FSOPhantom User Guide

Introduction

FSOPhantom (Faster, Sharper, Open Phantom) is a Monte-Carlo radiation transport platform built around GEANT4 10.5. FSOPhantom uses STL (STereo-Lithograph) file format triangular meshes to define the boundaries of each component part of the geometry the end user wishes to use for radiation transport. This guide will cover producing STL meshes using the open-source software 3D Slicer, importing these meshes into FSOPhantom, and simulating radiation transport for different energies, source particles and source organs to calculate the dose per particle and absorbed fraction for each such category of simulated particle.

This manual assumes the end user is using FSOPhantom for internal dosimetry calculations of a living organism. However, because FSOPhantom is built upon GEANT4 10.5, the entire library of macro commands is available within FSOPhantom. Further, FSOPhantom itself is fully open-source, and the rest of the GEANT4 10.5 engine is available by editing the C++ source code that comprises FSOPhantom. The end user may use FSOPhantom as a stepping off point into using GEANT4 to simulate transport through space materials, high-level waste storage facilities, or other physical processes that deviate from FSOPhantom's original intent as a biota phantom creation and simulation tool.

Installation and setup

FSOPhantom is written for GEANT4 10.5. Before installing FSOPhantom, the end user will need a working installation of GEANT4 10.5 along with any GEANT4 packages they wish to use. For Windows users, pre-compiled libraries of GEANT4 10.5 and a matching pre-compiled installation of FSOPhantom are available from the source repository with 64-bit binaries, multi-threading enabled, and the Qt graphical user interface enabled.

A full step-by-step guide for installing and configuring GEANT4 can be located at https://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/InstallationGuide/html/gettingstarted.html. Note that for Windows users that opt for the pre-compiled libraries, one must still follow the GEANT4 "Postinstall Setup" to set the appropriate environmental variables for GEANT4 to function properly.

Once GEANT4 is installed and fully configured, simply extract FSOPhantom to a local directory. FSOPhantom requires no additional libraries beyond those included with FSOPhantom or those already included as part of GEANT4.

Segmentation: Getting started with 3D Slicer

Before creating and simulating a biota phantom in FSOPhantom, a set of STLs describing the geometry of the organism in question is

necessary. FSOPhantom was built with 3D Slicer in mind as the primary tool for creating the geometry, by segmenting medical images such as CTs or MRIs. However, note that if desired, any software that can produce the desired geometry as error-free STL files may be utilized. If the STLs in question are already available, skip ahead to the chapter on "Getting started with FSOPhantom".

3D Slicer Introduction

3D Slicer can be downloaded from https://www.slicer.org. Slicer is a fully open-source medical image processing and analysis suite, funded primarily by the National Institutes of Health and developed over the previous two decades. Note that there is an enormous amount of user-developed add-ons available for Slicer, as well as new features constantly being added to the core application. This guide will cover the basics to produce the desired segmentations for a radiation transport phantom, but the 3D Slicer wiki (https://www.slicer.org/wiki/Main_Page) is a powerful aid in learning to use the 3D Slicer features beyond the scope of this guide (developed using Slicer 4.9.0).

Loading and Viewing Images

One 3D Slicer is running, the first step is loading up the MRI, CT or other image modality stacks into the viewer. If the images are provided as DICOMs, simply click on DCM, choose Import and load up the top

directory. If the images are provided as individual image files i.e. IMG0001.png, IMG0002.png, etc., the images will need to be set up as a sequence (see Figure 19). Click on Load Data found on either the toolbar or on the Welcome screen. From here, choose to add an entire directory (if all the images from a single imaging sequence are located in their own directory) or more likely choose files to add, and use shift-left click to select all the images that belong to the sequence of interest. If your image sequence is made up of multiple separate image files, check the "Show Options" box and **unselect** "Single File" to ensure 3D Slicer treats the images as a sequence rather than as individual one-image scans.

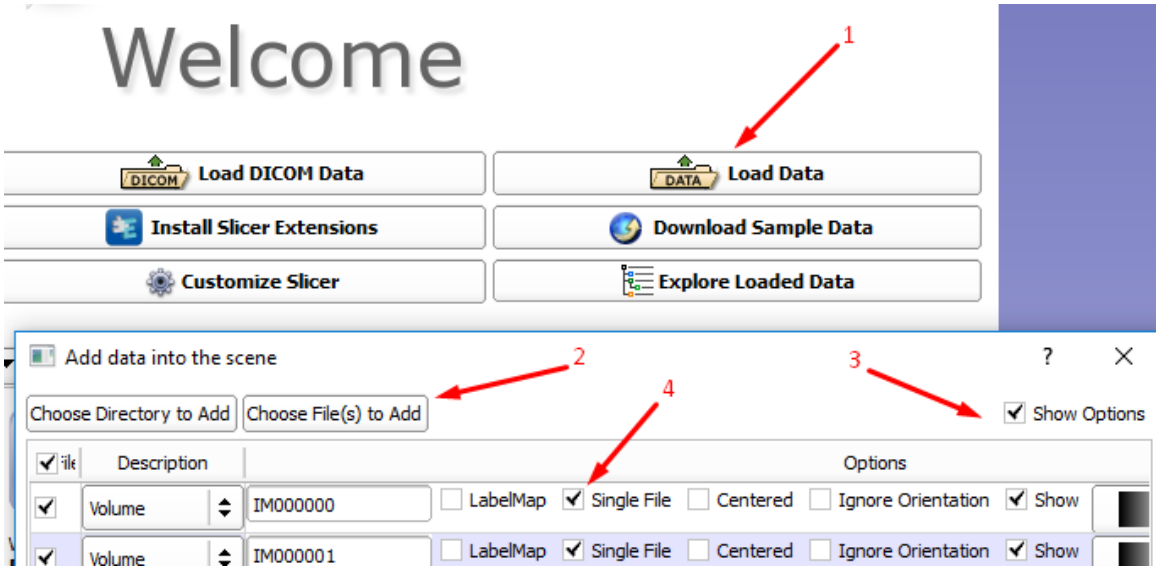


Figure 1: Loading a set of images as an image sequence (volume) in 3D Slicer

Now that the image sequence has been loaded, set up the scene view to better allow for viewing the contents of that sequence. Click the

Scene Layout  icon and select Conventional Four-Up. From this view, we can view the organism sliced along all three major axes; by default, the Red slice shows the Axial view, the yellow slice the Sagittal view, and the green slice the Coronal view. The 3D view in the upper right will be used later to show a live update of the segmentation as it is created, but another valuable use for this pane is to view all three slice-views displayed together in 3D.

