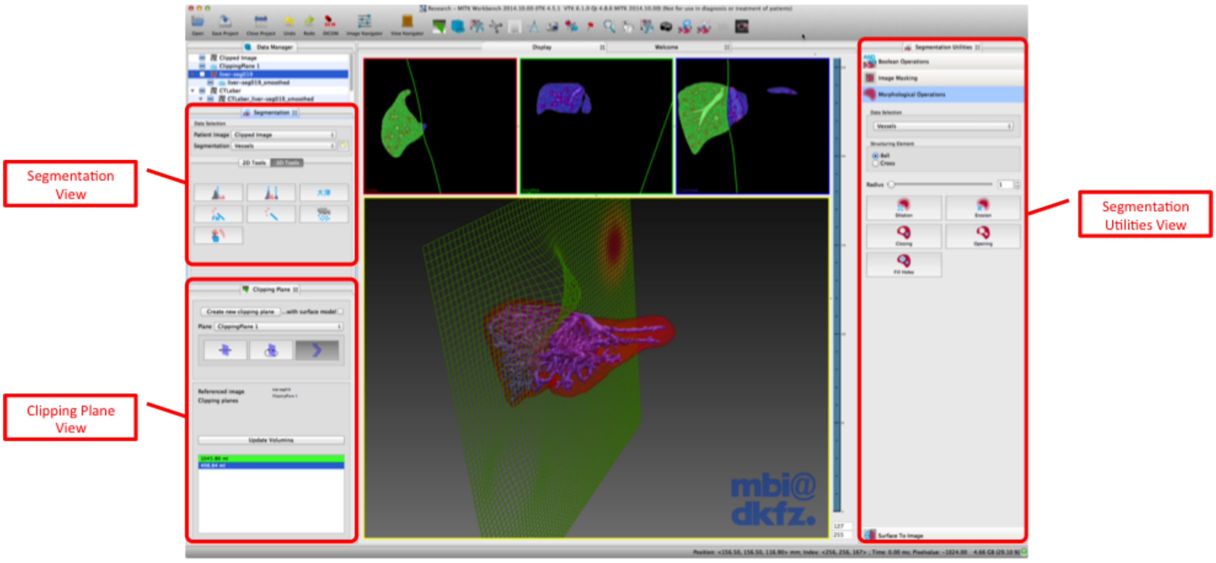
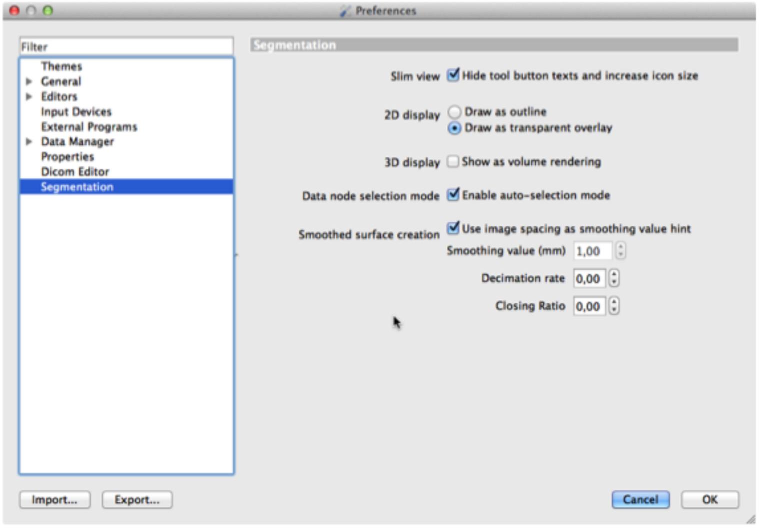


Figure 13.2: Segmentation Plugin consisting of the Segmentation View the Segmentation Utilities View and the Clipping Plane View

The segmentation plugin offers a number of preferences which can be set via the MITK Workbench application preference dialog:



Segmentation

Slim view ☑ Hide tool button texts and increase icon size

2D display ○ Draw as outline ● Draw as transparent overlay

3D display ☐ Show as volume rendering

Data node selection mode ☑ Enable auto-selection mode

Smoothed surface creation ☑ Use image spacing as smoothing value hint

Smoothing value (mm) 1,00

Decimation rate 0,00

Closing Ratio 0,00

Figure 13.3: Segmentation Plugin consisting of the Segmentation View the Segmentation Utilities View and the Clipping Plane View

The following preferences can be set:

- **Slim view:** Allows you to show or hide the tool button description of the Segmentation View
- **2D display:** Specify whether the segmentation is drawn as outline or as a transparent overlay
- **3D display:** Activate 3D volume rendering for your segmentation
- **Data node selection mode:** If activated the segmentation image combo box is always sychronized with the data manager selection.
- **Smoothed surface creation:** Set certain smoothing parameters for surface creation

If you wonder what segmentations are good for, we shortly revisit the concept of a segmentation here. A CT or MR image is made up of volume of physical measurements (volume elements are called voxels). In CT images, for example, the gray value of each voxel corresponds to the mass absorbtion coefficient for X-rays in this voxel, which is similar in many parts of the human body. The gray value does not contain any further information, so the computer does not know whether a given voxel is part of the body or the background, nor can it tell a brain from a liver. However, the distinction between a foreground and a background structure is required when:

- you want to know the volume of a given organ (the computer needs to know which parts of the image belong to this organ)
- you want to create 3D polygon visualizations (the computer needs to know the surfaces of structures that should be drawn)
- as a necessary pre-processing step for therapy planning, therapy support, and therapy monitoring

Creating this distinction between foreground and background is called *segmentation*. The Segmentation perspective of the MITK Workbench uses a voxel based approach to segmentation, i.e. each voxel of an image must be completely assigned to either foreground or background. This is in contrast to some other applications which might use an approach based on contours, where the border of a structure might cut a voxel into two parts.

The remainder of this document will summarize the features of the Segmentation perspective and how they are used.

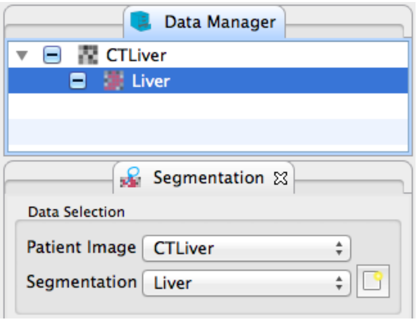
13.2 Technical Issues

The Segmentation perspective makes a number of assumptions. To know what this view can be used for, it will help you to know that:

- Images must be 2D, 3D, or 3D+t
- Images must be single-values, i.e. CT, MRI or "normal" ultrasound. Images from color doppler or photographic (RGB) images are not supported
- Segmentations are handled as binary images of the same extent as the original image

13.3 Image Selection

The Segmentation perspective makes use of the Data Manager view to give you an overview of all images and segmentations.



Patient Image: CTLiver
Segmentation: Liver

Figure 13.4: Data Manager is used for selecting the current segmentation. The reference image is selected in the drop down box of the control area.

To select the reference image (e.g. the original CT/MR image) use the patient image drop down box in the control area of the Segmentation view. The segmentation image selected in the Data Manager is displayed below in the segmentation drop down box. By default the auto selection mode is enabled, which always keeps the selection of the segmentation drop down box in synch with the selection in the data manager. If you disable the auto selection mode the selection of the right segmentation image has to be done via the drop down box. If no segmentation image exists or none is selected create a new segmentation image by using the "New segmentation" button on the right of the Segmentation drop down box. Some items of the graphical user interface might be disabled when no image is selected or the selected image does not fit to the patient image's geoemtry. In any case, the application will give you hints if a selection is needed.

13.4 Tool overview

MITK comes with a comprehensive set of segmentation tools. These tools can be differenciated between manual slice-based 2D segmentation tools and (semi-)automated 3D tools. The manual 2D tools require a big amount of user interaction and can only be applied to a single image slice whereas the 3D tools operate on the hole image. The 3D tools usually require a small amount of interaction like placin seedpoints of setting some parameters. You can switch between the different toolsets by switching the 2D/3D tab in the segmentation view.

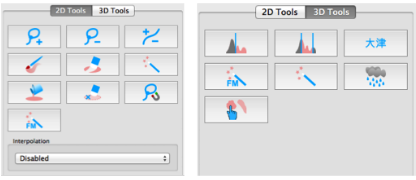


Figure 13.5: An overview of the existing tools in MITK. There are interactive 2D tools as well as (semi-)automated 3D tools

13.5 Manual Contouring

With manual contouring you define which voxels are part of the segmentation and which are not. This allows you to create segmentations of any structeres that you may find in an image, even if they are not part of the human body. You might also use manual contouring to correct segmentations that result from sub-optimal automatic methods. The drawback of manual contouring is that you might need to define contours on many 2D slices. However, this is moderated by the interpolation feature, which will make suggestions for a segmentation.

13.5.1 Creating New Segmentations

Unless you want to edit existing segmentations, you have to create a new, empty segmentation before you can edit it. To do so, click the "New manual segmentation" button. Input fields will appear where you can choose a name for the new segmentation and a color for its display. Click the checkmark button to confirm or the X button to cancel the new segmentation. Notice that the input field suggests names once you start typing and that it also suggests colors for known organ names. If you use names that are not yet known to the application, it will automatically remember these names and consider them the next time you create a new segmentation.

Once you created a new segmentation, you can notice a new item with the "binary mask" icon in the Data Manager tree view. This item is automatically selected for you, allowing you to start editing the new segmentation right away.

13.5.2 Selecting Segmentations for Editing

As you might want to have segmentations of multiple structures in a single patient image, the application needs to know which of them to use for editing. You select a segmenation by clicking it in the tree view of Data Manager. Note that segmentations are usually displayed as sub-items of "their" patient image. In the rare case, where you need to edit a segmentation that is not displayed as a a sub-item, you can click both the original image AND the segmentation while holding down CTRL or for Mac OS X the CMD on the keyboard.

When a selection is made, the Segmentation View will hide all but the selected segmentation and the corresponding original image. When there are multiple segmentations, the unselected ones will remain in the Data Manager, you can make them visible at any time by selecting them.

13.5.3 Selecting Editing Tools

If you are familiar with the MITK Workbench, you know that clicking and moving the mouse in any of the 2D render windows will move around the crosshair that defines what part of the image is displayed. This behavior is disabled while any of the manual segmentation tools are active – otherwise you might have a hard time concentrating on the contour you are drawing.

To start using one of the editing tools, click its button the the displayed toolbox. The selected editing tool will be active and its corresponding button will stay pressed until you click the button again. Selecting a different tool also deactivates the previous one.

If you have to delineate a lot of images, you should try using shortcuts to switch tools. Just hit the first letter of each tool to activate it (A for Add, S for Subtract, etc.).

