NeuroTessMesh Documentation

Version 0.2.4

Juan José García Cantero & Félix de las Pozas Álvarez

March 2022

Visualization & Graphics Lab, Universidad Rey Juan Carlos

Visualization
& Graphics Lab

CONTENTS:

CHAPTER

ONE

NEUROTESSMESH INTRODUCTION

NeuroLOTs is a set of libraries and tools that implement a method for generating neuronal meshes and for visualizing them at different levels of detail using GPU-based tessellation. As a part of NeuroLOTs, **NeuroTessMesh** provides a visual environment for the generation of 3D polygonal meshes that approximate the membrane of neuronal cells, from the morphological tracings that describe the morphology of the neurons. The 3D models can be tessellated at different levels of detail, providing either homogeneous or adaptive resolution along the model. The soma shape is recovered from the incomplete information of the tracings, applying a physical deformation model that can be interactively adjusted. The adaptive refinement process performed in the GPU generates meshes that allow good visual quality geometries at an affordable computational cost, both in terms of memory and rendering time. NeuroTessMesh is the front-end GUI to NeuroLOTs framework.

This documentation is for NeuroTessMesh version 0.2.4 software that can be obtained from the homepage.

1.1 Hardware requirements

The application requires a graphic card that supports OpenGL 4.0 at least. NVIDIA offers support for GPU tessellation from the GTX 400 serie on, and ATI/AMD from Radeon HD 5000 serie on. Previous series of both companies are not able to tessellate and consequently the mesh refinement will not work.

Note: The GPU Tessellation on Intel graphics cards is present only on certain models, for more information you can visit the next link: Supported APIs and Features for Intel Graphics Drivers

1.2 External Links

The homepage for NeuroLOTs and NeuroTessMesh is located at NeuroLOTs & NeuroTessMesh Homepage and the source code for the latest releases is available in the NeuroLOTs Github page and NeuroTessMesh Github page. For reporting bugs please use the NeuroLOTs Github Issues and the NeuroTessMesh Github Issues pages. If you have any questions or suggestions about NeuroLOTs or NeuroTessMesh refer to dev@vg-lab.es.

1.3 Installation and running

NeuroTessMesh can be downloaded from the NeuroLOTs & NeuroTessMesh Homepage for Linux and Mac operating systems and executed locally. Additionally it can be executed using a docker image.

1.3.1 Docker containers

The docker containers for **NeuroTessMesh** can be found on Docker Hub. It's recommended to use the highest tag number as it represents the latest official release.

1.3.2 Executing locally

The application options and parameters are:

OPTION	PARAMETER	DESCRIPTION
`--version`	*none*	Shows the version of the application.
`--help`	*none*	Shows the options and arguments used for executing the application.
`-bc`	*path_to_bc_file*	Load BlueConfig file.
`-swc`	*path_to_swc_file*	Load SWC file.
`-xml`	*path_to_xml_file*	Load XML scene file.
`-target`	*target_label*	Specifies target label of the BlueConfig file.
`-zeroeq`	*schema_id*	Enables ZeroEQ communications with the specified id.
`--json`	*path_to_json_file*	Load JSON data file.
`-ws` `--window-size`	*width height*	Specifies the size of the application window.
`-fs` `--fullscreen`	*none*	Sets the application window to fullscreen mode.
`-mw` `--maximize-window`	*none*	Maximizes the application window to the desktop resolution.
`-s` `--samples`	*samples_number*	Sets the samples number for OpenGL. Overwritten, if present, by the enviroment variable CONTEXT_OPENGL_SAMPLES.
`-nvs` `--no-vsync`	*none*	Disables vertical sync for OpenGL. Overwritten, if present, by the enviroment variable CONTEXT_OPENGL_VSYNC.
`-cv` `--context-version`	*major minor*	Sets the OpenGL version of the application. Overwritten, if present, by the enviroment variables CONTEXT_OPENGL_MAJOR and CONTEXT_OPENGL_MINOR.

If the options are incompatible or its parameters invalid the application will abort the execution and will show the help message in the console.

1.3.3 Test dataset

A test data for NeuroTessMesh can be downloaded from:

- http://neuromorpho.org/dableFiles/allen%20cell%20types/CNG%20version/H16-03-001-01-09-01_559391771_m.CNG.swc

1.3.4 Docker example


```
xhost +local:docker
# Pull the image.
docker pull vglab/neurotessmesh:0.2.4-nvidia-ubuntu-16.04
# Download example data
wget http://neuromorpho.org/dableFiles/allen%20cell%20types/CNG%20version/H16-03-001-01-09-01_559391771_m.CNG.swc
# Run example
docker run --gpus 1 -ti --rm -e DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix -v /etc/machine-id:/etc/machine-id -v $(pwd)/H16-03-001-01-09-01_559391771_m.CNG.swc:/H16-03-001-01-09-01_559391771_m.CNG.swc --privileged vglab/neurotessmesh:0.2.4-nvidia-ubuntu-16.04 /usr/bin/NeuroTessMesh -swc /H16-03-001-01-09-01_559391771_m.CNG.swc
```


CHAPTER

TWO

NEUROTESSMESH USER INTERFACE

2.1 NeuroTessMesh Application Toolbar

The application bar presents several icons to perform actions such as opening datasets, and showing or hiding configuration panels. The user can see the menu entries to associate each icon with function (Fig. 2.1).