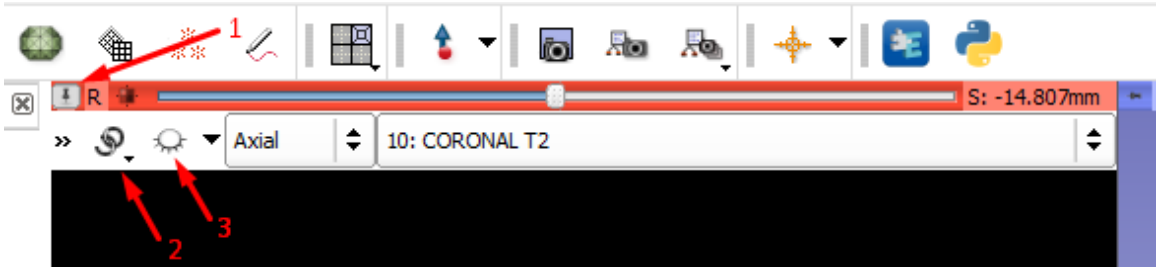


Figure 2: Linking and displaying slice window views in the 3D Viewer

Press the pin icon over the red slice view, which will cause the settings menu for that window to stay open (Figure 20). Then, click the Link/Unlink icon: this will tell 3D Slicer that any change we make to this slice view settings, we want to make to all the other slice views as well. Finally, click the Toggle Visibility icon, which should cause all three slice views to become visible in the 3D view. From this point, the 3D view can be rotated by clicking and dragging with the left mouse button inside the view. Zooming in and out can be done with the mouse wheel, while translation is accomplished by holding down the mouse 3 button

(typically this is the mouse wheel button). This 3D arrangement of slices can be critical to mentally orienting to the contents of a given slice.

Each slice view can be moved between individual slices using the mouse wheel, zoomed by dragging the right mouse button, and translated by dragging with mouse 3. In the slice views, the left mouse button controls the windowing settings, which sets how the intensity of the image is changed into brightness and contrast on the monitor. Holding down shift while moving the mouse over a given slice will instantly set the other slice views to set that same region visible; this feature is highly valuable when segmenting. Finally, at any time the slice view can be reset to show the entire slice by pressing the Match Extents icon (Figure 21). Once multiple image sequences have been loaded (just repeat the steps at the start of this section), the slice viewers can change between sequences using the drop-down boxes located in each slice view.

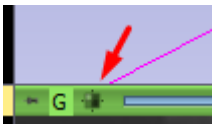


Figure 3: Resetting the view to the maximum extents

Figure 4: Creating a new linear transformFigure 5: Resetting the view to the maximum extents

Image Registration

Usually, when multiple image sequences have been taken on the same machine with the same organism positioning, all the image sequences will be natively aligned with each other. However, in some cases it may be desirable to use image sequences from different captures

for segmentation. For example, the FSOPhantom flatfish was created by segmenting both against CT images and against MRI images. If these images sequences are loaded individually, chances are that the two sets of images will not be aligned with each other. The process of rotating and translating these to align with each other is called registration.

First, the two image sequences will need to be manually adjusted to be reasonably close to being in alignment with each other. In the Modules section, select Transforms, and create a new Linear Transform (see Figure 22). In the scene view, set one of the slice viewers to show the first volume, and another to display the second volume. Change through the Axial/Sagittal/Coronal section until a roughly matching

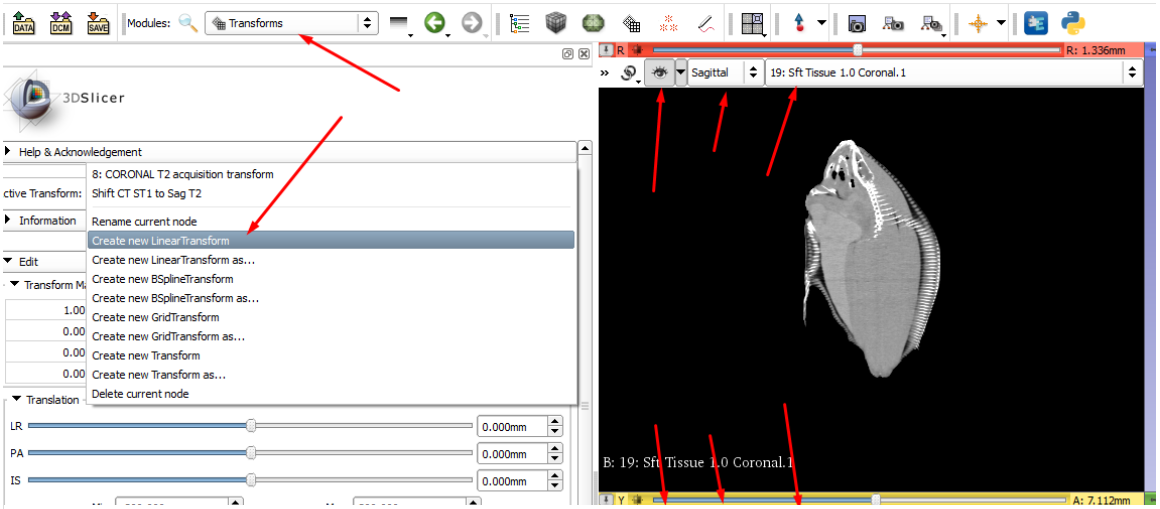


Figure 6: Creating a new linear transform

Figure 7: Using the level tracing tool to fill a regionFigure 8: Creating a new linear transform

slice is found for each. Then, look in the 3D View to see how the two image sequences are displaced and rotated from each other.

