Clara Diederichsen

Clara.diederichsen@deepspace.nz

Protocol for calculating Results

FYP – Mapping Transmitter Range onto Patient specific Brain Topology

To replicate the results described in Section and apply them to a new STL-file the following protocol shall be followed.

Important to note the file convention of using MeshLab\_ for the calculation of mlx file scripts that can be applied within MeshLab using the 2 degrees polynomials and MATLAB\_ for using the best fit polynomials and generating your own STL files.

To get you started an example MeshLab project has been saved called **‘Example.mlp’**. This contains exemplary brain topology and the transmitter.   
  
The folder 1\_MATLAB\_functions contains all the relevant MATLAB functions needed to write and read stl and mlp files.   
  
The folder 2\_MATLAB\_variables contains all the variable backups that ar being created by the MATLAB code.

The folder 3\_ Datafit\_testing contains the files for testing simplex surfaces such as flat and curved surfaces.

The folder 4\_STL\_MATLAB\_output contains the output of the MATLAB\_ files.

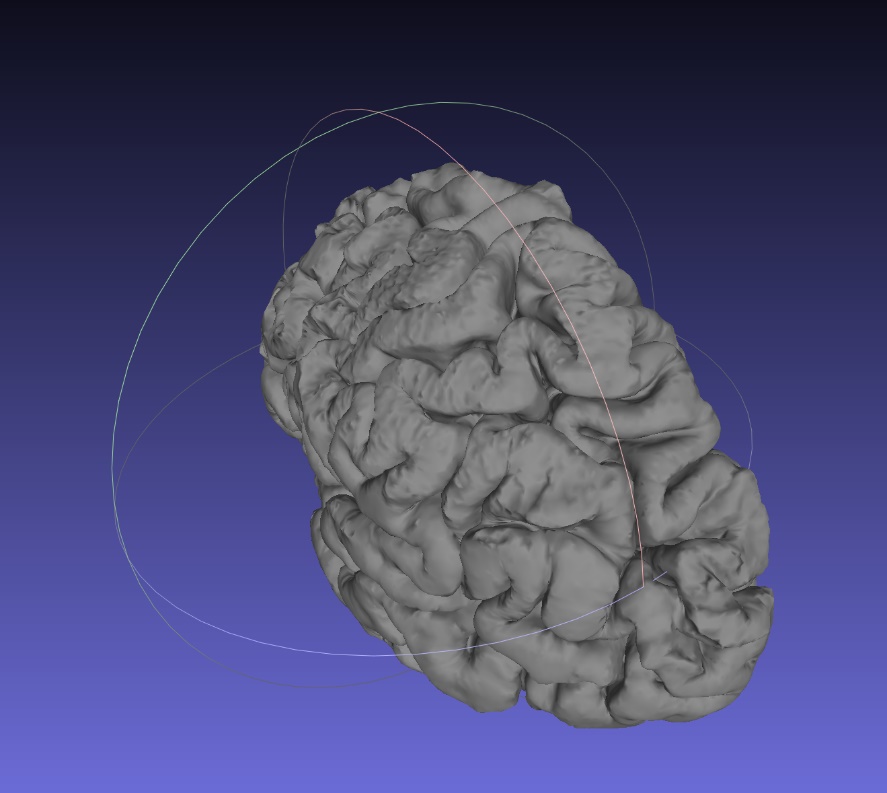
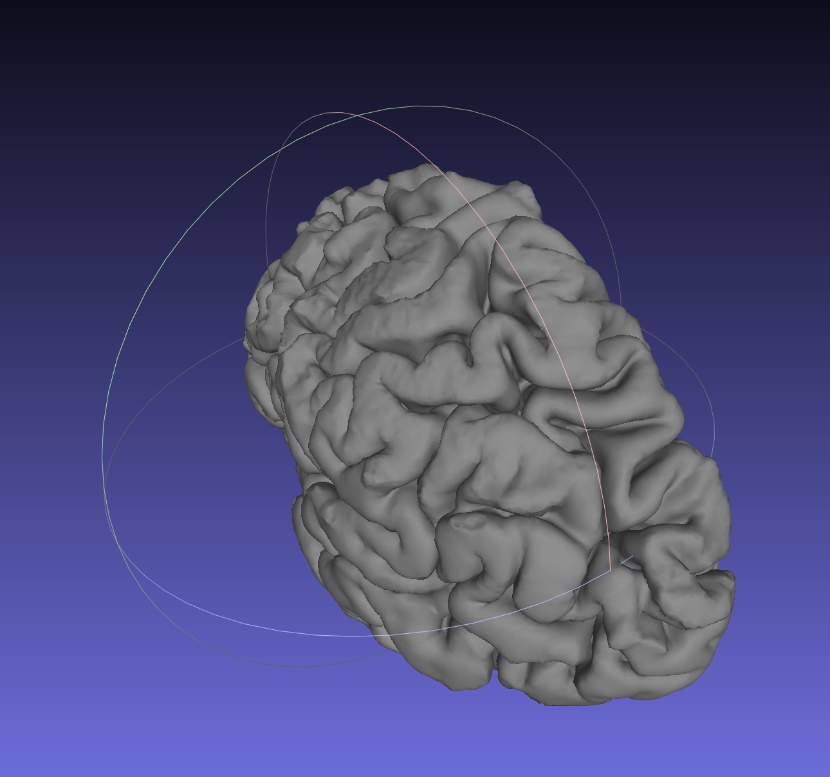
## Pre-processing of a large resolution STL-file