Fig. 2.1: NeuroTessMesh application tool bar.

2.1.1 File

The first three icons loads an scene or neuron morphology. The neuron morphology must be described following the SWC format Scenes composed of severan neurons must be described using an XML specific format (described in the **file formats** section).

- **Open BlueConfig**: Opens neuron morphology in BlueConfig format.
- **Open XML scene**: Opens a scene composed of several neurons.
- **Open SWC**: Opens a neuron morphology in SWC format.

2.1.2 Configuration

- **Tessellation params**:
  - **Subdivision level**: Maximum level of subdivisions for the visualization. The valid values are in the range of 1 and 30.
  - **Distance threshold**: Further distance to witch the subdivision is applied. The valid values are 0 to 1, 1 being the camera maximum visibility distance.
  - **Tangent smoothing**: Scales the modulus of the orientation vectors applied to smooth the neurite trajectories. The valid values are 0 to 1, 0 being no smoothing and 1 the maximum smoothing.
- **Tessellation criteria**: Select the tessellation criteria applied in the subdivision.
  - **Homogeneous**: Same level of subdivision for all the mesh.
  - **Camera distance**: The level of subdivisions is higher near the camera.

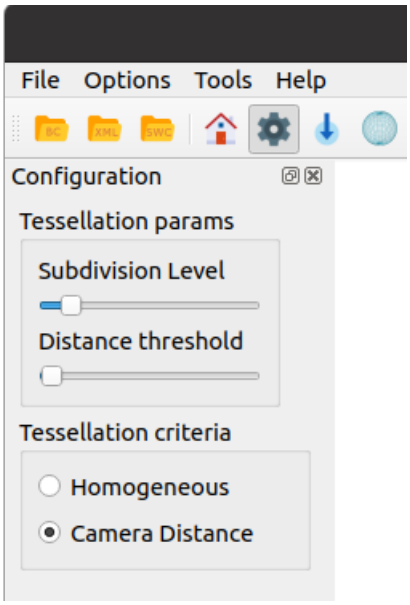

Fig. 2.2: Configuration panel.

2.1.3 Render panel

- **Color**: Select the color for each of the following entities.
  - **Background**
  - **Neuron**
  - **Selected neuron**
- **Render piece selection**: select the parts (full, only soma, only neurites or nothing) of the neuron to visualize. This selection can be done independently for each of the following types of neurons.
  - **Neuron**
  - **Selected neuron**

2.1.4 Edit and save panel

- **Select Neuron**: select the target neuron from a list of the neurons in the current scene.
- **Parameters**: different parameters to modify the reconstructed neuronal mesh.
  - **Radius factor**: scales the radius of the initial sphere used to generate the soma. [0-1].
  - **Neurite [n] factor**: factor that multiplies the distance from the neurite n to the soma. 0: Distance=0 (on the soma surface). 1: Current distance*2
- **Save button**: save the actual mesh reconstruction to “obj” file.

2.1.5 Scene camera controls

The camera can be manipulated using the mouse. The movements are:

- Rotation: mouse left button + mouse movement.

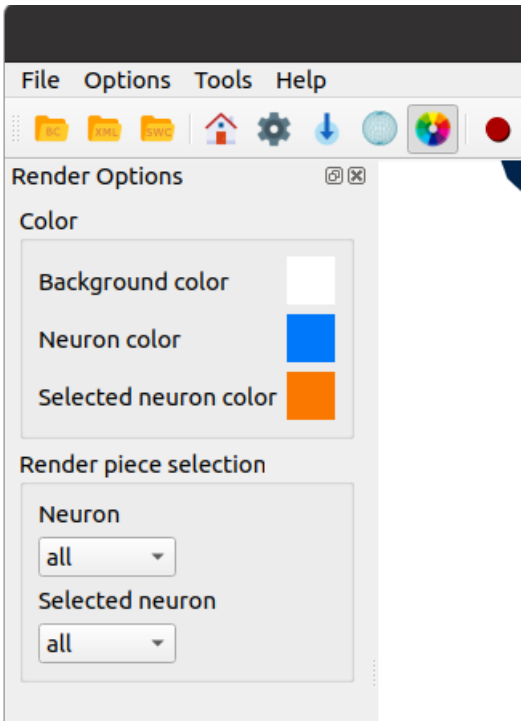

Fig. 2.3: Render panel.

- Translation: mouse scroll button + mouse movement.
- Zoom: mouse scroll to zoom in and out.

2.2 Recorder

The recording feature can be activated using the **Tools** menu or by clicking the Recorder icon in the toolbar. The user will be presented with the recorder configuration dialog (Fig. 2.5).

The recorder will generate a mp4 video if the media application **ffmpeg** is detected and available, if not the generated output will be individual frames. The frames per second of the output can be specified here. The user can specify the input of the recording, being the main application window or just the 3D viewport.