13.5.4 Using Editing Tools

All of the editing tools work by the same principle: you use the mouse (left button) to click anywhere in a 2D window (any of the orientations axial, sagittal, or frontal), move the mouse while holding the mouse button and release to finish the editing action.

Multi-step undo and redo is fully supported by all editing tools. Use the application-wide undo button in the toolbar to revert erroneous actions.

Figure 13.6: Add and Subtract Tools

Use the left mouse button to draw a closed contour. When releasing the mouse button, the contour will be added (Add tool) to or removed from (Subtract tool) the current segmentation. Hold down the CTRL / CMD key to invert the operation (this will switch tools temporarily to allow for quick corrections).

Figure 13.7: Paint and Wipe Tools

Use the slider below the toolbox to change the radius of these round paintbrush tools. Move the mouse in any 2D window and press the left button to draw or erase pixels. As the Add/Subtract tools, holding CTRL / CMD while drawing will invert the current tool's behavior.

Figure 13.8: Region Growing Tool

Click at one point in a 2D slice widget to add an image region to the segmentation with the region growing tool. Moving up the cursor while holding the left mouse button widens the range for the included grey values; moving it down narrows it. When working on an image with a high range of grey values, the selection range can be influenced more strongly by moving the cursor at higher velocity.

Region Growing selects all pixels around the mouse cursor that have a similar gray value as the pixel below the mouse cursor. This enables you to quickly create segmentations of structures that have a good contrast to surrounding tissue, e.g. the lungs. The tool will select more or less pixels (corresponding to a changing gray value interval width) when you move the mouse up or down while holding down the left mouse button.

A common issue with region growing is the so called "leakage" which happens when the structure of interest is connected to other pixels, of similar gray values, through a narrow "bridge" at the border of the structure. The Region Growing tool comes with a "leakage detection/removal" feature. If leakage happens, you can left-click into the leakage region and the tool will try to automatically remove this region (see illustration below).

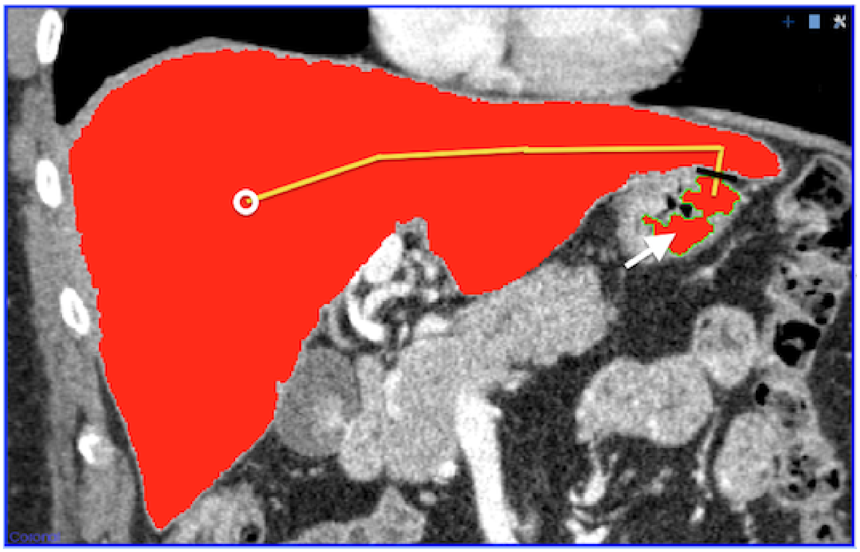


Figure 13.9: Leakage correction feature of the Region Growing tool

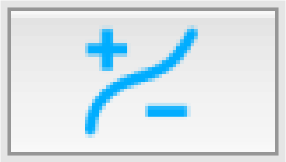


Figure 13.10: Correction Tool

You do not have to draw a closed contour to use the Correction tool and do not need to switch between the Add and

Substract tool to perform small corrective changes. The following figure shows the usage of this tool:

- if the user draws a line which starts and ends outside the segmenation AND it intersects no other segmentation the endpoints of the line are connected and the resulting contour is filled
- if the user draws a line which starts and ends outside the segmenation a part of it is cut off (left image)
- if the line is drawn fully inside the segmentation the marked region is added to the segmentation (right image)

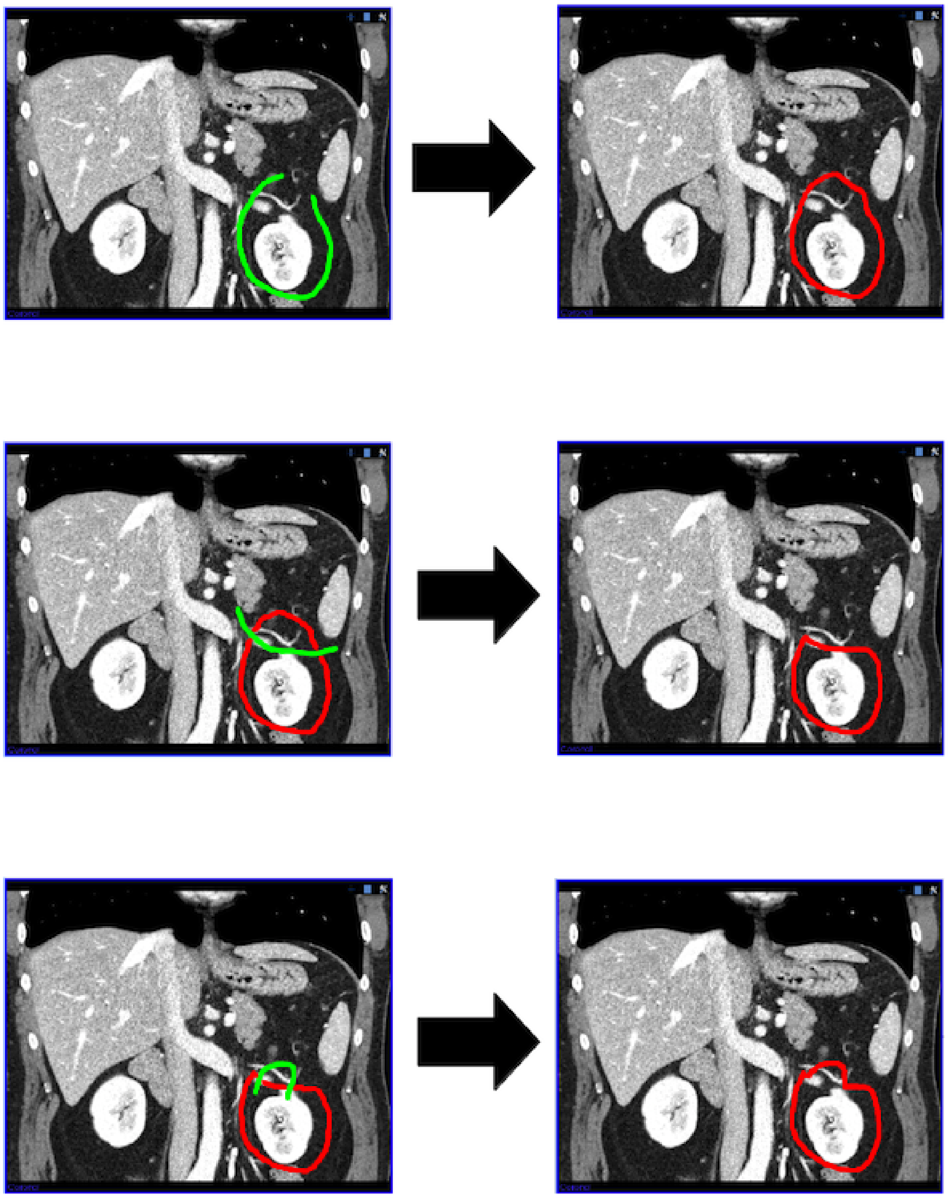


Figure 13.11: %Actions of the Correction tool illustrated.

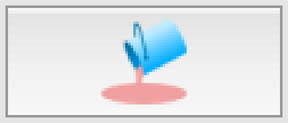


Figure 13.12: Fill Tool

Left-click inside a segmentation with holes to completely fill all holes.

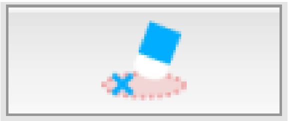


Figure 13.13: Erase Tool

This tool removes a connected part of pixels that form a segmentation. You may use it to remove so called islands (see picture) or to clear a whole slice at once (hold CTRL while clicking).

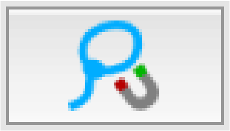


Figure 13.14: LiveWire Tool

The LiveWire Tool acts as a magnetic lasso with a contour snapping to edges of objects.

QmitkSegmentation_IMGLiveWireUsage.png

Figure 13.15: Steps for using LiveWire Tool

- (1) To start the Tool you have to double click near the edge of the object you want to segment. The initial anchor point will snap to the edge within a 3x3 region.
- (2) Move the mouse. You don't have trace the edge of the object. The contour will automatically snap to it.
- (3) To fix a segment you can set anchor points by single left mouse button click.
- (4) Go on with moving the mouse and setting anchor points.
- (5) To close the contour double click on the initial anchor point.
- (6) After closing the contour can be edited by moving, inserting and deleting anchor points.

The contour will be transfered to its binary image representation by deactivating the tool.

Figure 13.16: 2D Fast Marching Tool

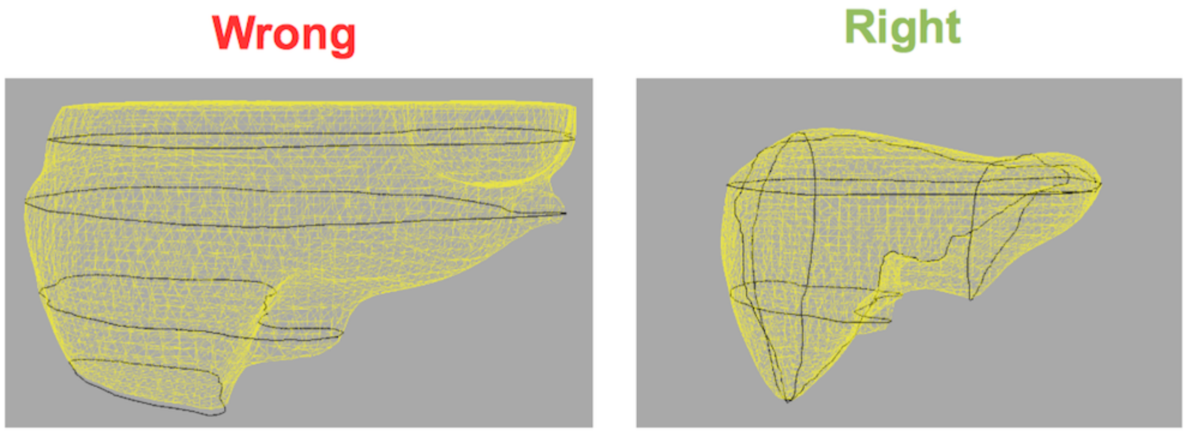
Provides a fast marching based 2D interaction segmentation tool. You start with setting seedpoints in an image slice. Via several sliders you can adapt parameters and see the fast marching result instantly.

