

A High-Resolution Head and Brain Computer Model for Forward and Inverse EEG Simulation

Alexandra Warner, Chris Johnson, Brett Burton, & Jess Tate
University of Utah, Salt Lake City UT

May 29, 2018

1 Introduction

Head models are used in many research areas, including imaging, mesh generation, visualization, and any bioelectric research topic that requires simulations such as forward and inverse problems.¹ Imaging is necessary for building any head model, and finding a dataset that includes all imaging modalities is rare since most research projects will include only the modalities necessary to the specific project. For each new project a new head model must be built, which is a time-consuming task.

In this project we describe a detailed, comprehensive pipeline to build a complete, high-resolution head model. Steps of the pipeline include image acquisition, preprocessing and registration, image segmentation, finite element mesh generation, simulations, and visualization. This paper describes two three-dimensional tetrahedral finite element meshes of different resolutions that serve as volume conductors to solve the forward problem. The simulations result in visualizations of the forward problem, with isotropic and anisotropic systems; functional image data mapped onto a tetrahedral mesh; electroencephalography (EEG) signals mapped onto net electrodes; and diffusion tensor data. Lastly, we create a complete, high-resolution brain segmentation that can be used to create three-dimensional tetrahedral volume and surface meshes.

Along with outlining the pipeline, this project will be released as an open-source dataset. The project took roughly a year to complete due to the many options in software and techniques, as well as the complexity of the data. The segmentation of the image data took roughly 100 hours, most being dedicated to manual editing. The mesh generation required several weeks and iterations for an appropriate size without holes. Since the segmentation was complex and high resolution, the tetrahedral meshes became extremely large, which made for very slow simulations. Further simplification methods failed because of the number of different tissues in the head. Registration of diffusion tensor data and functional MR data was difficult because transformations from other software packages were not compatible with the SCIRun problem-solving environment,² and functional MR data has not previously been used in SCIRun.

2 Methods

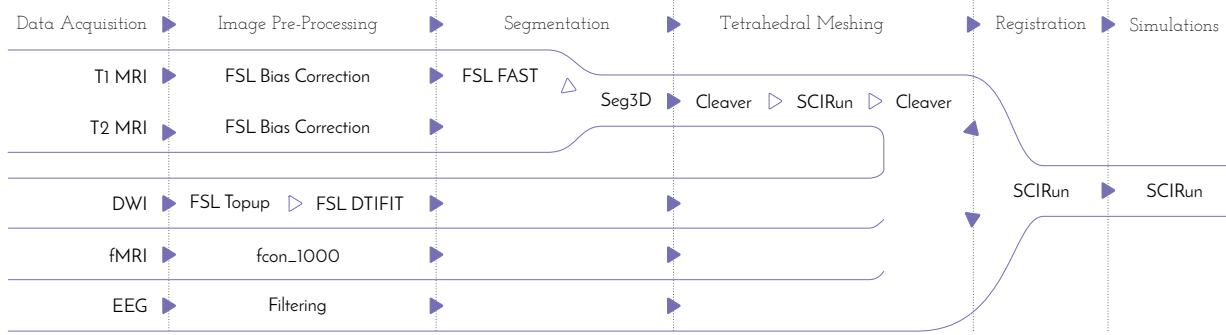


Figure 1: Comprehensive head/brain model pipeline

2.1 Data Acquisition

To construct a high-resolution, personalized, anisotropic volume conductor whole-head model, T_1 -, T_2 - weighted, diffusion weighted, and functional magnetic resonance images (MRI) were acquired on a healthy, female subject who is 23 years of age on a Skyra 3T full-body scanner (Siemens Medical Solutions, Erlangen, Germany).

The T_1 -weighted scan was performed with a three-dimensional magnetization-prepared, rapid-gradient echo (MPRAGE) sequence.³ The parameters used were as follows: echo time: 3.41ms; repetition time: 2500ms; flip angle: 7 °; resolution matrix size: 256x256 pixels; field of view: 256mm; 208 sagittal slices with a slice thickness of 1mm. Acquisition time was 10:42 minutes.

The T_2 -weighted scan was performed with a SPACE - sampling perfection with application-optimized contrast using different flip angle evolutions - sequence.⁴ The parameters used were as follows: echo time: 406ms; repetition time: 3200ms; resolution matrix size: 256x256 pixels; field of view: 256mm; 208 sagittal slices with a slice thickness of 1mm. Acquisition time was 5:34 minutes. The subject did not move in between the two scans so the scans did not need to be registered.

The diffusion weighted images (DWI) were acquired with multiband two-dimensional echo-planar imaging (EPI).⁵ Both phase encoding directions were performed (anterior to posterior and posterior to anterior) with 64 diffusion directions each. Further sequence parameters for each scan were as follows: echo time: 76.8ms; repetition time: 4070ms; flip angle: 90 °; resolution matrix size: 104x104 pixels; field of view: 208mm; 60 slices with 2.5mm slice thickness. Acquisition time was 5:05 minutes each.