In the advanced configuration dialog (Fig. 2.6) the user can modify additional options as which worker (video or frames) to use to generate the output. The advanced configuration dialog can be enabled by checking the **Advanced recorder options checkbox** in the **Tools** menu).

Using the advanced configuration an area of the application can be selected for recording as input.

If the output is a video the user can specify the location of the generated file using the Select button. If the output is a sequence of frames the user can specify the destination directory using the Select button in the dialog.

The dimensions of the output are shown and can be modified with the scale options.

While the recorder is working the associated toolbar button will remain down and the icon will change every second with an image of a "Stop" button with REC letters written inside it. The user must click it again to stop the recording. The recorder can also be stopped using the **Recorder** button in the **Tools** menu or the keyboard shortcut **Ctrl + R**.

Note: If the output is a video the filename of the output file will end with "_temporal" while recording. Once the

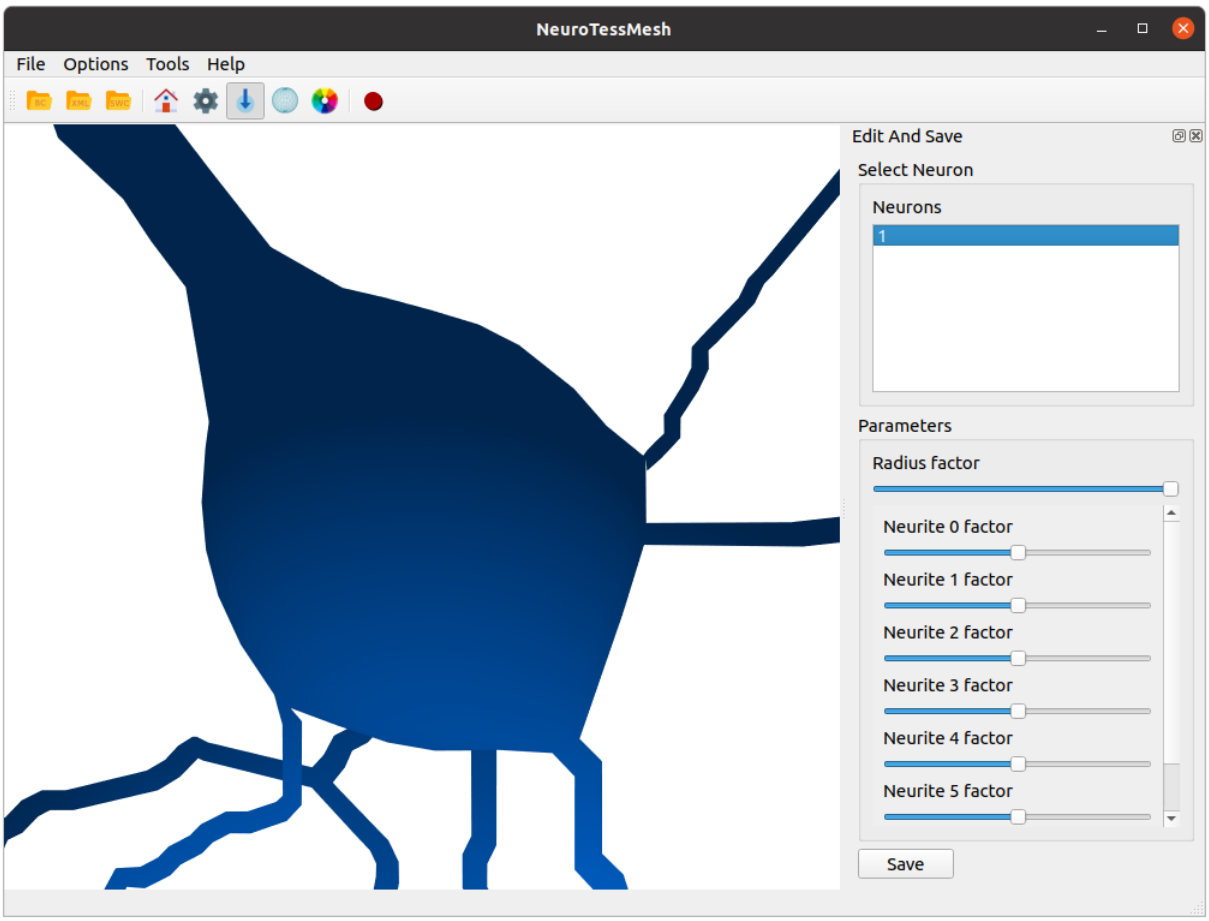

Fig. 2.4: Edit and save panel.