13.5.5 Interpolation

Creating segmentations for modern CT volumes is very time-consuming, because structures of interest can easily cover a range of 50 or more slices. The Manual Segmentation View offers two helpful features for these cases:

- **3D Interpolation**
- **2D Interpolation**

The 3D interpolation is activated by default when using the manual segmentation tools. That means if you start contouring, from the second contour onwards, the surface of the segmented area will be interpolated based on the given contour information. The interpolation works with all available manual tools. Please note that this is currently a pure mathematical interpolation, i.e. image intensity information is not taken into account. With each further contour the interpolation result will be improved, but the more contours you provide the longer the recalculation will take. To achieve an optimal interpolation result and in this way a most accurate segmentation you should try to describe the surface with sparse contours by segmenting in arbitrary oriented planes. The 3D interpolation is not meant to be used for parallel slice-wise segmentation.



Parallel slice-wise segmentation

Segmentation in arbitrary slices

Figure 13.17: 3D Interpolation HowTo

You can accept the interpolation result by clicking the "Accept" - button below the tool buttons. In this case the 3D interpolation will be deactivated automatically so that the result can be postprocessed without any interpolation running in background. During recalculation the interpolated surface is blinking yellow/white. When the interpolation has finished the surface is shown yellow with a small opacity. Additional to the surface, black contours are shown in the 3D render window. They mark the positions of all the drawn contours which were used for the interpolation. You can navigate between the drawn contours by clicking on the „Position“ - Nodes in the datamanager which are located below the selected segmentation. If you don't want to see these nodes just unckeck the „Show Position Nodes“ Checkbox and these nodes will be hidden. If you want to delete a drawn contour we recommend to use the Erase-Tool since Redo/Undo is not yet working for 3D interpolation. The current state of the 3D interpolation can be saved accross application restart. Therefor just click on save project during the interpolation is active. After restarting the application and load your project you can click on "Reinit Interpolation" within the 3D interpolation GUI area.

The 2D Interpolation creates suggestions for a segmentation whenever you have a slice that

- has got neighboring slices with segmentations (these do not need to be direct neighbors but could also be a couple of slices away) AND
- is completely clear of a manual segmentation – i.e. there will be no suggestion if there is even only a single pixel of segmentation in the current slice.

Interpolated suggestions are displayed in a different way than manual segmentations are, until you "accept" them as part of the segmentation. To accept single slices, click the "Accept" button below the toolbox. If you have segmented a whole organ in every-x-slice, you may also review the interpolations and then accept all of them at once by clicking "... all slices".

13.6 3D Segmenation tools

The 3D tools operate on the hole image and require usually a small amount of interaction like placing seed-points or specifying certain parameters. All 3D tools provide an immediate segmentation feedback, which is displayed as a transparent green overlay. For accepting a preview you have to press the "Comfirm" button of the selected tool. The following 3D tools are at your disposal:

13.6.1 3D Threshold tool

The Thresholding tool simply applies a 3D threshold to the patient image. All pixels with values equal or above the selected threshold are labeled. You can change the threshold by either moving the slider of setting a certain value in the spinbox.

Figure 13.18: 3D Threshold tool

13.6.2 3D Upper/Lower Threshold tool

The Upper/Lower Thresholding tool works similar to the simple 3D threshold tool but allows you to define an upper and lower threshold. All pixels with values within this threshold intervall will be labeled

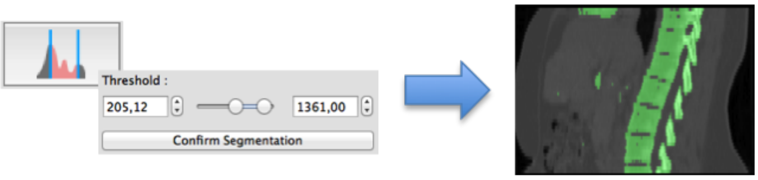


Figure 13.19: 3D Upper/Lower Threshold tool

13.6.3 3D Otsu tool

The 3D Otsu tool provides a more sophisticated thresholding algorithm. It allows you to define a number of regions. Based on the image histogram the pixels will then divided into different regions. There more regions you define the longer will the calculation take.

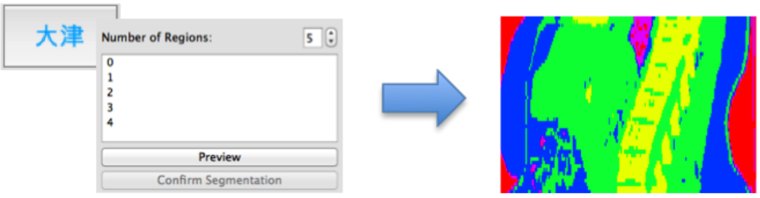

Number of Regions: 5

Figure 13.20: 3D Otsu tool

13.6.4 3D Fast Marching tool

The 3D Fast Marching tools works similar to the 2D pendant but on the hole image. Depending on you image's size the calculation will take some time. You can interactive set the parameters of the algorithm via the tool GUI.


Press shift-click to add seeds repeatedly.

Sigma: 0,80

Alpha: -4,68

Beta: 13,64

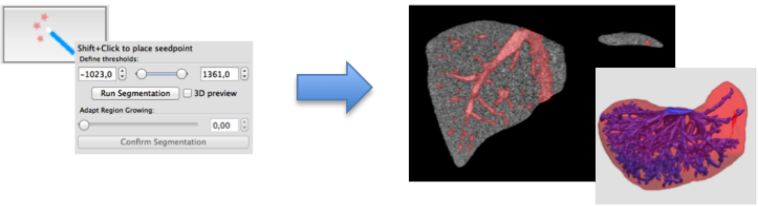
Stopping value: 2000,00

Threshold: -100 2000

Figure 13.21: 3D Fast Marching tool

13.6.5 3D Region Growing tool

The 3D Region Growing tool works similar to the 2D pendant. At the beginning you have to place a seedpoint and define a threshold intervall. If you press "Run segmentation" a preview is calculated, if the "3D preview" box is checked you will also see the result in 3D. By moving the "Adapt region growing slider" you can interactively adapt the result to you image.


Shift+Click to place seedpoint

Define thresholds: -1023,0 1361,0

☐ 3D preview

Adapt Region Growing: 0,00

Figure 13.22: 3D Region Growing tool

+

13.6.6 3D Watershed tool

This tool provides a watershed based segmentation algorithm.

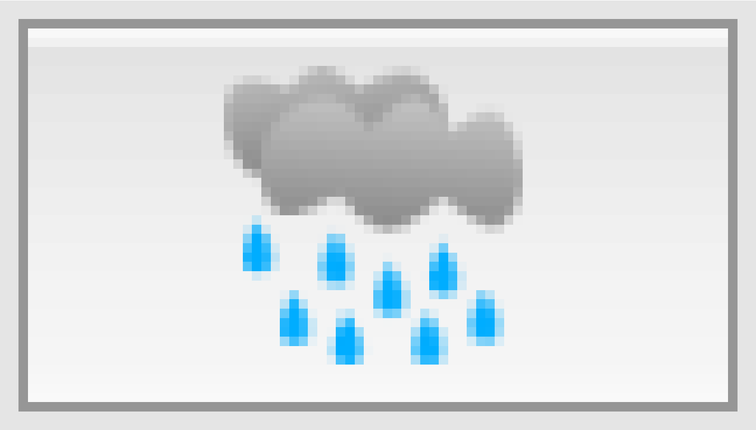


Figure 13.23: 3D Watershed tool

13.6.7 Picking tool

The Picking tool allows you to select islands within your segmentation. This is especially usefull if e.g. a thresholding delivered your several areas within your image but you are just interested in one special region.

Use shift click to pick a single region from the segmentation image.

Figure 13.24: Picking tool

13.7 Things you can do with segmentations

As mentioned in the introduction, segmentations are never an end in themselves. Consequently, the Segmentation view adds a couple of "post-processing" actions to the Data Manager. These actions are accessible through the context-menu of segmentations in Data Manager's list view

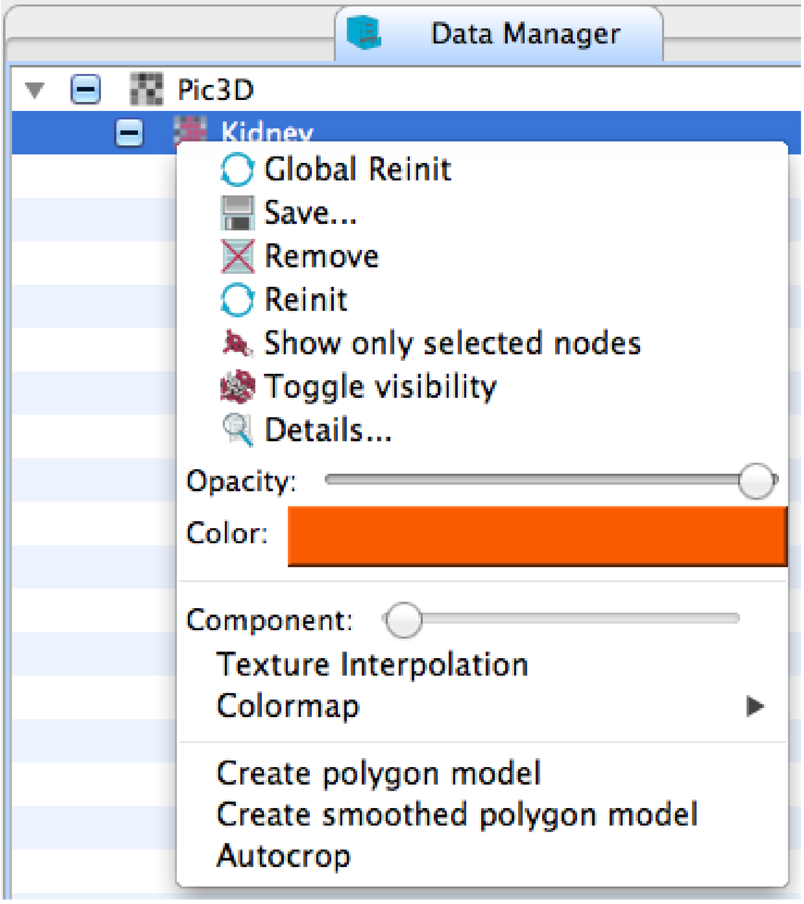


Figure 13.25: Context menu items for segmentations.