This process is applicable for STL-files which have a resolution larger than 500,000 faces, which is likely to slow down the calculation of the new results significantly and will as such not provide ideal results.

To check the size of the STL-file import it into MeshLab by starting MeshLab and then choosing   
**File->Import Mesh** and choosing the file to be processed. In the bottom right of the UI MeshLab will show a prompt of how many faces it has processed.  
If this number is larger than 500,000 choose the option **Filters -> Remeshing, Simplification and Reconstruction -> Simplification: Quadric Edge Collapse Decimation** from the dropdown menu at the top of the screen.   
Enter 500,000 as the desired **Target number of faces** and checkmark the **Preserve topology** option.  
After choosing to Apply this filter the mesh will be reduced to the target number of faces.

This filter can also be used to reduce the time of executing the MATLAB script to find the final assignment of the transmission scores to different faces but will reduce the resolution quality of the model as shown in Figure 17.

Figure 1: Visible loss of detail through Simplification using the Quadric Edge Collapse Filter and reducing   
the number of faces from 286210 (left) by 50% to 143105 faces (right)



## Calculation of transmission scores

To calculate the transmission scores associated with all the different faces in the model the transmitter position and the transmitter normal must be known.   
If the position of the transmitter is currently unknown, follow the first two steps described in Section 1.3.

### MATLAB Calculation of transmission scores

To calculate the transmission scores using MATLAB and the associated ‘poly45’ models, saved into ‘polynomials.mat’ the STL-file to be processed should be saved in the same folder that the MATLAB files are saved in.