In the Transforms window, go to the Apply Transform section and specify which of the volumes will be one to be transformed. Then, proceed to Edit the transform to get the rotation and translation to a good approximation. For example, in the 3D View displayed in Figure 23 for transforming the “SAGITTAL T2” to match the “Sft Tissue 1.0 Coronal 1” volume, there first needs to be a 90 degree rotation around the IS (Inferior-Superior) axis, and a large negative translation along that same axis. Don’t be afraid to experiment with translation and rotation to find the right adjustment, these are very easy to reset to zero if anything goes awry.

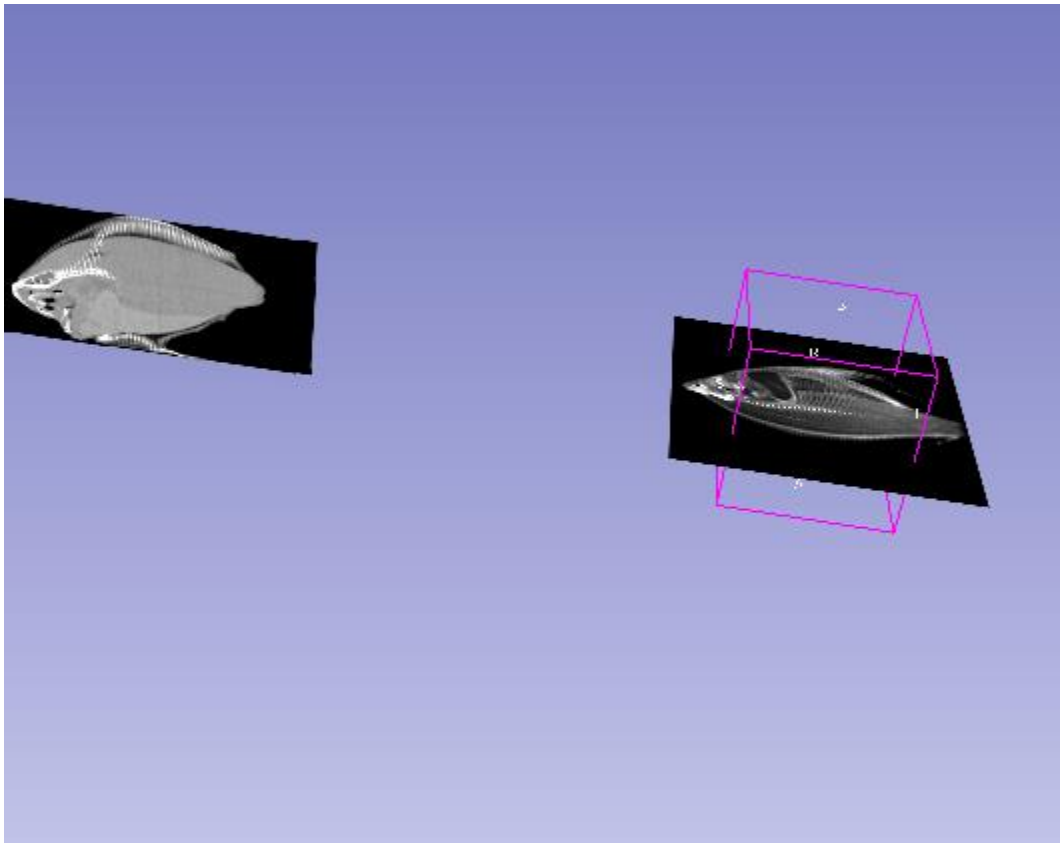


Figure 9: Checking the initial (mis)alignment of two different sets of images by comparing a similar slice

Once the two image sequences are nearly aligned, it will be difficult to narrow the alignment down any further using the 3D view. At this point, change the scene view to display just a single slice e.g. “Red Slice Only.” Now, view the expanded options for the slice view by clicking the ‘>>’ icon (Figure 24). Set the secondary volume to be visible by clicking the dropdown box for the foreground. Finally, adjust the foreground slider so that the slice for both volumes is visible at once.

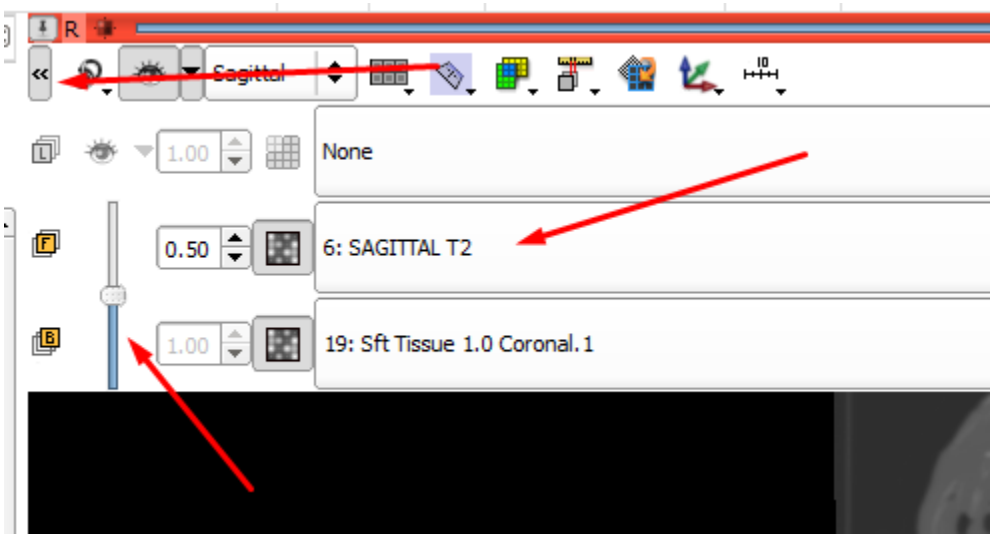


Figure 10: Setting a second image sequence as a foreground sequence, to view overlapping slices in the same slice viewer

From this point, the transform can be further fine-tuned. For some image sets, this may be sufficient to get a suitable alignment, such as when imaging fixed and mounted invertebrates where posture does not change, and image resolution may be low. However, for samples that still have flexible tissues, the organism's posture may not match exactly. If that is the case, the end user is advised to look at the BRAINS or Elastix General Registration modules. For fairly rigid samples, BRAINS is the ideal transformation, as it will keep the geometry relatively intact. For samples that have experienced a change of posture or that are otherwise hard to align well using BRAINS, Elastix is ideal.