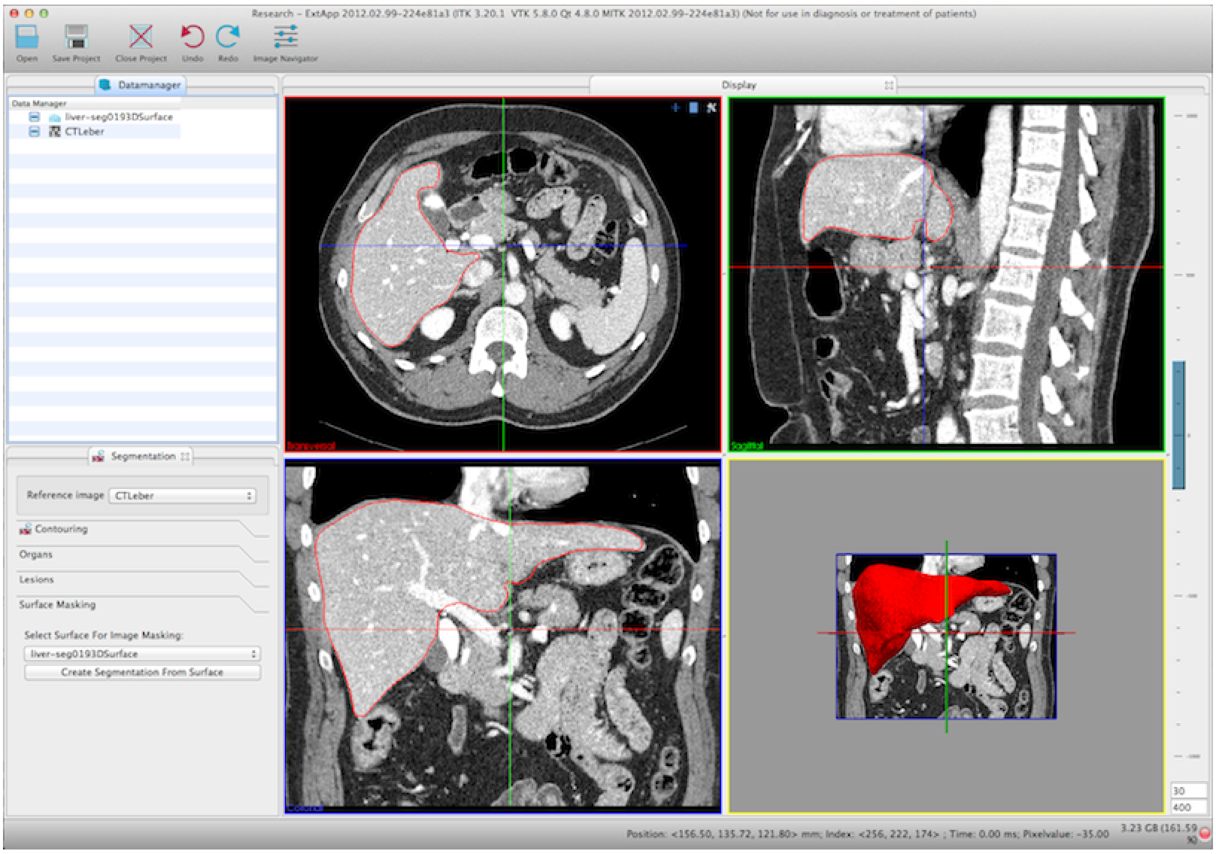
- **Create polygon model** applies the marching cubes algorithms to the segmentation. This polygon model can be used for visualization in 3D or other things such as stereolithography (3D printing).

- **Create smoothed polygon model** uses smoothing in addition to the marching cubes algorithms, which creates models that do not follow the exact outlines of the segmentation, but look smoother.

- **Autocrop** can save memory. Manual segmentations have the same extent as the patient image, even if the segmentation comprises only a small sub-volume. This invisible and meaningless margin is removed by autocropping.

13.8 Surface Masking

You can use the surface masking tool to create binary images from a surface which is used used as a mask on an image. This task is demonstrated below:



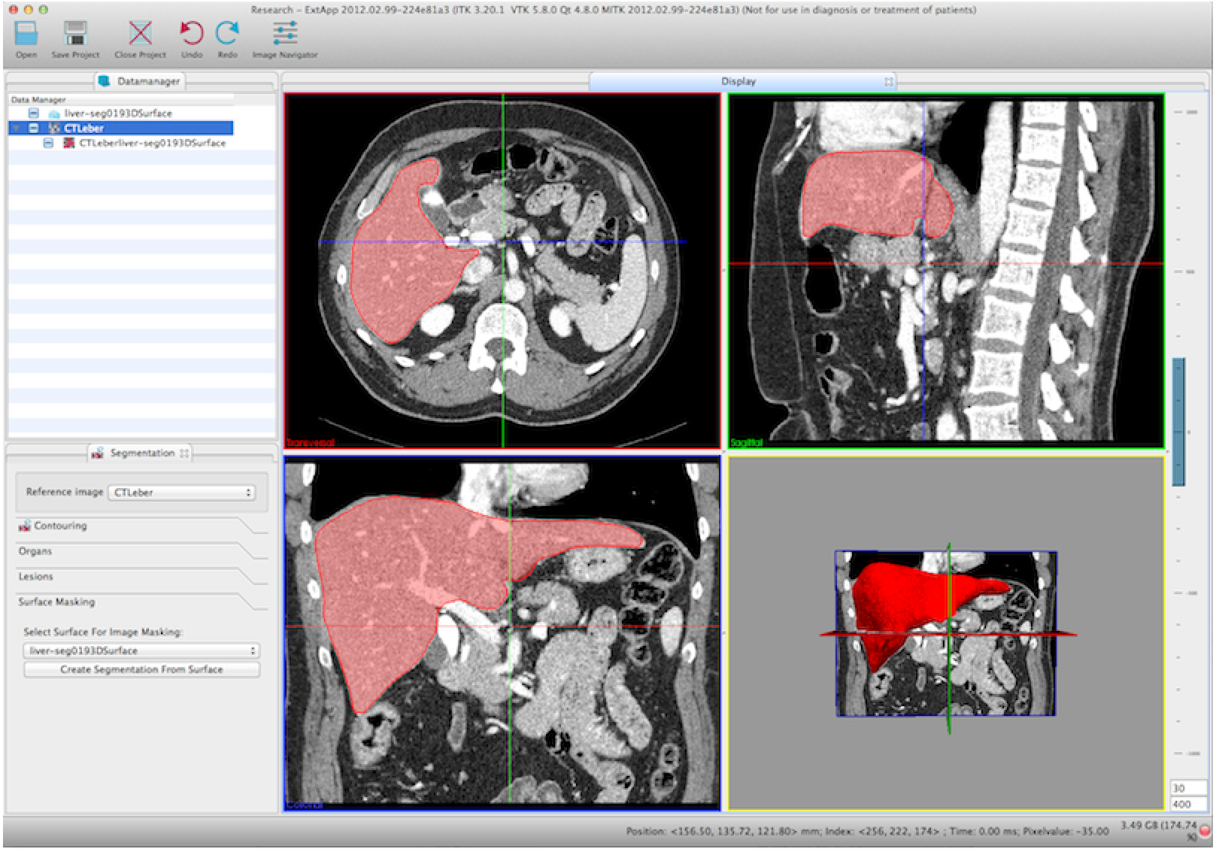
Reference image: CTLeber

Select Surface For Image Masking: liver-seg0193DSurface

Position: <156.50, 135.72, 121.80> mm; Index: <256, 222, 174> ; Time: 0.00 ms; Pixelvalue: -35.00

Figure 13.26: Load an image and a surface.

Select the image and the surface in the corresponding drop-down boxes (both are selected automatically if there is just one image and one surface)



Research - ExtApp 2012.02.99-224e81a3 (ITK 3.20.1 VTK 5.8.0 Qt 4.8.0 MITK 2012.02.99-224e81a3) (Not for use in diagnosis or treatment of patients)

Reference image CTLeber

Select Surface For Image Masking: liver-seg0193DSurface

Position: <156.50, 135.72, 121.80> mm; Index: <256, 222, 174> ; Time: 0.00 ms; Pixelvalue: -35.00

Figure 13.27: Create segmentation from surface

After clicking "Create segmentation from surface" the newly created binary image is inserted in the DataManager and can be used for further processing

13.9 Technical Information for Developers

For technical specifications see Technical design of QmitkSegmentation and for information on the extensions of the tools system How to extend the Segmentation view with external tools .

13.10 The Segmentation Utilities View

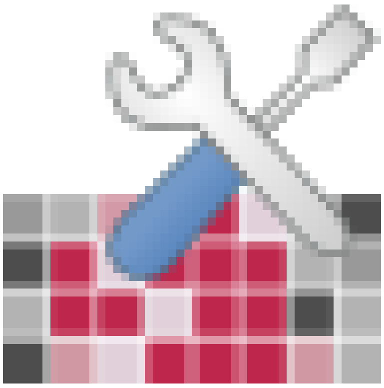
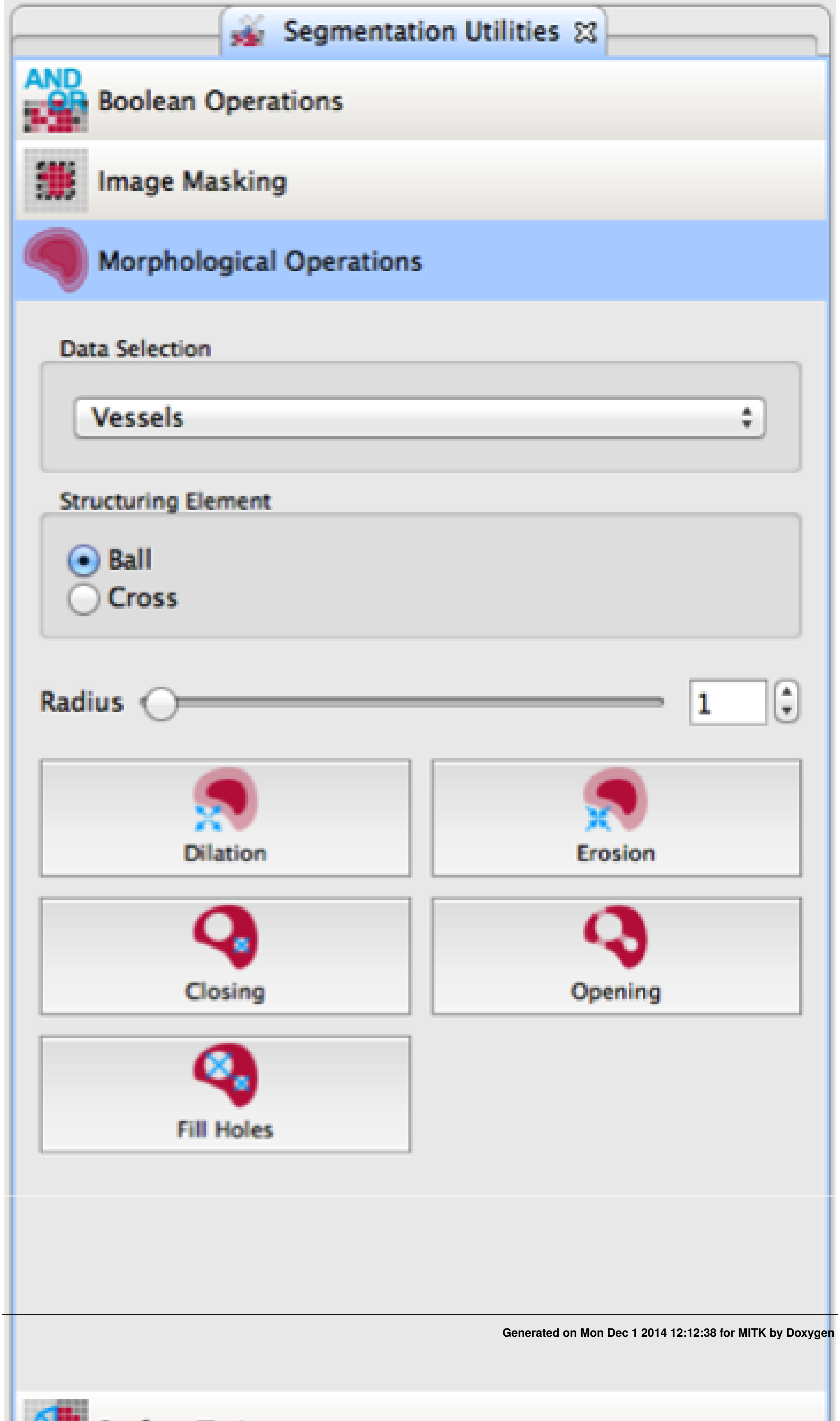