The functional MRI (fMRI) scans were acquired with a blood oxygenation level dependent contrast (BOLD) sequence. The following parameters were used: echo time: 76.8ms; repetition time: 780ms; flip angle: 55 °; resolution matrix size: 104x104 pixels; field of view: 210mm; 72 slices with 2mm slice thickness. Acquisition time was 10:32 minutes.

A continuous electroencephalogram (EEG) was recorded using a 256-channel HydroCel Geodesic Sensor Net that was connected to a NetAmps 400 amplifier and referenced online to a single vertex electrode. Channel impedances were kept at or below 50 kOhms and signals were sampled at 250Hz. The EEG was recorded while the subject sat quietly in a chair, alternating two-minute epochs of eyes open and eyes closed, for a total of twelve minutes.

All acquisition reports will be included with the dataset.

2.2 Preprocessing of Images

2.2.1 MRI Correction

Bias field signal is a low-frequency, smooth signal that corrupts MRI images due to inhomogeneities in the magnetic fields of the MRI machine by blurring images, thereby reducing the high frequencies of the images, such as edges and contours. The signal changes the intensity values of image pixels so that the same tissue has a different distribution of grayscale intensities across the image.⁶ We applied an estimated bias field correction on the T_1 and T_2 MRIs using FMRIB Software Library (FSL) FAST,⁷ which will be further described in Section 2.3. FSL’s basic user interface, which was used throughout this pipeline, is shown in Figure 2.

2.2.2 DWI Distortion Correction

DWIs performed with EPI sequences are prone to distortions from rapid switching of diffusion weighting gradients, movement from the scanning table, and movement from the subject. The diffusion data was collected with reversed phase-encode blips (anterior to posterior (AP) and posterior to anterior (PA)), resulting in pairs of images with distortions in opposite directions. From these pairs, we estimated the susceptibility-induced off-resonance field using a method⁸ similar to what is currently implemented in FSL.⁹ We then combined the two images into a single corrected one using FSL’s topup and eddy command line tools.

Before running these tools, we created an acquisition parameters text file with the FSL-defined total readout time. Two parameters are frequently required to calculate and apply field maps: the effective echo spacing and the total readout time for an EPI sequence. We used “effective” echo spacing, rather than the actual echo spacing, in order to include the effects of parallel imaging, phase oversampling, etc. We defined “effective” echo spacing as:

$$\text{Effective Echo Spacing (s)} = 1 / (\text{BandwidthPerPixelPhaseEncode} * \text{MatrixSizePhase})$$

The total readout time (FSL definition) was:

$$\text{Total readout time (FSL)} = (\text{MatrixSizePhase} - 1) * \text{EffectiveEchoSpacing}$$

The software package MRIConvert provided the acquisition information about a dicom series, as well as converted the MRI to a NiFTI format, including effective echo spacing and total readout time.¹⁰ To obtain this information, we loaded the dicom series for either DWI acquisition. We selected “Options” to ensure the DWI was saved in NiFTI format. We then selected “Convert All” to save all the files into the output directory specified upon opening MRIConvert. The text file included the FSL-defined total readout time, which was contained in the acquisition parameter file in seconds. MRIConvert also output the b-values and b-vectors files, which were the same for both the DWI AP and DWI PA scans. The last input file required was an “index.txt” file, which contained one column with 65 rows (for 64 directions plus the b0 image) of 1’s.