Basic Segmentation

Segmentation is the process of marking out which regions in each part of the image sequence belong to which organs or tissues of interest.

For FSOPhantom to produce realistic results, the entire organism should be segmented, although regions of low interest to the end-user may be grouped together simply by similar density & composition (see e.g. the "Muscle/Soft Tissue" segmentation in the Flatfish and Trout, which serve as catch-all segments). For ease of use in 3D Slicer, the end user is also advised to have a segment dedicated to the air/void surrounding the organism, as it is of great utility to exclude certain tools from attempting to segment outside the body of the organism in question. To get started, in the Modules menu, select Segment Editor. From here, choose the segmentation to edit or create a new segmentation, then set the "Master volume" to the image sequence to use for segmentation.

The basic tools for editing a segmentation are straightforward. The Paint tool 

will mark a region as belonging to that segment as the user clicks and drags with their mouse over the image. By default, this works as a circular brush, only as thick as the current slice. This will be used primarily to touch-up regions that have been mostly segmented using the other tools but can also be used when other tools fail to segment a region correctly. For some cases, such as when filling in a structure free hand that did not have sufficient contrast in the original images, the sphere brush can be more useful, which paints with a brush that is spherical. The sphere brush is also quite useful when painting in

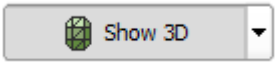
the 3D View rather than in an individual slice view, and it is advised to keep the 3D View visible whenever using the sphere brush for an extended period of time.

The Draw tool 

is primarily valuable when using a screen with a stylus. This tool allows the user to mark out the boundaries of a region by either clicking to add points to the boundary or clicking and dragging to draw a continuously variable boundary. Continue to click to add new points to the boundary. When nearly complete, right click once then left click on the final point on the boundary to close the region and mark it in.

The Erase tool 

functions almost identically to the Paint tool, with the exception that it erases existing segments.

Once some segments have been created, clicking the Show 3D button 

will allow the user to visualize the entire segmentation in the 3D View window.

Masking

For all tools, masking is a powerful utility that can make otherwise extremely cumbersome tasks manageable and straightforward.

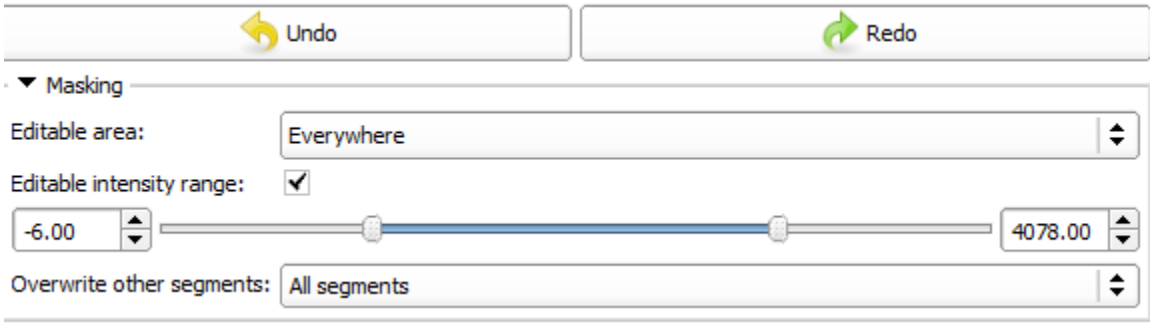


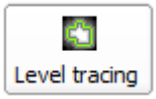
Figure 11: Using the masking tool

Masking allows the end user to control which parts of the image can be actively segmented and which will be ignored by the current tool. Masking by editable area allows the user to use existing segments to control which regions are still editable, and this can be done by excluding all segments or just visible segments, inside of existing segments or outside of said segments. Earlier it was mentioned that having a segment for the air/void surrounding the body of the organism can be a valuable tool: masking is the key reason for this. Using the void segment to exclude regions outside of the organism can keep tools like Level Tracing or Margin from erroneously following small variations in the background of the air near the organism.

Masking can also be performed by setting an editable intensity range, and this can also be combined with the editable area tool to mask

by both intensity and existing segments. For example, the eye choroids and the skeleton both have similar high intensities in the flatfish CTs used as examples throughout this document. Merely using the Thresholding tool won't work to segment the eye choroids, as it will also capture the skeleton. Rather than having to draw the eye choroids by hand, following the exact contours in the images, masking makes the process easy. Simply set the editable intensity range to include the eye choroids, at which point a single click of the Paint -> Sphere brush per eye will segment that eye. Finally, setting the editable area masking to exclude the now-created eye choroid segment means that the skeleton can now be completely segmented in a single use of the Thresholding tool.

Level Tracing

Level tracing 

is a version of an automated Draw tool. However, instead of manually drawing the boundary, 3D Slicer will attempt to find a loop that follows the same intensity as whichever voxel is currently highlighted in the slicer viewer. By pressing left-click, the previewed boundary will be segmented (Figure 26).

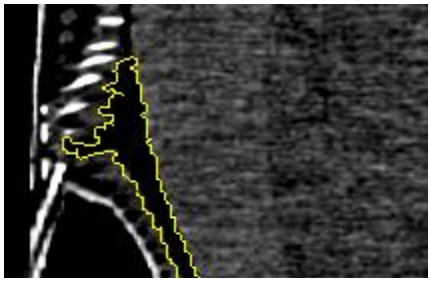


Figure 12: Using the level tracing tool to fill a region

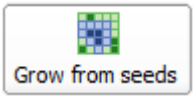
Figure 13: Banding artifacts that may be present in an image sequence that are not easy to spot with anti-aliasing enabledFigure 14: Using the level tracing tool to fill a region

Note that in some images, there may be artifacts present that prevent the level tracing tool from segmenting what appears to be an obvious boundary. To determine if this is the cause, turn off background interpolation and check for obvious banding artifacts in the raw images (Figure 27).

