An Open Source Rendering Pipeline for Neuroimaging Data

ABSTRACT

Despite the existence of high level technologies available for visualising 3D information, imaging data seems confined to proprietary software linked to DICOM and NIFTI files shipped by manufacturers of costly MRI equipment. Another issue that arises from this is that many of these pieces of software are built purely for stability, opting to forego modern rendering solutions that can drastically increase quality of final renders for use in scientific communication and publication, or are wholly confined to 2D stacked image representations. This report describes a methodology that retains the stability and flexibility of the proprietary programs, but uses open source intermediates and modern 3D software to convert MRI information into 3D files that can be rendered in a path traced engine. It is my aim for this to allow for more efficient and effective scientific communication and outreach of neuroimaging data, and for entities with less funding or necessary expertise in neuroimaging to have the ability to create high-quality figures for communication purposes.

METHODS

MRI Data Acquisition and Formatting

By its very nature, MRI data is often protected information and can be understandably difficult to obtain. Within this report the Yale Longitudinal Dataset of Brain Metastases on MRI with Associated Clinical Data *(Chadha et al, 2025)* is used due to their generous open access policies. This workflow will function for any clinical data obtained in the file formats DICOM (.dcm) or NIFTI (.nii).

The example render used here is of Patient ID YG_04YGLO8ATWRL, from 2013-07-25 and uses the PRE scan information. This dataset, alongside many other imaging sets, provides it's data in the NIFTI format to preserve patient anonymity and trim out excess information usually preserved in a DICOM file. Unfortunately, InVesalius (the program we use to create the 3D object necessary for the final render) only accepts imports in a DICOM format. Therefore, we must convert the NIFTI file back to DICOM. This is done via the python program nii2dcm *(Roberts, 2025)*. For further instruction, consult the documentation found on the project's GitHub repository.

DICOM Masking and .STL Conversion

Now in the DICOM file format, we import the folder of stacked images into InVesalius *(Amorim et al, 2015)*. Following a quality check, the images are masked in order to select desired tissues and remove noise. For the purposes of the example render, the default range preset was used although results may be significantly improved with manual masking. After masking, the generated 3D information can be exported as an .STL file for import into Blender *(Blender Team, 2025)*.

Alternatively to automatic masking, I would strongly recommend using the manual masking feature in InVesalius. This workflow has seen significant improvements to ventricular and cortical detail utilizing this instead of simple automatic selection.

Cleaning and Rendering in Blender

Here all of the steps taken to create the final animated render will be described. However, as many bioscientists have limited familiarity with blender, a template file is provided in this projects repository. Upon loading the template file, you must import the generated .STL of the MRI and align it with the rectangle labelled “Occlusion Plane” by placing it approximately above the origin (0,0,0). After this, I recommend cleaning your mesh both manually and using the tools available in Blender as described. First, manually remove any artefact mesh data present outside of the brain model. Next I would advise applying a re-mesh modifier to the brain model to reduce polycount and automatically clean mesh topology. In the example case, this was done via an octree re-mesh using depth 8. Note that this will modify the raw MRI information, technically rendering it less accurate. It will, however, significantly reduce render times and improve visuals. If using the template, this is all that needs to be done. Here, you can render out your final animation and any images and optionally apply any materials you wish to your brain mesh.

In the case of creating the template, after completing topographic cleanup a camera was created and aligned with the desired final angle. Bidirectional soft lights were also added. Finally, a cube was created and linked via Boolean modifier to the brain mesh. This allows rendering the animation of brain slices in 3D. Keyframes were added to the cube, and it was marked non-visible in the render in order for it to occlude the brain mesh. A rudimentary coloured principled BSDF shader was added to the brain mesh and subsurface scattering applied.