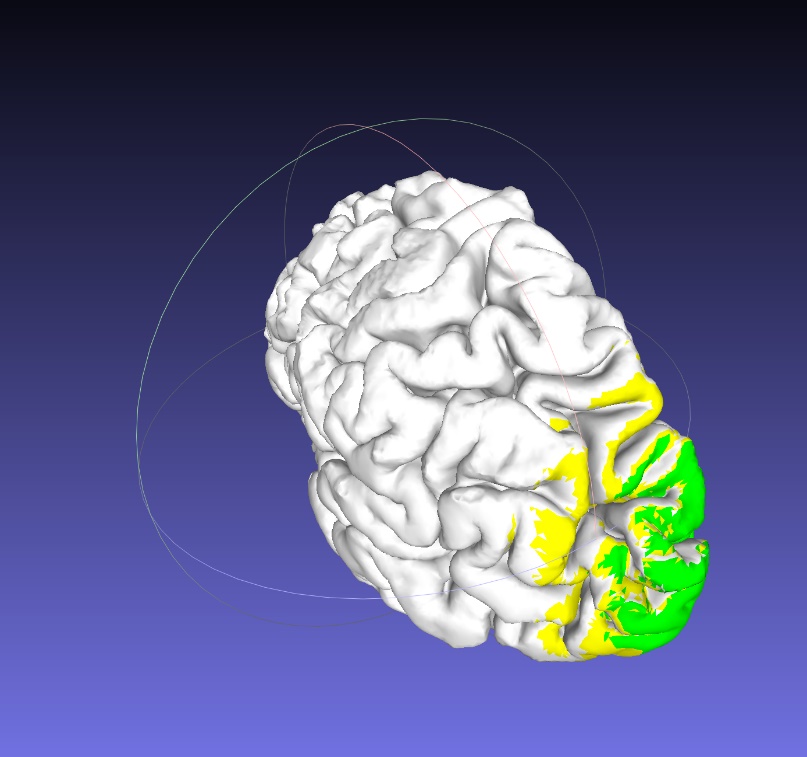
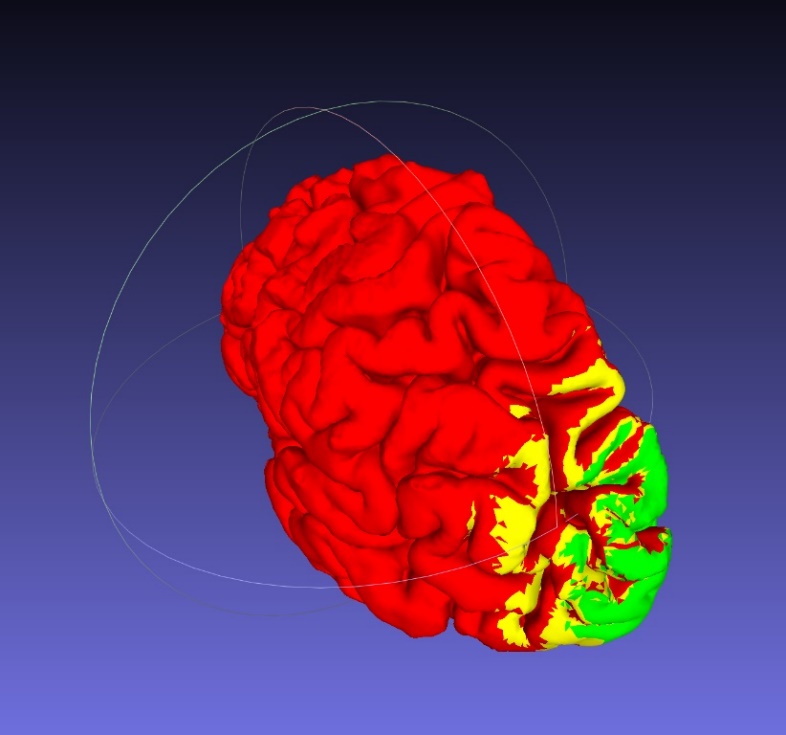
Process the files by opening MATLAB and the .m file called **‘calculate\_transmission\_scores.m’** and running it by using the MATLAB editor or ‘F5’. The code will prompt the user for the name of the STL-file to be processed.

Input the name of the stl file without the stl file ending, iE **topology** for opening the STL-file called ‘topology.stl’.

The user is then prompted to either choose the most recently calculated transmitter position saved as ‘transmitter.mat’ or input the **transmitter position and its normal** of the form of a horizontal vector [x,y,z].

Specify the colour scheme by typing either 1 or 2.