Figure 15: Banding artifacts that may be present in an image sequence that are not easy to spot with anti-aliasing enabled

Figure 16: Grow from seeds tool: beforeFigure 17: Banding artifacts that may be present in an image sequence that are not easy to spot with anti-aliasing enabled

Grow From Seeds

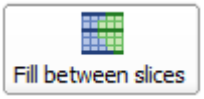
The Grow from Seeds tool 

, when combined with high quality, high contrast image sequences, can largely segment an organism automatically. To use this tool, first a small 'seed' segment should be painted for each organ or tissue that is to be segmented. **It is critical that the initial seeds do not overlap with other organs or tissues**. Then, proceed to the Grow from Seeds tool and choose Initialize. At this

point, the tool will preview how it will auto-segment the rest of the image sequence. Further edits to the initial seeds may be performed using the paint or erase tools, at which point the seeds tool will automatically update (select Grow from Seeds -> Update if it does not update automatically). See Figures 28 and 29 for a before and after of the Grow from Seeds tool when applied to CT images of a pine tree in order to auto-segment the rings. Do note that the tool attempts to grow each seed outwards, seeking out areas of high contrast as boundaries when seeds begin to interfere with each other. There is an upper limit to how far the seeds tool will grow, which may be artificially limited by setting how much of the slice is visible when the tool is first initialized. To

change the maximum region once limited in this fashion, Cancel and then re-Initialize the tool.

Fill Between Slices

The Fill Between Slices tool 

is useful when there are far too many slices to be manually segmented, and the image contrast is too low to segment the burdensome regions using Grow from Seeds. Fill Between Slices takes as input several fully segmented slices and interpolates between the input segmentations to fill in the missing regions.

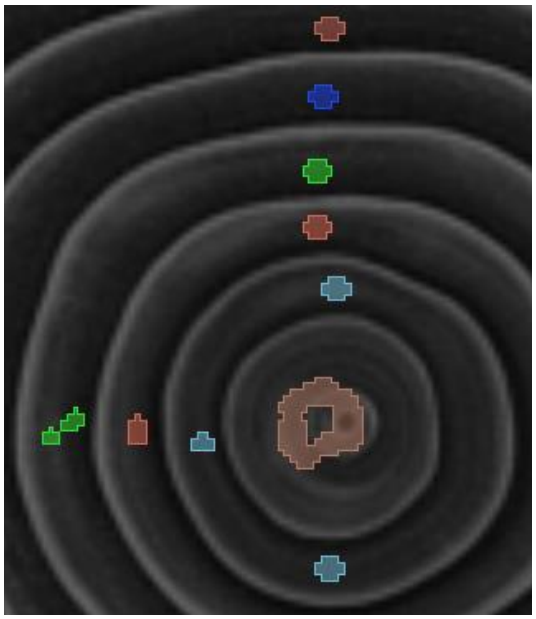


Figure 18: Grow from seeds tool: before

Figure 21: Grow from seeds tool: after

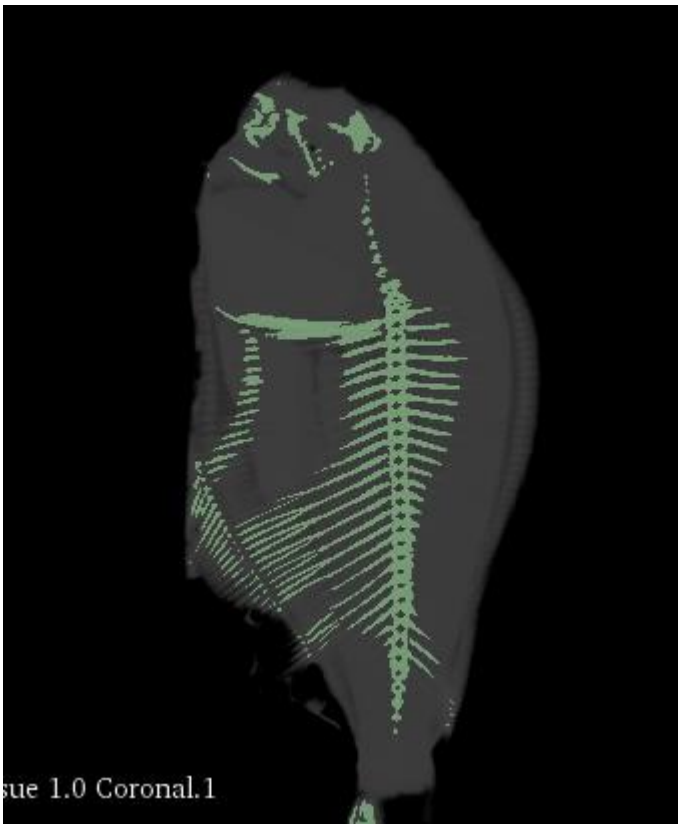
Figure 19: Grow from seeds tool: afterFigure 20: Grow from seeds tool: before

Figure 22: Thresholding toolFigure 23: Grow from seeds tool: after

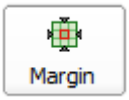
Threshold

The Thresholding tool either marks a region or creates a mask based on the minimum and maximum intensity. When segmenting an organism, this can make the process of segmenting the skeleton in a CT a process that takes a matter of minutes instead of a matter of days or weeks. For MRIs with sufficient contrast, this can be used to segment out white and grey matter in the brain, although in practice the thresholding tool will more often be used in conjunction with masking. When preparing a mask by intensity, the Thresholding tool allows for a live preview of what regions that mask will apply to: simply select "Use for masking" to create an intensity mask using the current settings. Masking and the Thresholding tool also combine well by masking by area, for more details on how this may work see the section on Masking.

Figure 24: Thresholding tool

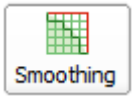


Margin

The Margin tool 

can be used to grow or shrink the boundaries of a segment. One use case for this tool is the creation of a lining segment, such as adding a “skin” segment where there exists no clear boundary in the image contrast. A similar use case comes in the creation of an intestinal lining segment, in an instance where the intestines and their contents are not discernable from each other (or where the intestines needed to be drawn freehand using the sphere brush). For full details on this use case, see the section on Logical Operators.

Smoothing

The Smoothing tool 

provides the means to smooth out unnaturally jagged regions that can be produced by some of the automated tools. The kernel size sets how large of a region is examined at a time during the smoothing process.