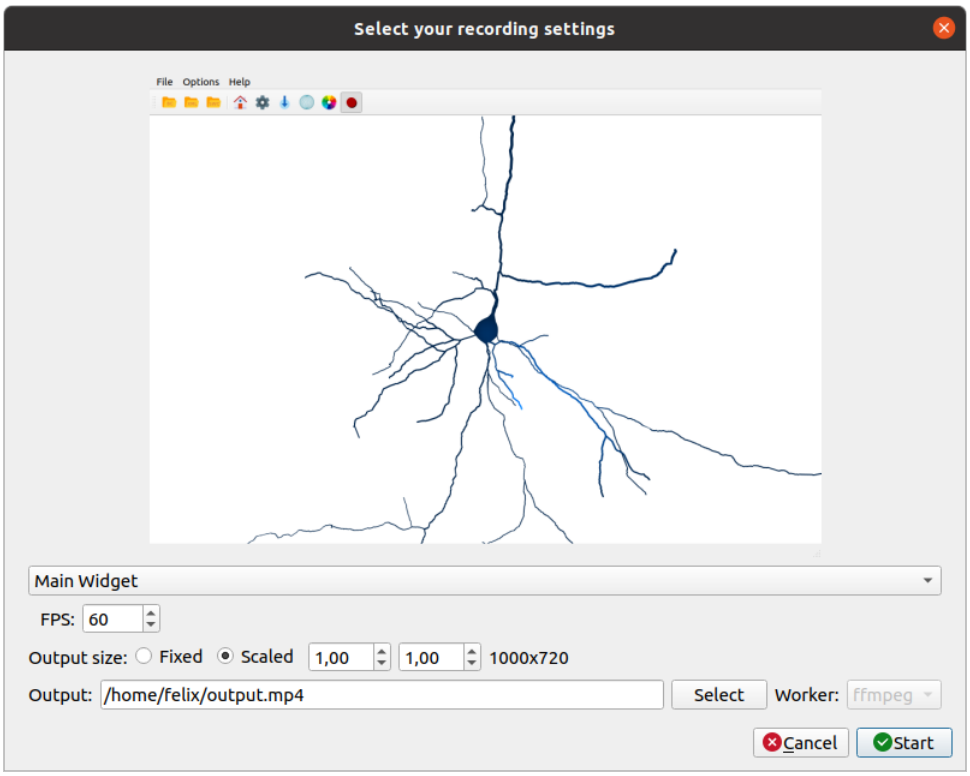

Fig. 2.5: Recorder standard configuration dialog.

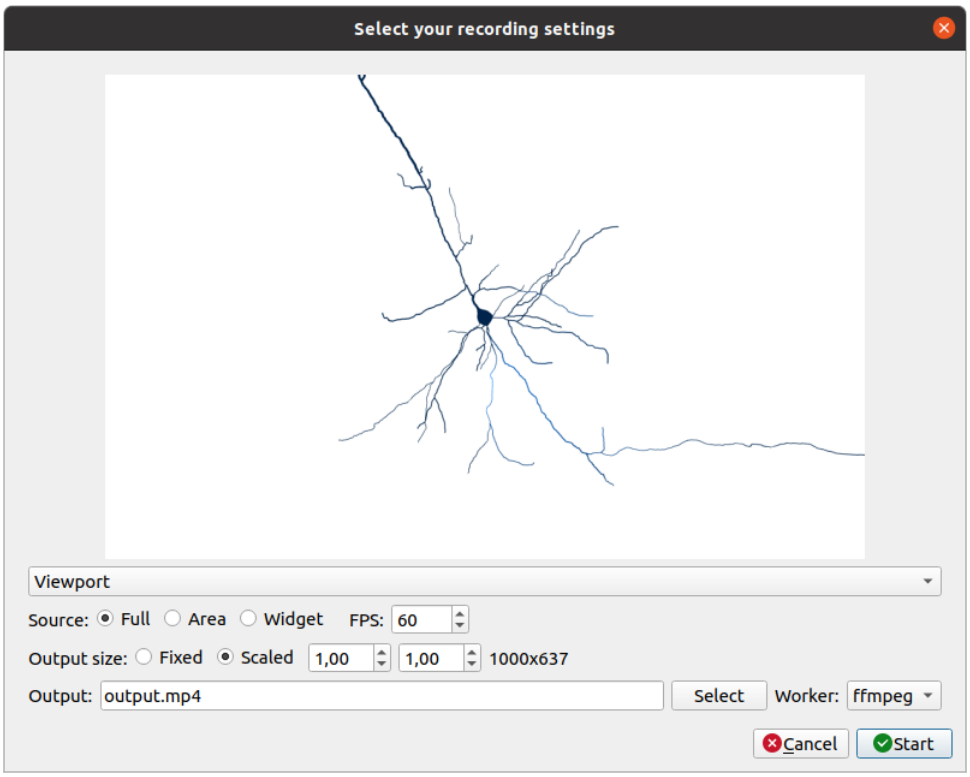

Fig. 2.6: Recorder advanced configuration dialog.

recording has finished it will change to the selected output name in the configuration dialog. The application will warn if the user tries to exit the application while a recording is being made.

Warning: To record a video the system must have **ffmpeg** installed, preferably with Nvidia hardware acceleration. If ffmpeg is not available in the system only the **images** worker will be available in the recorder.