RESULTS

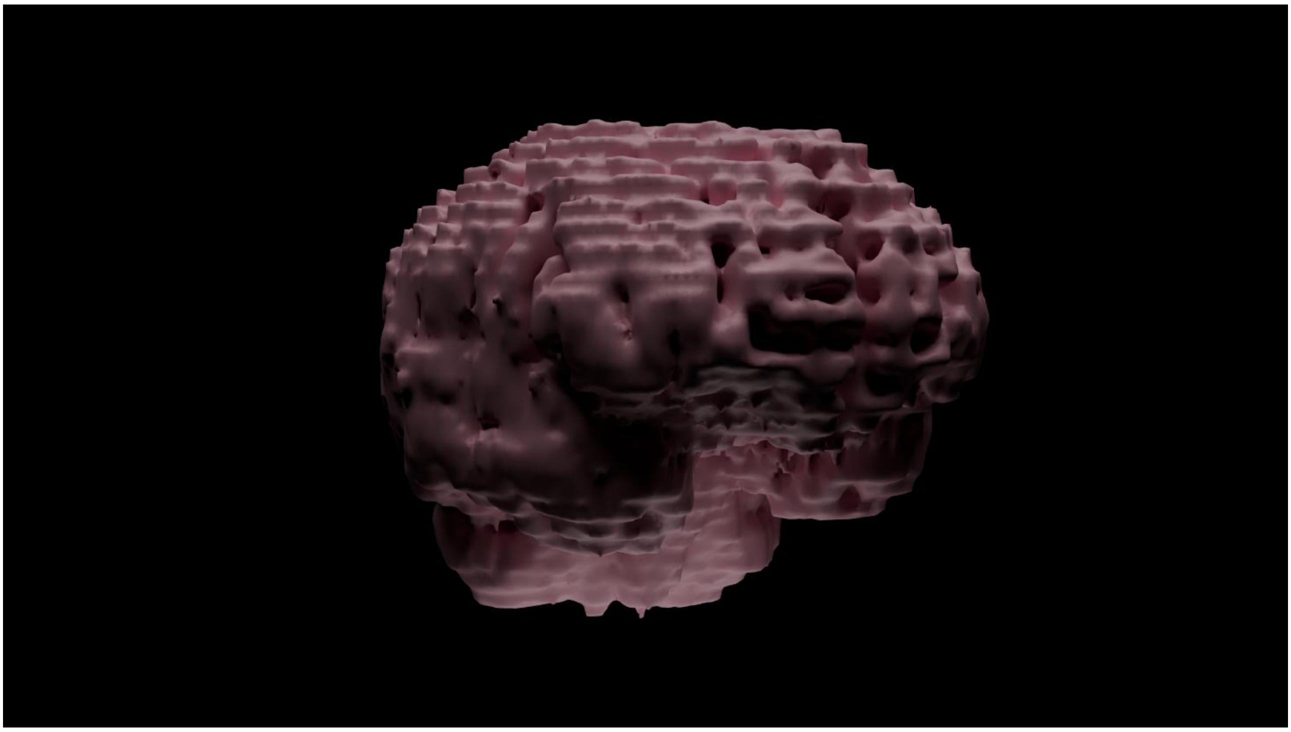

Figure 1: *3D render of MRI brain scan data from Yale Brain Mets Longitudinal Study. Final render performed in Cycles. Accompanying animation available in GitHub repository.*

The above image in *Fig. 1* shows the final render obtained from the pipeline. Note there are still artefacts present in the form of horizontal banding seen around the brain. These artefacts are a result of the data being derived from images stacked in the z-axis. This could be mitigated by obtaining higher resolution MRI data.

For the following, refer to the animation found in the GitHub repository, henceforth referred to as *Fig. A1.* In *Fig. A1,* despite still retrieving a good quality render, we see a general smoothness and lack of ventricular detail. This is likely due to the band-based automatic masking process in InVesalius. This could be addressed via either manual masking, or alternatively use of a more advanced open source software such as SPM if format allows.

DISCUSSION

Despite obtaining a high quality render that is certainly suitable for scientific communication purposes, there is certainly more that could be extracted from this pipeline. Lack of interior ventricular detail and artefacts inherently present due to file format could be mitigated, and if so would drastically improve render quality. This will be thoroughly investigated and this repository updated in the near future.

Currently under investigation as a significant potential improvements would be integration of SPM to allow for more accurate topological exports. This program would also perfectly fit the aim of this pipeline as it is also open source. For a list of improvements I aim to make, please consult the repository readme file.

In the meantime, I hope that this provides a relatively simple method of creating more visually appealing renders of MRI information for scientific communication through a methodology which is entirely free and open source. My sincere acknowledgements to the creators of Blender, InVesalius and nii2dcm for creating the invaluable tools used in this pipeline.

REFERENCES

Amorim, P., Moraes, T., Silva, J., Pedrini, H. 2015. *InVesalius: An Interactive Rendering Framework for Health Care Support.* Lecture Notes in Computer Science **9474**:45-54.

Blender Team. 2025. *Blender Documentation*. Retrieved from https://docs.blender.org/. [Accessed July 6, 2025].

Chadha, S. et al. 2025. *Yale longitudinal dataset of brain metastases on MRI with associated clinical data (Yale-Brain-Mets-Longitudinal)*. The Cancer Imaging Archive. doi: 10.7937/3YAT-E768.

Roberts, T., 2025. *Nii2Dcm.* GitHub Repository. Retrieved from: https://github.com/tomaroberts/nii2dcm. [Accessed July 10, 2025].