Figure 13.28: Icon of the Segmentation Utilities View



Data Selection
Vessels

Structuring Element
● Ball
○ Cross

Radius 1

13.10.1 Overview

The **Segmentation Utilities View** allows you to postprocess existing segmentations

13.10.2 Image Selection

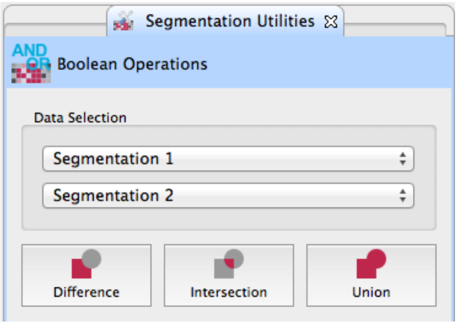
Usually the data selection in the Segmentation Utilities View is done via drop down box which let you just select the appropriate data.

13.10.3 Boolean Operations

Boolean operations allow you to create the

- **Union:** Combines two existing segmentations
- **Intersection:** Keeps just the overlapping areas of two existing segmentations
- **Difference:** Subtracts one segmentation from the other

of two segmentations. The selected segmentations must have the same geometry (size, spacing, ...)



Segmentation 1

Segmentation 2

Figure 13.30: Boolean operations of the SegmentationUtlitiesView

13.10.4 Image masking

You can mask your grey value image with either an existing segmentation or a surface. The result will be an image containing only the pixels that are cover by the respective mask.

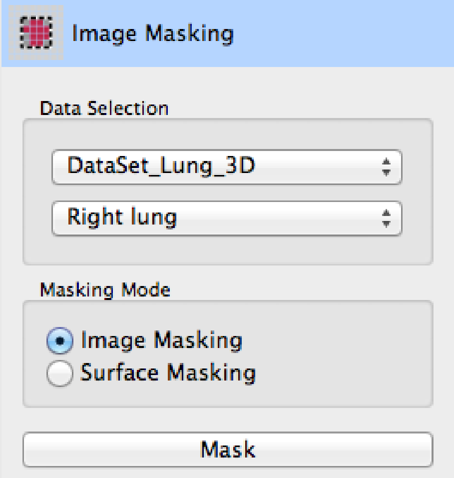


Figure 13.31: Image masking widget of the Segmentation Utilities View

13.10.5 Morphological Operators

The morphological operators are applied to a single segmentation image. Based on a given structuring element the underlying segmentation will be modfied. MITK provides a **ball** and a **cross** as structuring elements. The follow operators are at your disposal:

- **Dilation:** Each labeled pixel within the segmentation will be dilated based on the selected structuring element
- **Erosion:** Each labeled pixel within the segmentation will be eroded based on the selected structuring element
- **Opening:** A dilation followed by an erosion, used for smoothing edges or eliminating small objects
- **Closing** An erosion followed by an dilation, used for filling small holes
- **Fill Holes** Fills bigger holes within a segmentation

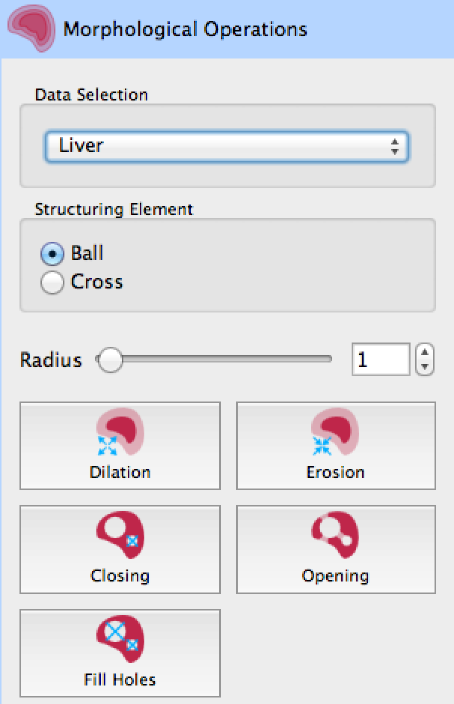


Figure 13.32: Morphological operators widget of the Segmentation Utilities View

13.10.6 Surface to binary image

This widget lets you fill you meshes into an empty binary image. It is required that a reference grey value image is present. The created binary image will have the same geometrical properties like the reference image

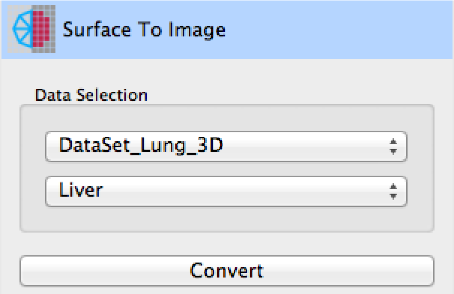


Figure 13.33: Surface to image widget of the Segmentation Utilities View

13.11 The Clipping Plane

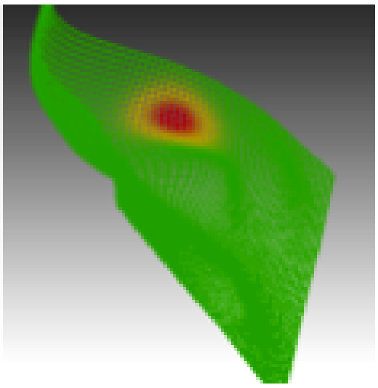
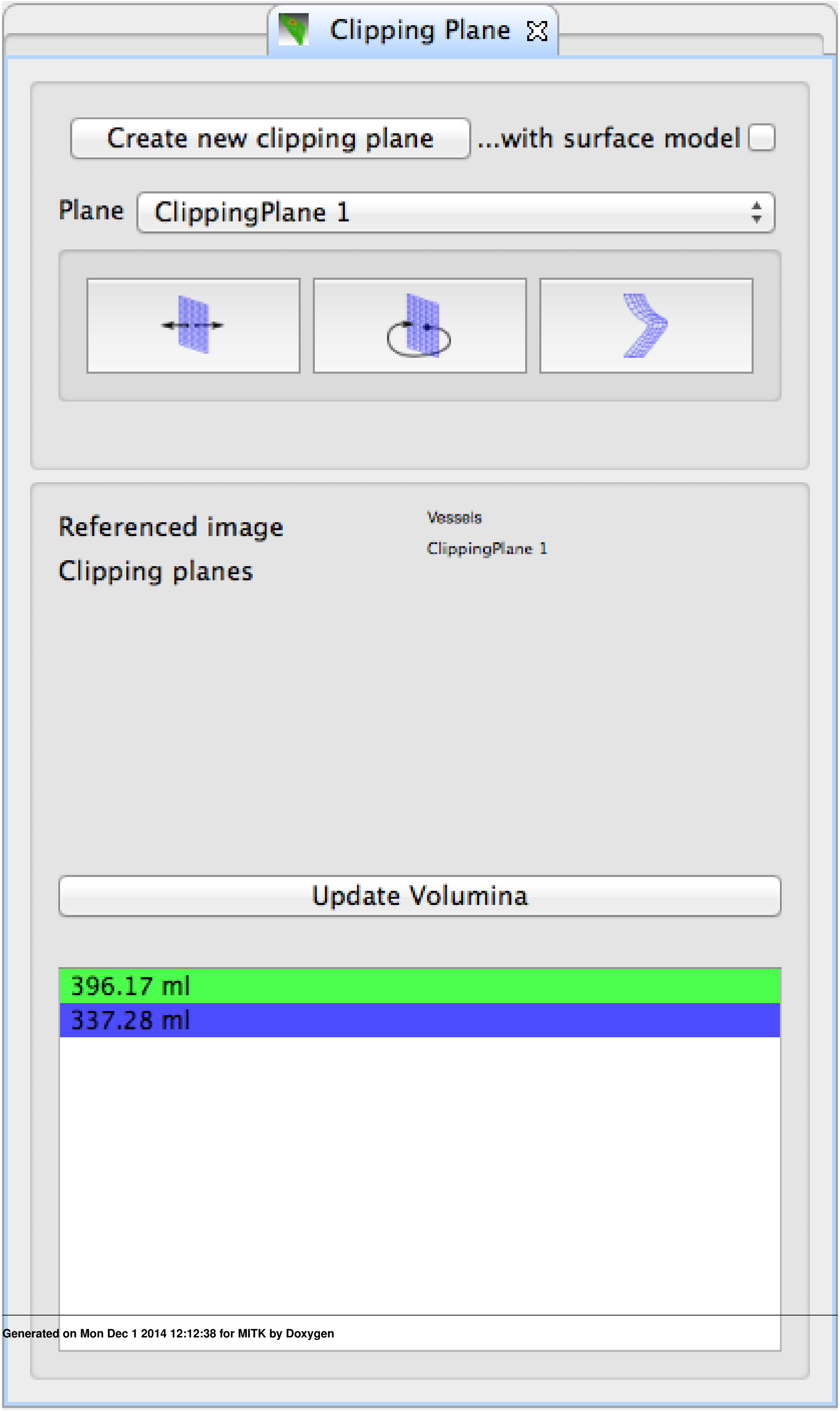


Figure 13.34: Icon of the Clipping Plane Plugin



Plane ClippingPlane 1

Referenced image Vessels

Clipping planes ClippingPlane 1

396.17 ml
337.28 ml

13.11.1 Overview

The **Clipping Plane view** allows you to create clipping planes and calculate the volumina of the devided parts.