2.3 Keys and shortcuts

- **Open BlueConfig**: Ctrl + Shift + B
- **Open XML Scene**: Ctrl + Shift + X
- **Open SWC**: Ctrl + Shift + S
- **Exit application**: Ctrl + Shift + Q
- **Home (reset view)**: Ctrl + H
- **Update on idle**: Ctrl + I
- **Show FPS on idle update**: Ctrl + S
- **Show wireframe**: Ctrl + W
- **Render options**: Ctrl + B
- **Edit and save**: Ctrl + M
- **Configuration**: Ctrl + C
- **Toggle Recorder**: Ctrl + R

CHAPTER

THREE

RUNBOOK

Note: This runbook will test the main functionalities of NeuroTessMesh and can be considered as a basic tutorial for the application, but **not** as a complete and exhaustive tutorial of all its functionalities.

1. Load a “swc” file (Fig. 3.1).

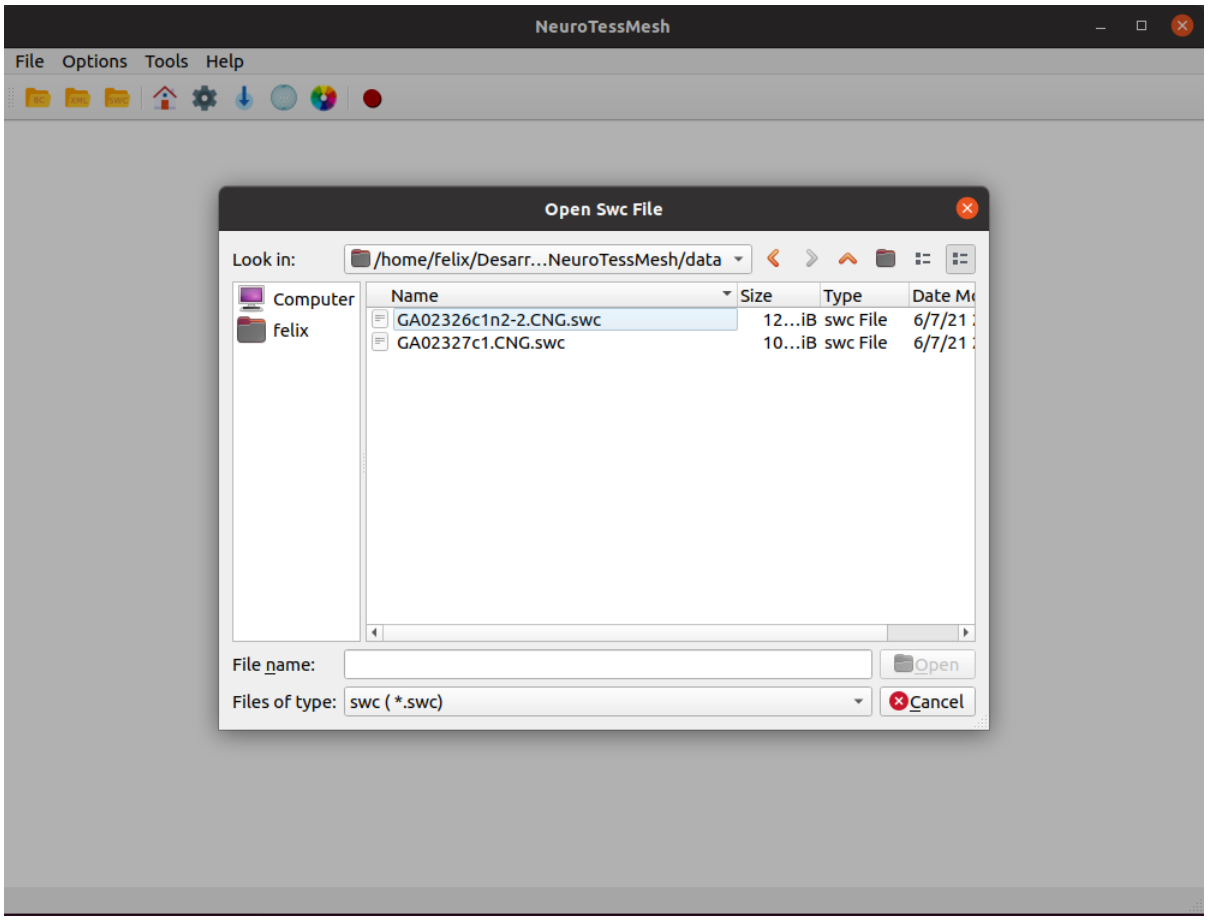

Fig. 3.1: Opening a file.

2. Visualize and navigate through the reconstructed neuronal mesh using the scene camera controls as previously explained in user interface section (Fig. 3.2).