Scissors

The Scissors tool 

can be used to delete or fill large areas from a segmentation with a minimal amount of input. The shape can be

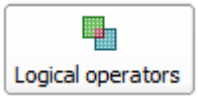
drawn free form on a slice, which will then be projected along that axis, erasing or filling that region (or the area outside of that region, depending on the operation selected). This tool can also be used in the 3D View to draw the cutting boundaries rather than projecting a 2D slice. This provides an alternative to the Paint -> Sphere Brush for quickly filling in large volumes that cannot be automatically segmented.

Islands

The Islands tool 

allows the user to act upon a segment in terms of separate connected components. Islands is useful for cleaning up the results of automatic segmentation. It can be used to split up separate regions into their own unique segments. Often, automatic segmentations will produce small regions around minor artifacts present in image disconnected from the rest of the organism, Islands is excellent for quickly and automatically deleting such small isolated regions.

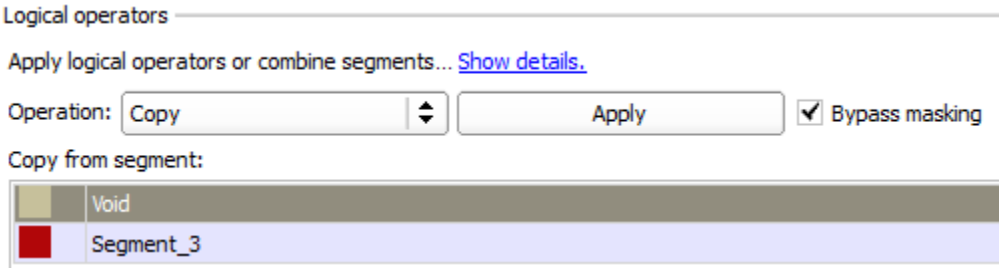
Logical Operators

The Logical Operators tool 

allows one to copy, add, subtract, or intersect two segments, or to fill, clear or invert an existing segment. The logical operators are used mostly in conjunction with other tools. For example, using an existing “Void” segment, a “Skin” segment

will be created in the example flatfish, even though there is no clear delineation of the skin in the original images. To start, a copy of the void is first copied to the new segment (see Figure 31).

Figure 27: Logical operators tool: copying one segment from another



Now, using the Margin tool, turn off "Overwrite other segments" in Masking, then grow the new segment by the desired thickness of skin (in this example, 3mm was used). Finally, return to the Logical Operators section, and choose to subtract

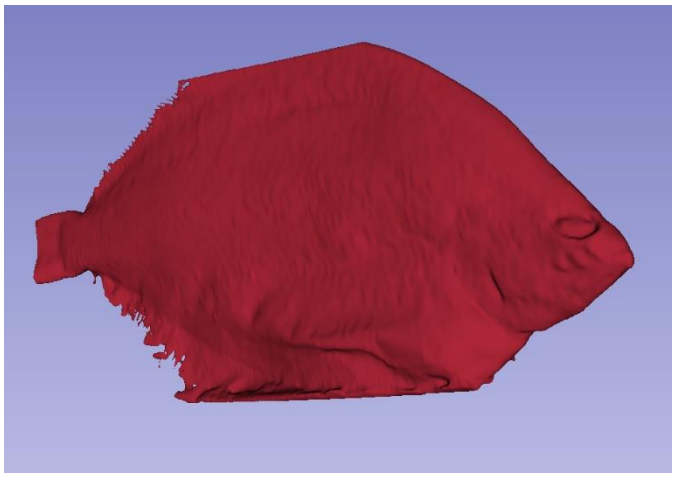


Figure 28: 3D view of a new 'skin' segment created using the logical operators, masking and margin tools

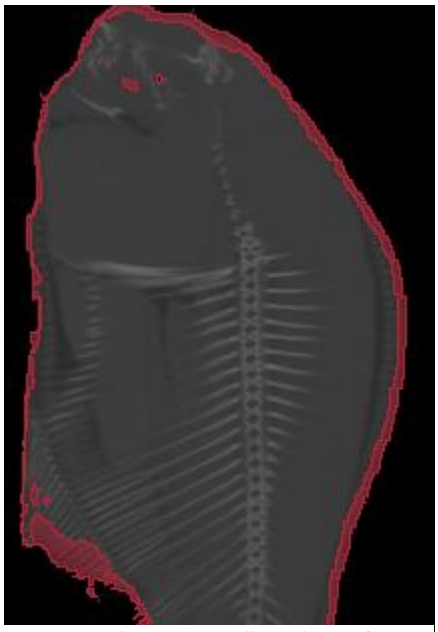


Figure 31: Slice view of a new 'skin' segment created using the logical operators, masking and margin tools

Figure 32: Exporting segmentations as modelsFigure 33: Slice view of a new 'skin' segment created using the logical operators, masking and margin tools

the original “Void” from the new segment, which will leave behind only the new region that was added by the Margin tool (Figure 33). The resulting Skin segment can be further refined by using the Islands tool to identify and keep only the largest island, or to split islands to segments in the event the “Void” segment was smaller than the full volume and a floating ‘box’ of skin was added to the exterior.

Exporting Segmentations as STL Models

To export the segmentations as STL models, the first step is to convert them. Initially, they are a segmentation, which is a map of voxels in space based on the voxel dimensions in the image sequence. They need to be converted to a model, which will be a collection of triangular faces that define the boundaries of those volumes which are no longer constrained to the voxel