13.11.2 Technical Issue

To use the Update Volumina function your image should be binary.

13.11.3 Image Selection

The Clipping Plane view makes use of the Data Manager view to give you an overview of all images, segmentations and clipping planes.

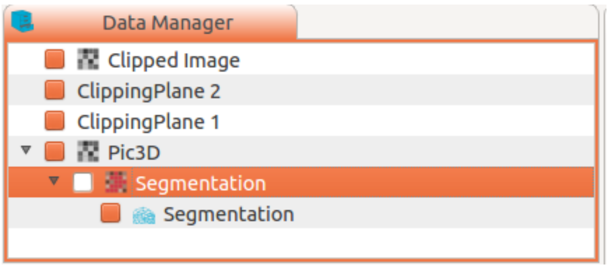


Figure 13.36: Data Manager is used for selecting the current clipping plane. The reference plane is selected in the drop down box of the control area.

To select the reference plane use the drop down box in the control area of the Clipping Plane view or the Data Manager. The clipping plane selected in the Data Manager is displayed below the drop down box. If no clipping plane exists or none is selected create a new clipping plane by using the "Create new clipping plane" button. Some items of the graphical user interface might be disabled when no plane is selected. In any case, the application will give you hints if a selection is needed.

13.11.4 Creating New Clipping Plane

If you want to create a new clipping plane select an image from the Data Manager and press the button "Create new clipping plane". Optionally you can enable the "...with surface model" option.

13.11.5 Interaction with the planes

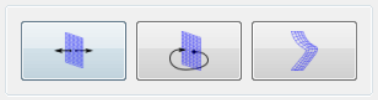


Figure 13.37: The interaction buttons

You have different options to interact with the clipping planes:

13.11.5.1 Translation

In Translation mode you can change the position of the clipping plane.

- Click the Translation Button
- Move mouse over the selected clipping plane (the plane changes its color from blue to green)
- Hold mouse-left button and move the mouse orthogonally to the plane

13.11.5.2 Rotation

In Rotation mode you can change the angle of the clipping plane.

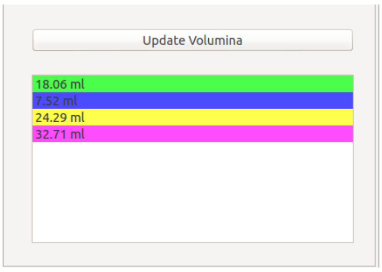
- Click the Rotation Button
- Move mouse over the selected clipping plane (the plane changes its color from blue to green)
- Hold mouse-left button and move the mouse in the direction it should be rotated

13.11.5.3 Deformation

In Deformation mode you can change the surface of the clipping plane.

- Click the Deformation Button
- Move mouse over the selected clipping plane (the plane changes its color from blue to green). The deformation area is highlighted in red and yellow.
- On mouse-scrolling you can change the size of the deformation area (Scroll-Down = smaller / Scroll-Up = bigger).
- Hold mouse-left button and move the mouse orthogonally to the plane to deformate the plane

13.11.6 Update Volumina



18.06 ml
7.52 ml
24.29 ml
32.71 ml

Figure 13.38: The 'Update Volumina' button

Calculating the volumina of the segmentation parts, which are devided by the clipping plane(s).

- Create a segmentation (see Segmentation-Manual)
- Create one or more clipping plane(s)
- Use the interaction buttons (Translation, Rotation, Deformation) to adjust the clipping plane for intersecting the segmentation
- (You can decide which planes shouldnt be included for the calculation by changing their visibility to invisible)
- Click button "Update Volumina" button
- The intersected parts are displayed in different colors and their volumina are shown beneath the "Update Volumina" button

13.12 Technical design of QmitkSegmentation

13.12.1 Introduction

QmitkSegmentation was designed for the liver resection planning project "ReLiver". The goal was a stable, well-documented, extensible, and testable re-implementation of a functionality called "ERIS", which was used for manual segmentation in 2D slices of 3D or 3D+t images. Re-implementation was chosen because it seemed to be easier to write documentation and tests for newly developed code. In addition, the old code had some design weaknesses (e.g. a monolithic class), which would be hard to maintain in the future.

By now Segmentation is a well tested and easily extensible vehicle for all kinds of interactive segmentation applications. A separate page describes how you can extend Segmentation with new tools in a shared object (DLL): How to extend the Segmentation view with external tools.

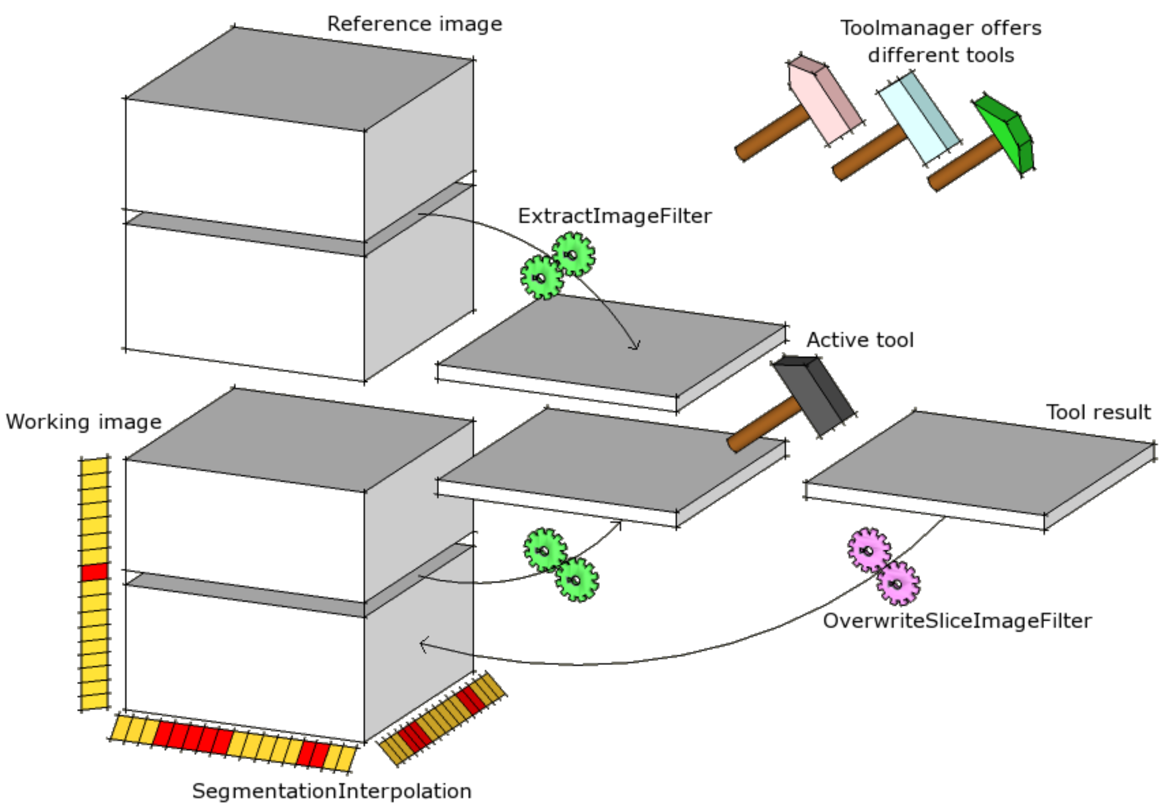
13.12.2 Overview of tasks

We identified the following major tasks:

1. **Management of images**: what is the original patient image, what images are the active segmentations?
2. **Management of drawing tools**: there is a set of drawing tools, one at a time is active, that is, someone has to decide which tool will receive mouse (and other) events.
3. **Drawing tools**: each tool can modify a segmentation in reaction to user interaction. To do so, the tools have to know about the relevant images.
4. **Slice manipulation**: drawing tools need to have means to extract a single slice from an image volume and to write a single slice back into an image volume.
5. **Interpolation of unsegmented slices**: some class has to keep track of all the segmentations in a volume and generate suggestions for missing slices. This should be possible in all three orthogonal slice direction.
6. **Undo**: Slice manipulations should be undoable, no matter whether a tool or the interpolation mechanism changed something.
7. **GUI**: Integration of everything.

13.12.3 Classes involved

The above blocks correspond to a number of classes. Here is an overview of all related classes with their responsibilities and relations:



1. **Management of images**: mitk::ToolManager has a set of reference data (original images) and a second set of working data (segmentations). mitk::Tool objects know a ToolManager and can ask the manager for the currently relevant images. There are two GUI elements that enable the user to modify the set of reference and working images (QmitkToolReferenceDataSelectionBox and QmitkToolWorkingDataSelectionBox). GUI and non-GUI classes are coupled by itk::Events (non-GUI to GUI) and direct method calls (GUI to non-GUI).
2. **Management of drawing tools**: As a second task, ToolManager manages all available tools and makes sure that one at a time is able to receive MITK events. The GUI for selecting tools is implemented in QmitkToolSelectionBox.
3. **Drawing tools**: Drawing tools all inherit from mitk::Tool, which is a mitk::StateMachine. There is a number of derivations from Tool, each offering some helper methods for specific sub-classes, like manipulation of 2D slices. Tools are instantiated through the itk::ObjectFactory, which means that there is also one factory for each tool (e.g. mitk::AddContourToolFactory). For the GUI representation, each tool has an identification, consisting of a name and an icon (XPM). The actual drawing methods are mainly implemented in mitk::SegTool2D (helper methods) and its sub-classes for region growing, freehand drawing, etc.
4. **Slice manipulation**: There are two filters for manipulation of slices inside a 3D image volume. mitk::ExtractImageFilter retrieves a single 2D slice from a 3D volume. mitk::OverwriteSliceImageFilter replaces a slice inside a 3D volume with a second slice which is a parameter to the filter. These classes are used extensively by most of the tools to fulfill their task. mitk::OverwriteSliceImageFilter cooperates with the interpolation classes to inform them of single slice modifications.
5. **Interpolation of unsegmented slices**: There are two classes involved in interpolation: mitk::SegmentationInterpolationController knows a mitk::Image (the segmentation) and scans its contents for slices with non-zero pixels. It keeps track of changes in the image and is always able to tell, which neighbors of a slice (in the three orthogonal slice directions) contain segmentations. The class also performs this interpolation for single slices on demand. Again, we have a second class responsible for the GUI: QmitkSlicesInterpolator enables/disables interpolation and offers to accept interpolations for one or all slices.
6. **Undo**: Undo functionality is implemented in mitk::OverwriteSliceImageFilter, since this is the central place where all image modifications are made. The filter stores a binary difference image to the undo stack as a

mitk::ApplyDiffImageOperation. When the user requests undo, this ApplyDiffImageOperation will be executed by a singleton class DiffImageApplier. The operation itself observes the image, which it refers to, for itk::↩ DeleteEvent, so no undo operation will be executed on/for images that have already been destroyed.