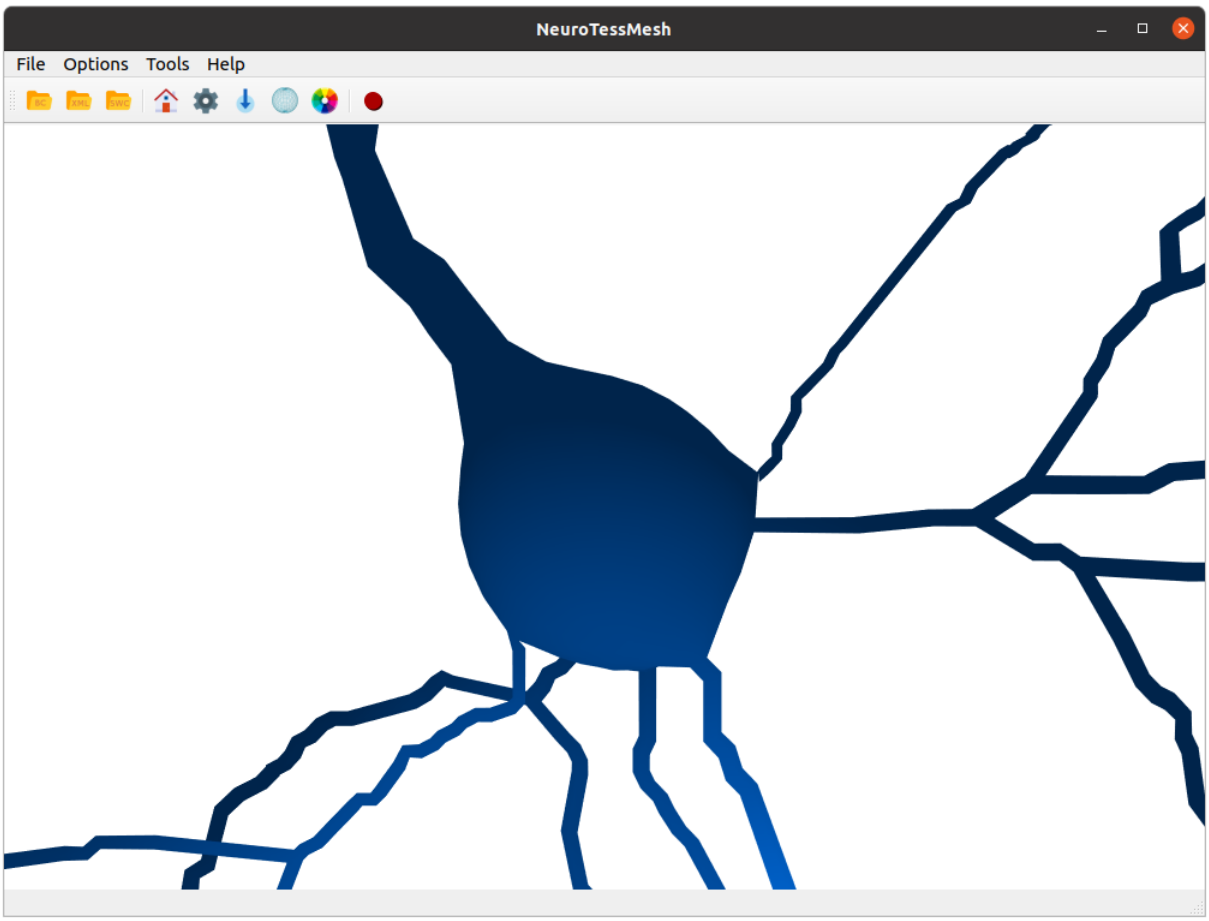

Fig. 3.2: 3D View navigation.

3. Change the render options. In this example the Subdivision level has been increased to 10, the Distance threshold has been also increased and the Tessellation criteria has been modified to Homogeneous criteria (Fig. 3.3).