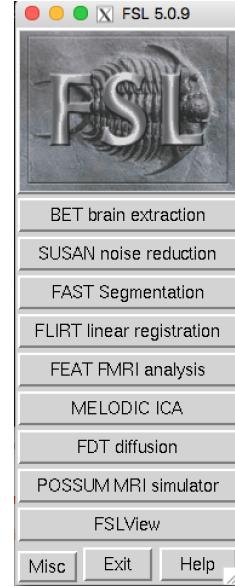


Figure 2: FMRIB software library user interface

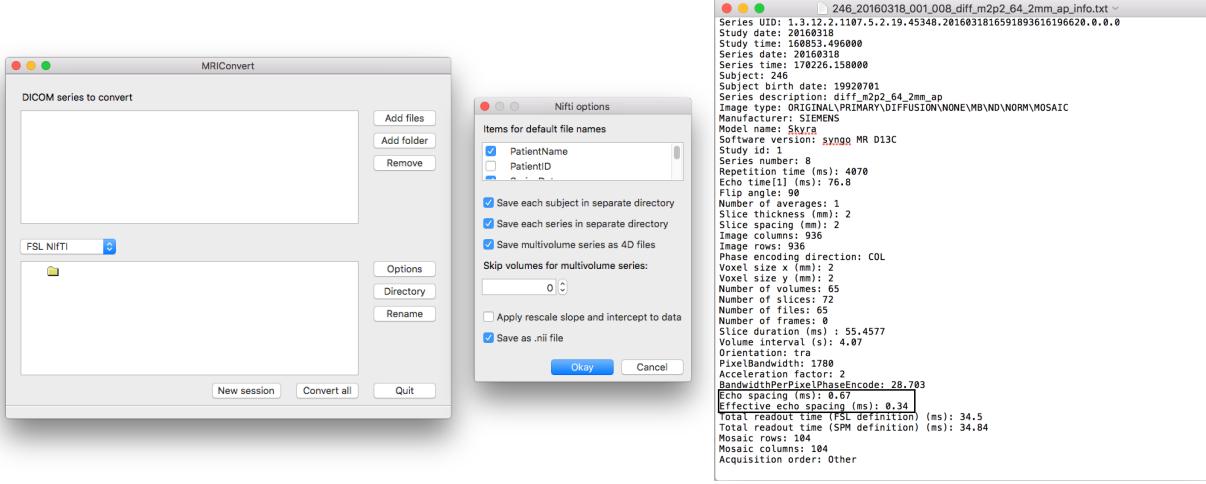


Figure 3: MRIConvert (*left*) with options (*middle*) & output (*right*)

```
0 -1 0 0.0345
0 1 0 0.0345
```

Figure 4: Acquisition parameters text file

We made a separate folder for topup results that included the following files: the acquisition parameters file; the index file; b-values; b-vectors; and the DWI AP and DWI PA files. For the following commands, we renamed the DWI AP image DWI_up and DWI PA image DWI_down. We renamed the b-values and b-vectors dwi.bval and dwi.bvec, respectively. After all files were in place, we executed the following command line commands:

```
fslroi DWI\_up b0\_up 0 1
fslroi DWI\_down b0\_down 0 1

fslmerge -t both\_b0 b0\_up b0\_down

topup --imain=both\_b0 --datain=acq_params.txt --config=mine.cnf --out=topup\_results
applytopup --imain=b0\_up,b0\_down --inindex=1,2 --datain=acq_params.txt
--topup=topup\_results --out=b0\_hifi

bet b0\_hifi b0\_hifi\_brain -m -f 0.2
eddy --imain=DWI\_up --mask=b0\_hifi\_brain\_mask --index=index.txt --acqp=acq_params.txt
--bvecs=dwi.bvec --bvals=dwi.bval --fwhm=0 --topup=topup\_results --flm=quadratic
--out=eddy\_unwarped
```

By running these commands, we first obtained the b0 image, which is the baseline image used for calculating field maps, for both encoding directions. Then the two b0's were merged together into one file, topup and eddy were applied for distortion correction, and ‘bet’ was applied for brain extraction. The distortion corrected file was named “eddy_unwarped.nii.”

2.2.3 Diffusion Tensor Images

After we corrected the DWI images, we calculated diffusion tensor images (DTI) using FSL’s DTIFIT toolbox.¹¹ Upon opening FSL, we chose “FDT Diffusion,” followed by “DTIFIT Reconstruct diffusion tensors” in the drop-down menu and input files manually; Table 1 lists the files selected.

Table 1: DTIFIT input files

Diffusion weighted data	eddy_unwarmed.nii
BET binary brain mask	b0_hifi_brain_mask.nii
Output basename	desired output location
Gradient directions	dwi.bvec
b values	dwi.bval

DTIFIT output the eigenvalues (named L1, L2, and L3) and the eigenvectors (named V1, V2, and V3) for the diffusion tensor field. We converted the files from NiFTI format to nrrd format using ITK-SNAP,¹² although there was a loss of precision. We then input the files into SCIRun to build the tensor field using the eigenvalue and eigenvectors. The SCIRun “CalculateFieldData” module requires only two eigenvectors as input because it calculates the third eigenvector automatically since it is orthogonal.