7. **GUI**: The top-level GUI is the functionality QmitkSegmentation, which is very thin in comparison to ERI↩ S. There are separate widgets for image and tool selection, for interpolation. Additionaly, there are some methods to create, delete, crop, load and save segmentations.

13.13 How to extend the Segmentation view with external tools

- Introduction
- What might be part of an extension
  - Tool classes
  - GUI classes for tools
  - Additional files
- Writing a CMake file for a tool extension
- Compiling the extension
- Configuring ITK autoload

13.13.1 Introduction

The application for manual segmentation in MITK (Segmentation view) comes with a tool class framework that is extensible with new tools (description at Technical design of QmitkSegmentation). The usual way to create new tools (since it is mostly used inside DKFZ) is to just add new files to the MITK source code tree. However, this requires to be familiar with the MITK build system and turnaround time during development might be long (recompiling parts of MITK again and again).

For external users who just want to use MITK as a library and application, there is a way to create new segmentation tools in an MITK external project, which will compile the new tools into a shared object (DLL). Such shared objects can be loaded via the ITK object factory and its autoload feature on application startup. This document describes how to build such external extensions.

Example files can be found in the MITK source code in the directory `${MITK_SOURCE_DIR}/Q↩ Applications/ToolExtensionsExample/`.

13.13.2 What might be part of an extension

The extension concept assumes that you want to create one or several new interactive segmentation tools for Segmentation or another MITK functionality that uses the tools infrastructure. In the result you will create a shared object (DLL), which contains several tools and their GUI counterparts, plus optional code that your extension requires. The following sections shortly describe each of these parts.

13.13.2.1 Tool classes

A tool is basically any subclass of mitk::Tool. Tools are created at runtime through the ITK object factory (so they inherit from itk::Object). Tools should handle the interaction part of a segmentation method, i.e. create seed points, draw contours, etc., in order to parameterize segmentation algorithms. Simple algorithms can even be part of a tool. A tools is identified by icon (XPM format), name (short string) and optionally a group name (e.g. the group name for Segmentation is "default").

There is a naming convention: you should put a tool called `mitk::ExternalTool` into files called `mitk↩ ExternalTool.h` and `mitkExternalTool.cpp`. This is *required* if you use the convenience macros described below, because there need to be ITK factories, which names are directly derived from the file names of the tools. For the example of mitk::ExternalTool there would be a factory called `mitk::ExternalToolFactory` in a file named `mitkExternalToolFactory.cpp`.

13.13.2.2 GUI classes for tools

Tools are non-graphical classes that only implement interactions in renderwindows. However, some tools will need a means to allow the user to set some parameters – a graphical user interface, GUI. In the Qt3 case, tool GUIs inherit from QmitkToolGUI, which is a mixture of QWidget and itk::Object. Tool GUIs are also created through the ITK object factory.

Tools inform their GUIs about state changes by messages. Tool GUIs communicate with their associated tools via direct method calls (they know their tools). See mitk::BinaryThresholdTool for examples.

Again a naming convention: if the convenience macros for tool extension shared objects are used, you have to put a tool GUI called `QmitkExternalToolGUI` into a files named `QmitkExternalToolGUI.cpp` and `QmitkExternalToolGUI.h`. The convenience macro will create a factory called `QmitkExternalToolGUIFactory` into a file named `QmitkExternalToolGUIFactory.cpp`.

13.13.2.3 Additional files

If you are writing tools MITK externally, these tools might depend on additional files, e.g. segmentation algorithms. These can also be compiled into a tool extension shared object.

13.13.3 Writing a CMake file for a tool extension

Summing up the last section, an example tool extension could comprise the following files:

```
mitkExternalTool.h       \
mitkExternalTool.png      >--- implementing mitk::ExternalTool (header, icon, implementation)
mitkExternalTool.cpp     /

QmitkExternalToolGUI.h     ,-- implementing a GUI for mitk::ExternalTool
QmitkExternalToolGUI.cpp /

externalalgorithm.h          \
externalalgorithm.cpp         \
externalalgorithmsolver.h      >-- a couple of files (not related to MITK tools)
externalalgorithmsolver.cpp   /
```

This should all be compiled into one shared object. Just like ITK, VTK and MITK we will use CMake for this purpose (I assume you either know or are willing to learn about www.cmake.org) A CMake file for the above example would look like this:

```
project( ExternalTool )

find_package(ITK)
find_package(MITK)
find_package(Qt3)

add_definitions(${QT_DEFINITIONS})

set( TOOL_QT3GUI_FILES
     QmitkExternalToolGUI.cpp
   )

set( TOOL_FILES
     mitkExternalTool.cpp
   )

set( TOOL_ADDITIONAL_CPPS
     externalalgorithm.cpp
     externalalgorithmsolver.cpp
   )

set( TOOL_ADDITIONAL_MOC_H
)

MITK_GENERATE_TOOLS_LIBRARY(mitkExternalTools)
```

Basically, you only have to change the definitions of `TOOL_FILES` and, optionally, `TOOL_QT3GUI_FILES`, `TOOL_ADDITIONAL_CPPS` and `TOOL_ADDITIONAL_MOC_H`. For all .cpp files in `TOOL_FILES` and `TOOL_QT3GUI_FILES` there will be factories created assuming the naming conventions described in the sections above. Files listed in `TOOL_ADDITIONAL_CPPS` will just be compiled. Files listed in `TOOL_ADDITIONAL_MOC_H` will be run through Qts meta object compiler `moc` – this is neccessary for all objects that have the Q_OBJECT macro in their declaration. `moc` will create new files that will also be compiled into the library.

13.13.4 Compiling the extension

For compiling a tool extension, you will need a compiled version of MITK. We will assume MITK was compiled into /home/user/mitk/debug. You need to build MITK with **BUILD_SHARED_CORE** turned on!

You build the tool extension just like any other CMake based project:

- know where your source code is (e.g. /home/user/mitk/tool-extension-src)
- change into the directory, where you want to compile the shared object (e.g. /home/user/mitk/tool-extension-debug)
- invoke cmake: `ccmake /home/user/mitk/tool-extension-src`
- configure (press c or the "configure" button)
- set the ITK_DIR variable to the directory, where you compiled ITK
- set the MITK_DIR variable to the directory, where you compiled MITK: `/home/user/mitk/debug`
- configure (press "c" or the "configure" button)
- generate (press "g" or the "generate" button)

This should do it and leave you with a or project file or Makefile that you can compile (using make or VisualStudio).

13.13.5 Configuring ITK autoload

If the compile succeeds, you will get a library mitkExternalTools.dll or libmitkExternalTools.so. This library exports a symbol `itkLoad` which is expected by the ITK object factory.

On application startup the ITK object factory will search a list of directories from the environment variable `ITK_AUTOLOAD_PATH`. Set this environment variable to your binary directory (`/home/user/mitk/tool-extension-debug`).

The ITK object factory will load all shared objects that it finds in the specified directories and will test if they contain a symbol (function pointer) `itkLoad`, which is expected to return a pointer to a itk::ObjectFactoryBase instance. If such a symbol is found, the returned factory will be registered with the ITK object factory.

If you successfully followed all the steps above, MITK will find your mitk::ExternalTool on application startup, when the ITK object factory is asked to create all known instances of mitk::Tool. Furthermore, if your mitk::ExternalTool claims to be part of the "default" group, there will be a new icon in Segmentation, which activates your tool.

Have fun! (And Windows users: welcome to the world of DLLs)

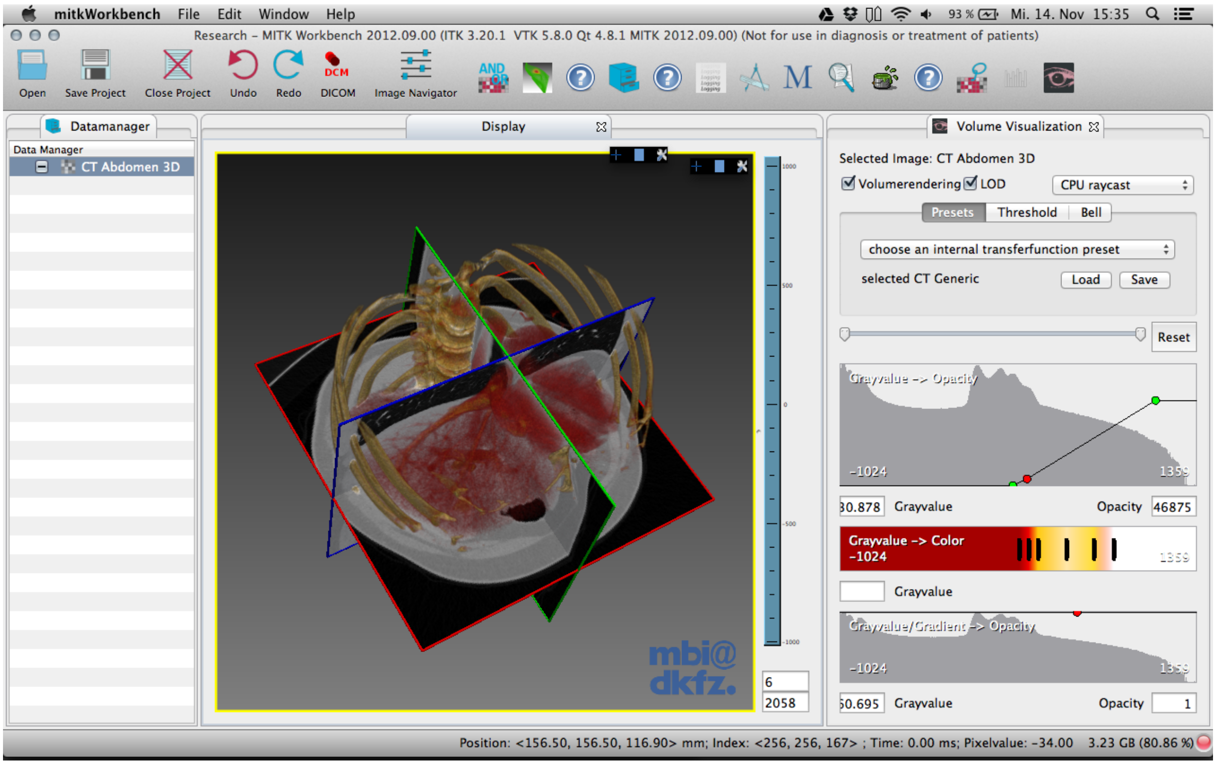
Chapter 14

The Volume Visualization Plugin

Figure 14.1: Icon of the Volume Visualization Plugin

14.1 Overview

The **Volume Visualization Plugin** is a basic tool for visualizing three dimensional medical images. MITK provides generic transfer function presets for medical CT data. These functions, that map the gray-value to color and opacity, can be interactively edited. Additionally, there are controls to quickly generate common used transfer function shapes like the threshold and bell curve to help identify a range of grey-values.