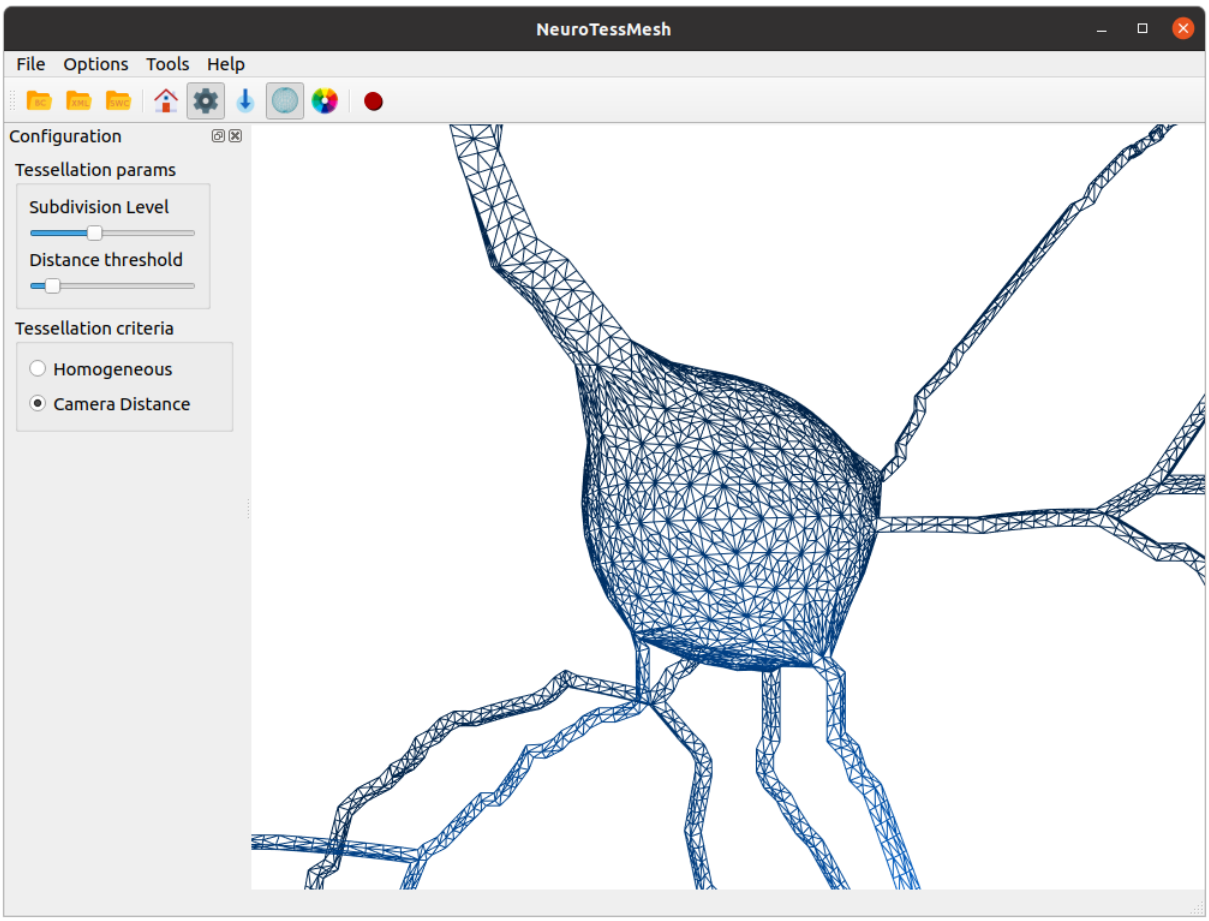

Fig. 3.3: Render options manipulation.

4. Modify the soma reconstruction params. In this example the soma volume has been decreased setting the Radius factor to 0.75 and the starting points of the neurites have been displaced using the Neurite [n] factors (Fig. 3.4).