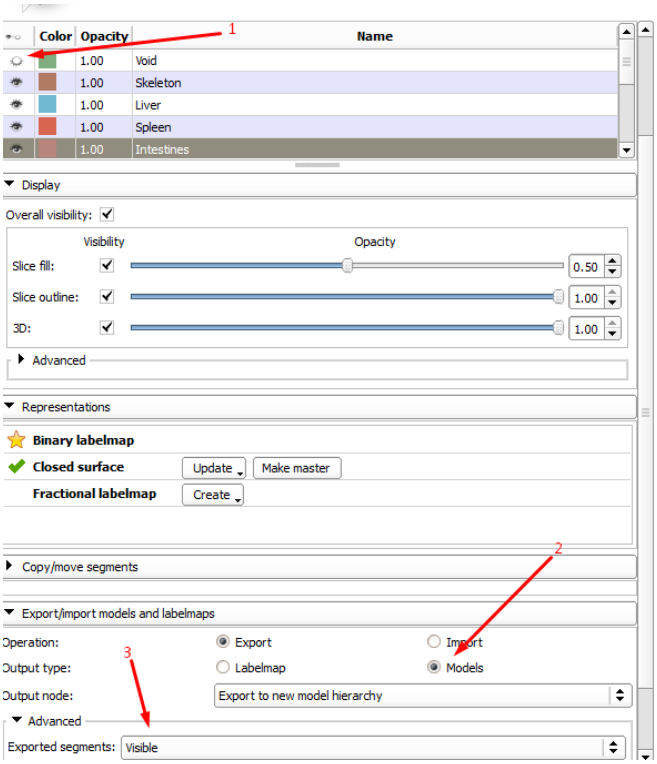


Figure 34: Exporting segmentations as models

Figure 35: Exporting models as STL model filesFigure 36: Exporting segmentations as models

boundaries. This allows the models to realistically reproduce curving organic structures from the rectangular nature of MRI and CT voxels.

First, go to the Segmentations module (Figure 34). If there are any segments that should not be included in the model (such as a Void/Air surrounding segment), set that segment's visibility to off. Then, go to Export/Import Models and Labelmaps, and choose to Export with output type of Models. If excluding any unwanted segments, open the Advanced section, and change Exported segments from All to Visible. Finally, press Export to export the models.

Now all that is left is to save the models. Press the Save icon 

, and find the models in the Save Scene window (Figure 35). Change the File Format to "STL (.stl)", and if necessary, change the directory to wherever the user would like the STL files to be saved.

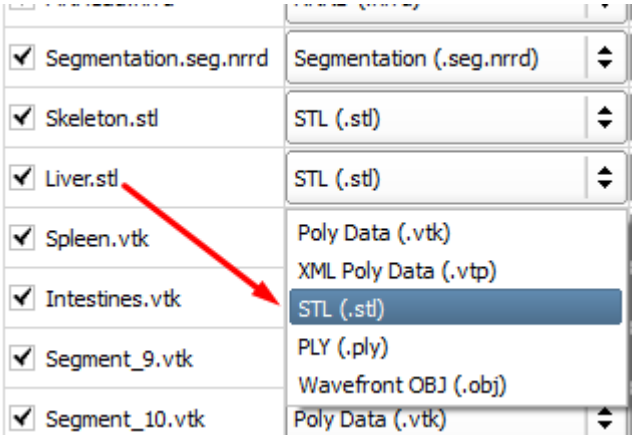


Figure 37: Exporting models as STL model files

Using FSOPhantom

Figure 38: Running FSOPhantom in interactive mode with the Qt UI activeFigure 39: Exporting models as STL model files

Getting started with FSOPhantom is fairly straightforward: for an interactive, user-interface

driven session, simply double-click on the FSOPhantom icon or run "FSOPhantom" from the command line. FSOPhantom will proceed to ask several questions regarding the geometry of the problem. These questions, in detail, are:

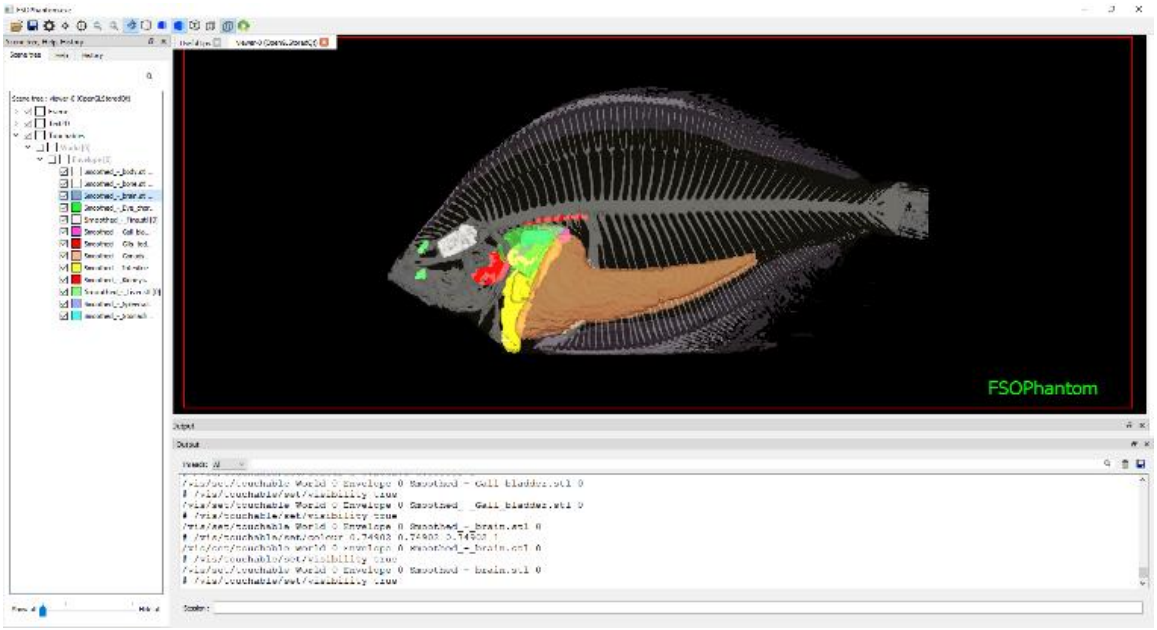


Figure 40: Running FSOPhantom in interactive mode with the Qt UI active

Figure 41: MDEF (Material Definition) file format exampleFigure 42: Running FSOPhantom in interactive mode with the Qt UI active

Name for run-results output file:

Besides giving the results of any simulation run at the console, FSOPhantom saves the material specifications and the run results in whatever filename is specified here. By default, this will be in the same directory as FSOPhantom was started in, use a full file path to save the file elsewhere.

How many STLs will be loaded?

This tells FSOPhantom how many STL files in total comprise your model. This tells it how many volumes and materials it will be using.