Research – MITK Workbench 2012.09.00 (ITK 3.20.1 VTK 5.8.0 Qt 4.8.1 MITK 2012.09.00) (Not for use in diagnosis or treatment of patients)

Selected Image: CT Abdomen 3D

☑ Volumerendering ☑ LOD — CPU raycast

choose an internal transferfunction preset

selected CT Generic

30.878 Grayvalue — Opacity 46875

50.695 Grayvalue — Opacity 1

Position: <156.50, 156.50, 116.90> mm; Index: <256, 256, 167> ; Time: 0.00 ms; Pixelvalue: -34.00 3.23 GB (80.86 %)

14.2 Enable Volume Rendering

14.2.1 Loading an image into the application

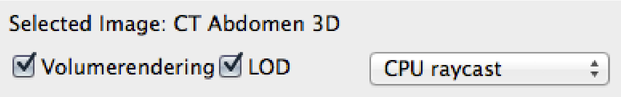
Load an image into the application by

- dragging a file into the application window.
- selecting file / load from the menu.

Volume Visualization imposes following restrictions on images:

- It has to be a 3D-Image Scalar image, that means a normal CT or MRT.
- 3D+T are supported for rendering, but the histograms are not computed.
- Also be aware that volume visualization requires a huge amount of memory. Very large images may not work, unless you use the 64bit version.

14.2.2 Enable Volumerendering


Selected Image: CT Abdomen 3D

☑ Volumerendering ☑ LOD CPU raycast

Select an image in datamanager and click on the checkbox left of "Volumerendering". Please be patient, while the image is prepared for rendering, which can take up to a half minute.

14.2.3 The LOD & GPU checkboxes

Volume Rendering requires a lot of computing resources including processor, memory and graphics card.

To run volume rendering on smaller platforms, enable the LOD checkbox (level-of-detail rendering). Level-of-detail first renders a lower quality preview to increase interactivity. If the user stops to interact a normal quality rendering is issued.

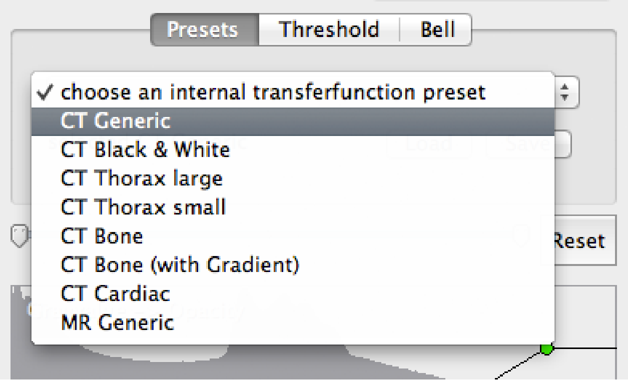
The GPU checkbox tries to use computing resources on the graphics card to accelerate volume rendering. It requires a powerful graphics card and OpenGL hardware support for shaders, but achieves much higher frame rates than software-rendering.

14.3 Applying premade presets

14.3.1 Internal presets

There are some internal presets given, that can be used with normal CT data (given in Houndsfield units). A large set of medical data has been tested with that presets, but it may not suit on some special cases.

Click on the "Preset" tab for using internal or custom presets.



- "CT Generic" is the default transferfunction that is first applied.
- "CT Black&White" does not use any colors, as it may be distracting on some data.
- "CT Cardiac" tries to increase detail on CTs from the heart.
- "CT Bone" emphasizes bones and shows other areas more transparent.
- "CT Bone (Gradient)" is like "CT Bone", but shows from other organs only the surface by using the gradient.
- "MR Generic" is the default transferfunction that we use on MRT data (which is not normalized like CT data).
- "CT Thorax small" tries to increase detail.
- "CT Thorax large" tries to increase detail.

14.3.2 Saving and loading custom presets

After creating or editing a transferfunction (see Customize transferfunctions in detail or Interactively create transferfunctions), the custom transferfunction can be stored and later retrieved on the filesystem. Click "Save" (respectively "Load") button to save (load) the threshold-, color- and gradient function combined in a single .xml file.

14.4 Interactively create transferfunctions

Beside the possibility to directly edit the transferfunctions (Customize transferfunctions in detail), a one-click generation of two commonly known shapes is given.

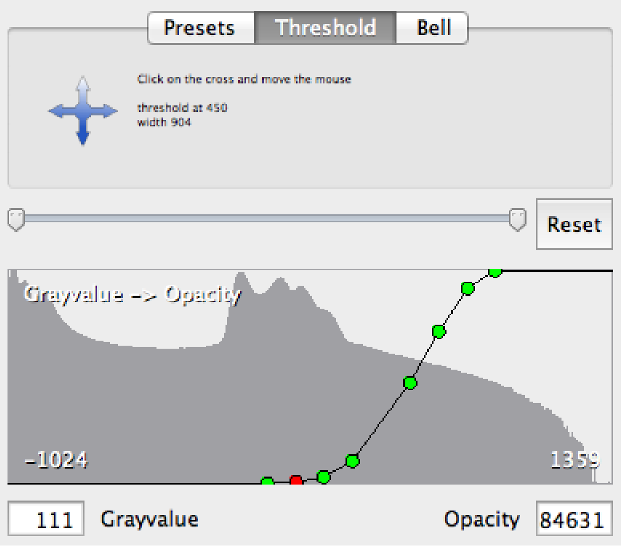
Both generators have two parameters, that can be modified by first clicking on the cross and then moving the mouse up/down and left/right.

The first parameter "center" (controlled by horizontal movement of the mouse) specifies the gravalue where the center of the shape will be located.

The second parameter "width" (controlled by vertical movement of the mouse) specifies the width (or steepness) of the shape.

14.4.1 Threshold

Click on the "Threshold" tab to active the threshold function generator.



Click on the cross and move the mouse

threshold at 450
width 904

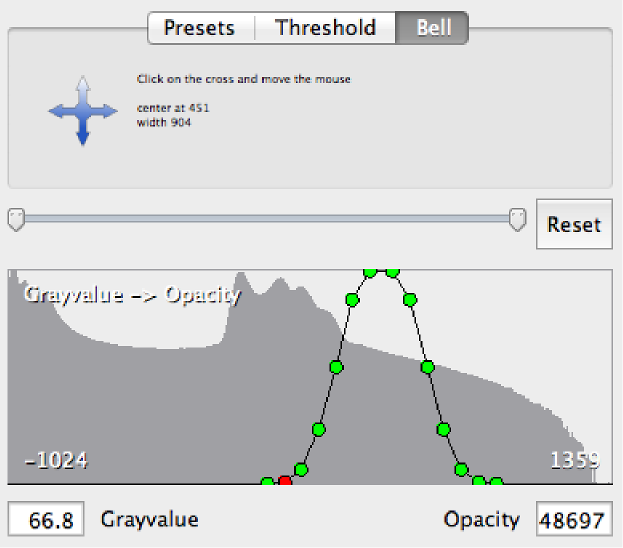
111 Grayvalue

Opacity 84631

A threshold shape begins with zero and raises to one across the "center" parameter. Lower widths results in steeper threshold functions.

14.4.2 Bell

Click on the "Bell" tab to active the threshold function generator.



Click on the cross and move the mouse

center at 451
width 904

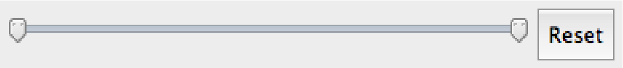
66.8 Grayvalue

Opacity 48697

A threshold shape begins with zero and raises to one at the "center" parameter and the lowers agains to zero. The "width" parameter correspondens to the width of the bell.

14.5 Customize transferfunctions in detail

14.5.1 Choosing grayvalue interval to edit



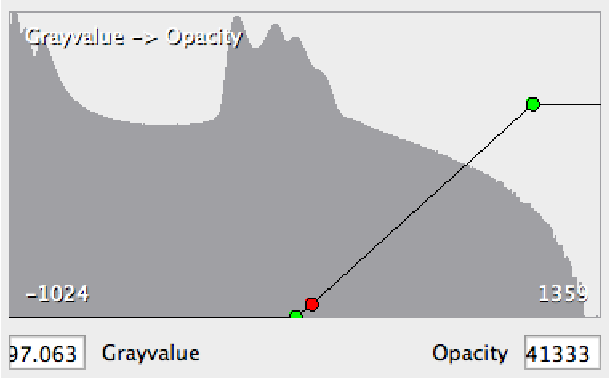
To navigate across the grayvalue range or to zoom in some ranges use the "range"-slider.

All three function editors have in common following:

- By left-clicking a new point is added.
- By right-clicking a point is deleted.
- By left-clicking and holding, an exisiting point can be dragged.
- By pressing arrow keys, the currently selected point is moved.
- By pressing the "DELETE" key, the currently selected point is deleted.
- Between points the transferfunctions are linear interpolated.

There are three transferfunctions to customize:

14.5.2 Grayvalue -> Opacity



97.063 Grayvalue

Opacity 41333

Figure 14.2: grayvalues will be mapped to opacity.

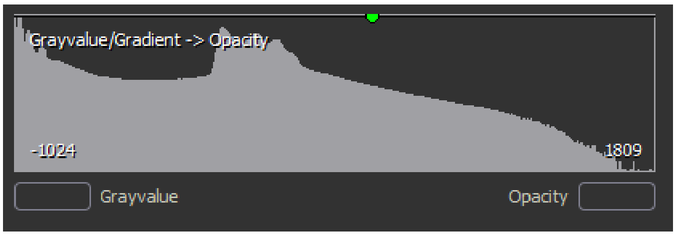
An opacity of 0 means total transparent, an opacity of 1 means total opaque.

14.5.3 Grayvalue -> Color

Figure 14.3: grayvalues will be mapped to color.

The color transferfunction editor also allows by double-clicking a point to change its color.

14.5.4 Grayvalue and Gradient -> Opacity



Here the influence of the gradient is controllable at specific grayvalues.