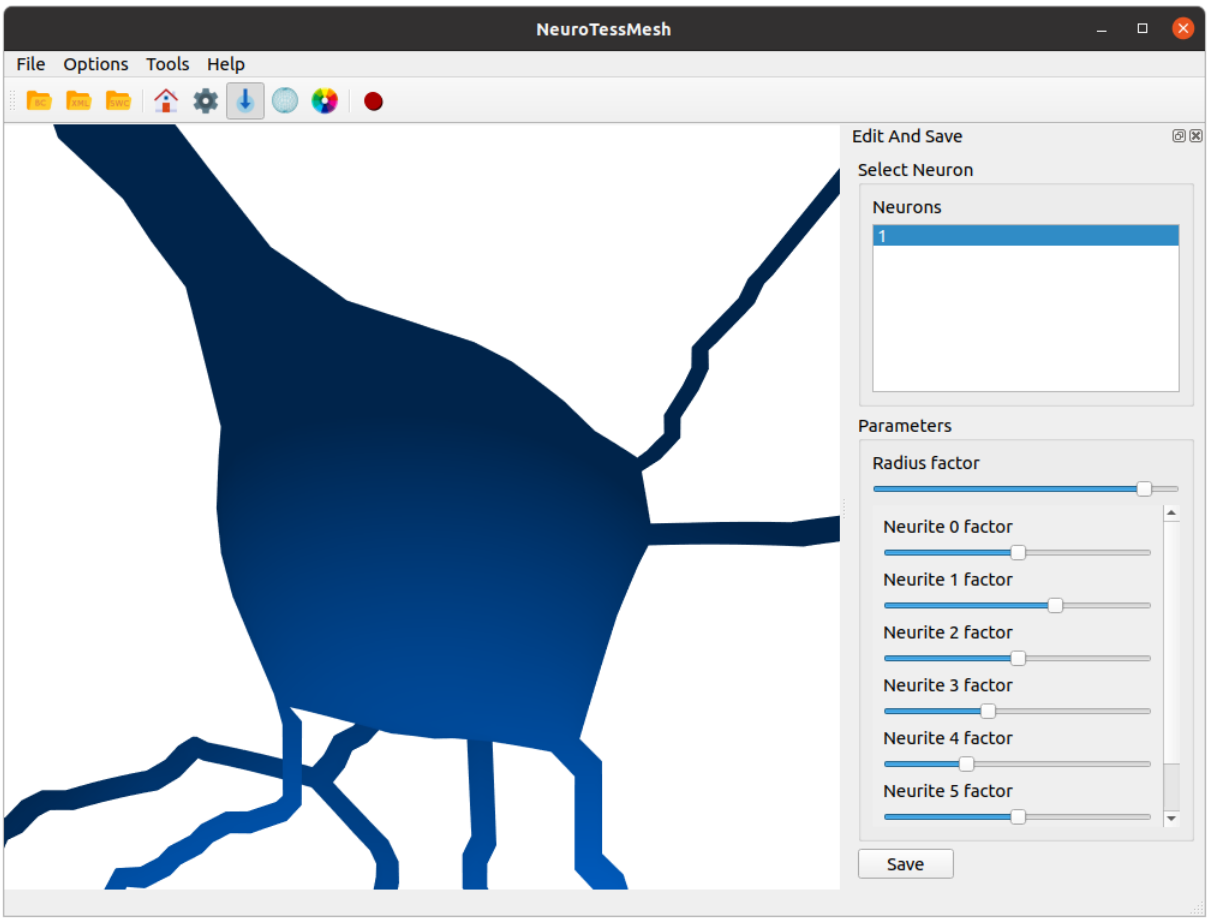

Fig. 3.4: Reconstruction parameters manipulation.

5. Save the results to a “obj” file (Fig. 3.5).

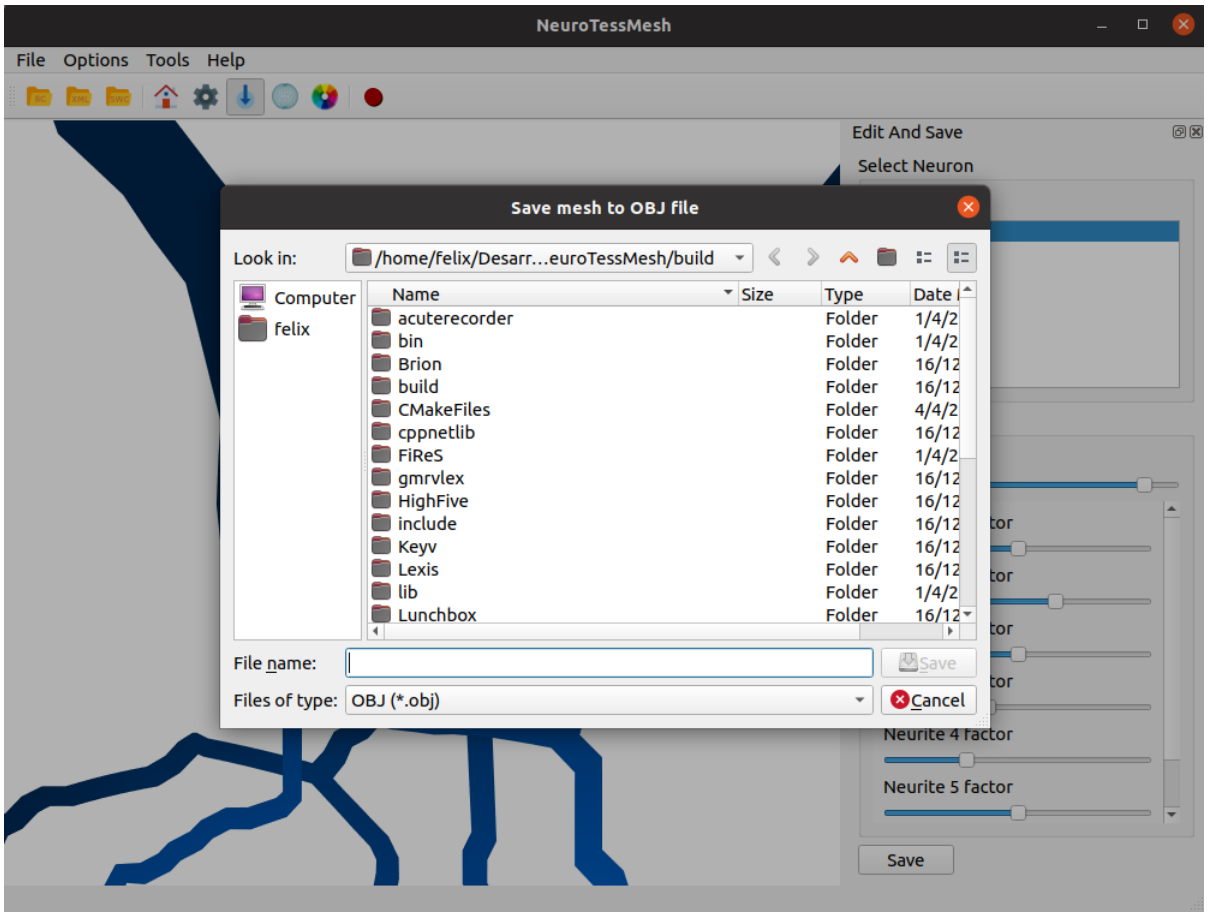

Fig. 3.5: Save results to file.

CHAPTER

FOUR

NEUROTESSMESH FILE FORMATS

This section describes the format of the XML file used to describe a scene composed of several neurons.

4.1 XML Scene file

Note: Work in progress.

CHAPTER

FIVE

ACKNOWLEDGMENTS

This project has been made at the Universidad Rey Juan Carlos partially supported by the Brain Simulation Platform funded from the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 785907 (Human Brain Project SGA2).