Figure 2: Comparison between the two different color schemes employed with [red/yellow/green] on the left and [white/yellow/green] on the right where both schemes correspond to [no transmission/lossy/lossless]



Once these three inputs have been specified the code will automatically resume execution and evaluation of every face within the STL-file and assigning it its corresponding transmission value.   
The output of the code are three different STL-files that are saved in the colour corresponding to the transmission score.

The three output files created and saved into **4\_STL\_MATLAB\_out** are:

* lole.stl which contains all the faces with a quality score of ‘lossless’ saved in green
* lo.stl which contains all the faces with a quality score of ‘lossy’ saved in yellow
* notr.stl which contains all the faces with a quality score of ‘no transmission’ saved in red/white

To display the results, open all three meshes in MeshLab by opening MeshLab and choosing   
**File->Import Mesh.**To display the position of the transmitter, load the ‘transmitter.stl’ file into MeshLab and save the Meshes as a MeshLab project into the same folder.

### MeshLab Calculation of transmission scores

To calculate the transmission scores using the MeshLab filter, the STL-file to be processed should be saved in the same folder that the MATLAB files are saved in.

To calculate the transmission scores using the MeshLab filter open MeshLab. Import the Mesh to be processed and import the desired STL file to be processed.

Import the **‘transmitter.stl’** file and position it at the desired position using the translate/transform option.

Once the transmitter is in the correct position save the Meshes as a MeshLab project.

Open MATLAB and open the file called **‘MeshLab\_recalculate\_transmission\_scores.m’**.

Follow the prompts by specifying if the transmitter position has been updated using MeshLab or MATLAB. Type **y** to choose an update using MeshLab.

Choose whether you want to process the previously used MeshLab Project. This is saved to be the example project if you haven’t run any of the code yet. Type **y** if you want to analyse this model.

If you are analysing a different model you must type **n** and then input the name of the .mlp file you are working on and specifying the transmitter file position as indicated in Figure 3.

Choose what colour scheme you want the output of the filter to be as shown in Figure 2.

Choose the file name for the generated .mlx script.

Apply the generated Script file by choosing **Filters->Show current Filter Script->Open Script** and selecting the filter that was created by the MATLAB file**.**

Click apply and the model should be updated in the selected colour scheme within seconds.

If it doesn’t apply instantly, make sure that the settings of the are corresponding to the ones shown in and toggle the color selection. If it still hasn’t updated, run the filter by manually putting in the r,g,b strings that MATLAB calculated and input them into the Per Face Colour function found under:

**Filters->Color Creation and Processing->Per Face Color Function**



Figure 3: file position of the transmitter in MeshLab, which in this case is 1

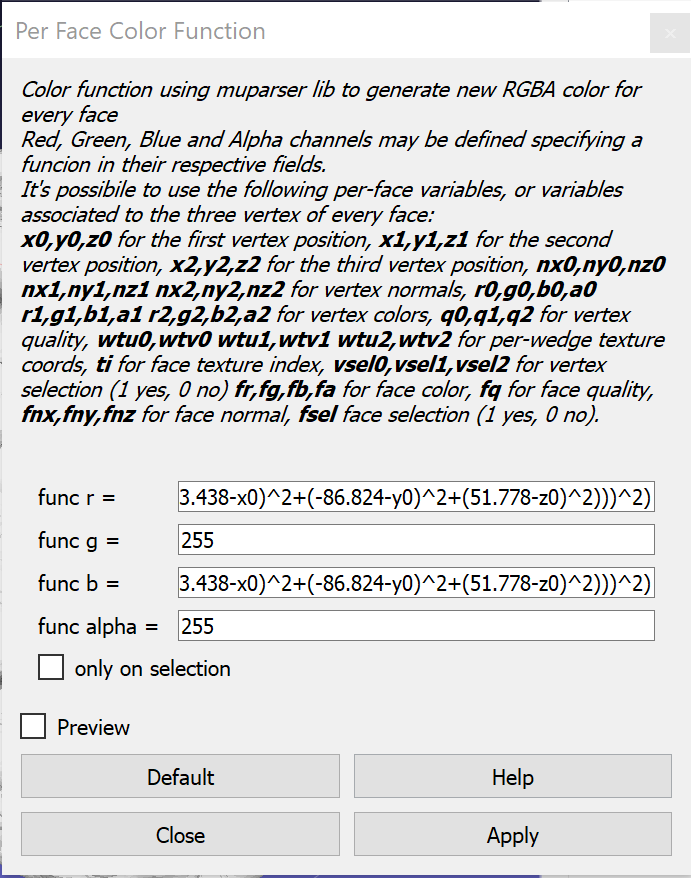


Figure 4: Per face color function manual input

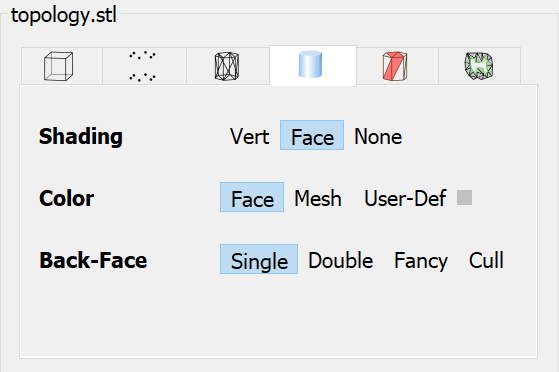


Figure 5: MeshLab settings to display the updated color scheme

## Calculation of transmitter position

If the transmitter position and normal is currently unknown, import the processed STL-files into MeshLab first and then import the **‘transmitter.STL’** file in MeshLab and transform/translate the transmitter to the desired position.

Save the MeshLab project within the same folder.

To recalculate the Transmission scores with a new transmitter position, open the MATLAB file called **‘MATLAB\_update\_transmitter\_pos’**. This file will prompt the user to then input the name of a MeshLab project file. Specify the directory using parenthesis.

The transformation matrix saved in the MeshLab project will be used to calculate the new transmitter and transmitter normal and save these variables into **‘transmitter.mat’**.

These values can be used to apply the Calculation of transmission scores by using either of the protocols described in Section 1.2.

## Recalculation of polynomials

To recalculate the polynomials using a different data set collected, firstly open MATLAB and the MATLAB script **‘datafitting.m’**.

Save the data in a text file called ‘data.txt’ or adjust the file directory specified in the ‘datafitting.m’ file accordingly. The data shall be in the same format specified in the document outlining the collection of the data points [1].

Specify the name of the data file in which you want to save the new polynomials.   
Use ‘polynomials’ for a recalculation using the MATLAB script using any degree of freedom along x and y desired.   
Use ‘polynomials\_22’ for a recalculation of the transmission using the MeshLab filter and specify ‘poly22’. Any number of additional degrees of freedom will be ignored when the MATLAB script to write the muparser R,G,B-strings are generated.

Specify the desired robust fitting method using the variables fit\_lole and fit\_lo.

Verify the correct fit of the data points with the generate plots.

To generate the same evaluation of the goodness of fit open the Curve fitting tool in the Apps tab of MATLAB. Iterate the number of degrees of freedom used for the polynomials and observe the changing of the goodness of fit. Choose the model which provides the highest Adj. R-squared score as well as the lowest RMSE values, indicating a better data fit.

## Quick recalculation

To speed up the recalculation process a script was generated that after the first execution of a MeshLab calculation of the transmission scores, omits the need for reoccurring input by the user. To use this script open **‘MeshLab\_recalculate\_quickly.m’.**

After saving your MeshLab project with the updated transmitter position run the MATLAB code and reapply the .mlx script.

## Analysing a new topology

To analyse a new topology, open the .stl file in MeshLab and then import the transmitter. Save this as a .mlp file and then follow the protocol for analysing the topology using Meshlab outlined in Section 1.2.2.