Transform needed? (3 queries)

Ideally, any STL files should be located close to the x,y,z, coordinate 0,0,0. This is due to an internal requirement for GEANT4, that the World volume must always include 0,0,0. The three queries in this subsection allow the user to offset their model in the x,y,z, directions to correct for this if that is the case. If the model is not located close to 0,0,0, the results will still be valid. However, the geometry loading time and the model run time may be longer than desired, as the World volume (located outside of the surrounding envelope) will be as large as it needs to be to encase both the model and 0,0,0.

Select Mesh (STL) Definition File

Select the first of the STL files from the model to load here. The loading order does not matter, it influences only the order the volumes will be listed in the user interface.

Select Material Definition File

Select the material definition file (*.mdef by default) that describes the material composition for the model just loaded. A material definition file is a plain text (ASCII) file with the following format (see example Figure 37):

Line 1 gives the material name as a single word (no spaces), space, the material density in g/cm3, space, and the number of elements that comprise the material.

Lines 2 and beyond give the symbol for an element, space, then the weight fraction for that element.

```
1  ICRP_Large_Intestine 1.045 9
2  H 0.106
3  C 0.115
4  N 0.022
5  O 0.751
6  Na 0.001
7  P 0.001
8  S 0.001
9  Cl 0.002
10 K 0.001
```

Figure 43: MDEF (Material Definition) file format example

Figure 44: A simple 2-STL example geometry.cfg fileFigure 45: MDEF (Material Definition) file format example

X/Y/Z Multiplier:

FSOPhantom automatically encases the model in an envelope of surrounding material. The envelope is initially calculated as the parallelepiped with x, y and z extents that capture the most extreme x, y and z components of the entire model. The multipliers given here tell FSOPhantom how much larger to make the surrounding envelope relative to that minimal bounding box. Other output on screen will give the end user the dimensions of that minimal bounding box, should the

user desire a specific size unrelated to the bounds of the model. When in doubt, it is advised to make this a large volume, ideally many times larger than the largest mean free path of the particles & energies that will be run using this geometry. However, when simulating a dose to the organism from the surrounding volume from very short-ranged particles, a much smaller bounding box may be more desirable.

Select Material Definition File for surrounding environment

Much like selecting a material definition file for the model itself, this query asks for an identically formatted material definition file to define the material used to create the surrounding envelope. Common surrounding envelopes might be soil, air or water.

Further terminal output

Although there are no further queries to respond to at the terminal, the user is advised to monitor the upcoming output after specifying the last material file in the previous step. FSOPhantom will calculate and output the exact set of macros necessary in order to confine a source to each of the input organs. Since FSOPhantom was originally created for internal dosimetry (Source -> Target) transport, this step is crucial to note (and copy to re-use in the future) if the user plans to perform such internal dosimetry.

Geometry.cfg

FSOPhantom optionally[1] supports the use of a file named 'geometry.cfg' located in the same directory as FSOPhantom. Geometry.cfg serves as answers to all the geometry queries that FSOPhantom would normally ask of the user while initially loading. This feature is especially useful when running FSOPhantom in batch mode (where the user interface is not loaded) or when loading especially complex geometry (where loading times may be inconvenient).

Figure 38 shows an example geometry.cfg file. The first line specifies the filename that should be used for any results. Line two answers how many STLs will be loaded. Line three answers the x, y and z offset questions in a single line with spaces between answers. Line 4,5 then 6,7 give the pairs of paths to STL files and paths to the matching MDEF file. (Note that however many STL

```
1  Flatfish - Caffrey Compositions.txt
2  2
3  0 0 0
4  C:\Users\Delvan\Smoothed - bone.stl
5  C:\Users\Delvan\ICRU-44 Cortical Bone.mdef
6  C:\Users\Delvan\Smoothed - muscle.stl
7  C:\Users\Delvan\ICRP Muscle.mdef
8  10 10 10
9  C:\Users\Delvan\water.mdef
```

Figure 46: A simple 2-STL example geometry.cfg file

Figure 47: Whole body source uniform concentration absorbed gamma fraction for the FSOPhantom flatfish and an 845g Ulanovsky & Pröhl ellipsoidFigure 48: A simple 2-STL example geometry.cfg file

[1] Note that at the time of release, FSOPhantom on Linux requires the use of the Geometry.cfg file. This will be addressed in a later patch.

files were specified in line 2, there should be that many pairs of lines to locate all of the STLs in question). The last two lines give the surrounding envelope size multiplier, with the x, y and z offsets separated by a space, and finally the path to the MDEF for the surrounding envelope. If FSOPhantom locates 'geometry.cfg' located in the same directory, it will attempt to load straight into either interactive mode or batch mode.

Example macro file - # marks comments

```
#This tells FSOPhantom how many threads to utilize for multi-threading.

/run/numberOfThreads 4

#This initializes the run

/run/initialize

#Optional, these two control how much feedback the user gets

#0, 1 or 2

/control/verbose 1

/run/verbose 1

#This section creates a parallelepiped isotropic volume source

#that is confined to just the Trout_-_Intestines.stl segment.

/gps/pos/type Volume
```

```
/gps/pos/shape Para

/gps/pos/halfx 2.20428

/gps/pos/halfy 1.28647

/gps/pos/halfz 8.50015

/gps/pos/centre 2.22376 0.188734 3.45016

/gps/pos/confine Trout_-_Intestines.stl

/gps/ang/type iso

#Set the particle to be mono-energetic, and gammas

/gps/ene/type Mono

/gps/particle gamma

#Set the particle energy to 1,000 keV, then run 50,000 particles

/gps/energy 1000 keV

/run/beamOn 50000

#Change the particle type to electron

/gps/particle e-

#Change the energy to 5 keV and run 50,000 particles

/gps/energy 5 keV
```

```
/run/beamOn 50000

#Change the energy to 10 keV and run 50,000 particles

/gps/energy 10 keV

/run/beamOn 50000
```

Batch mode

Rather than running FSOPhantom inside an interactive session, the user can run FSOPhantom in batch mode. This mode does not load any user interface: it executes a single specified macro file, then exits. To run FSOPhantom in batch mode, open a command line, navigate to the folder that contains FSOPhantom, and run: FSOPhantom “Name of macro file here”