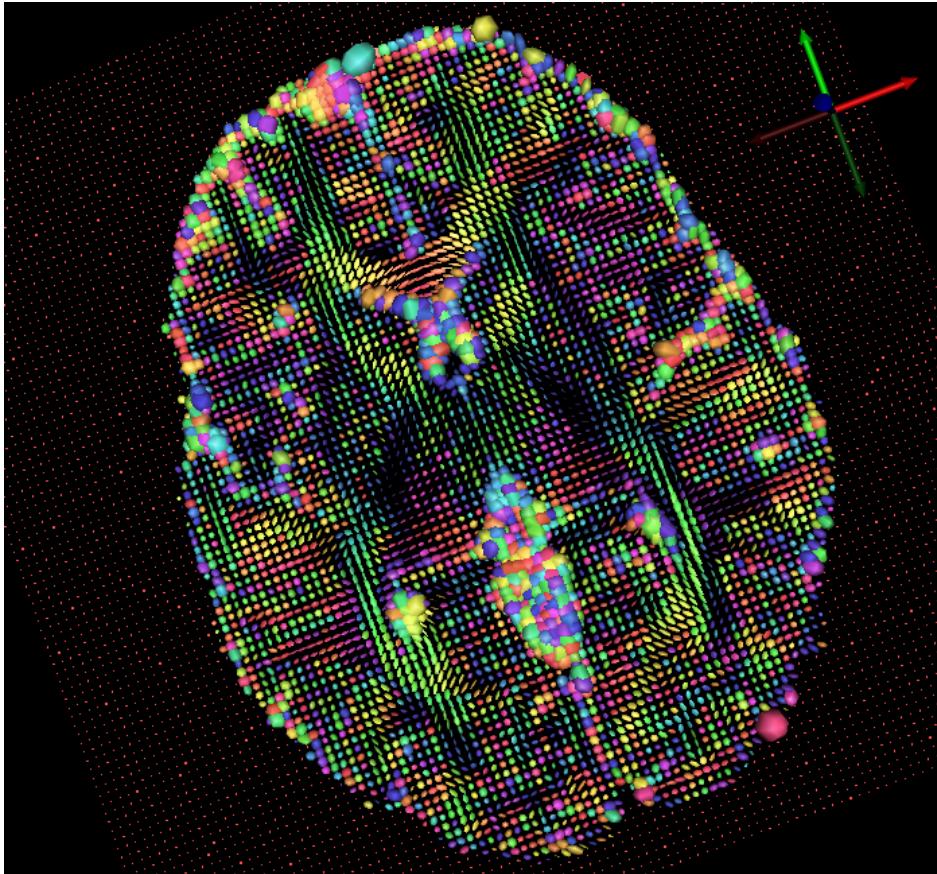


Figure 5: Diffusion tensor visualization using SCIRun

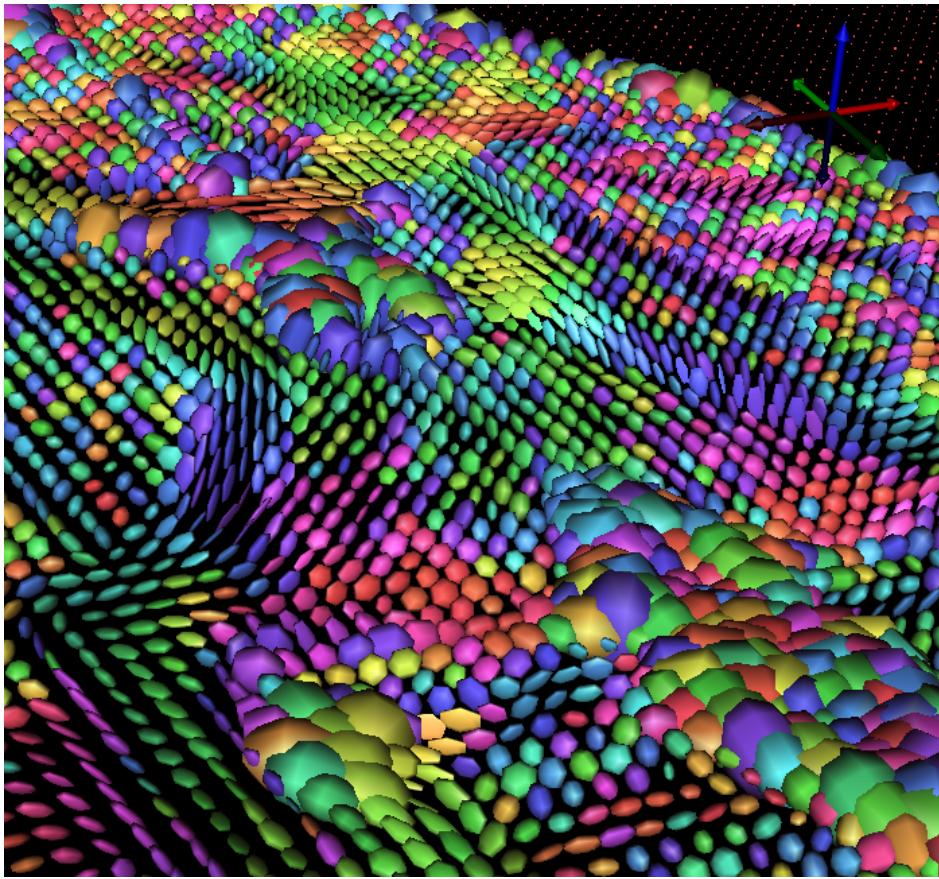


Figure 6: Diffusion tensor visualization using SCIRun

We built the tensor field in SCIRun rather than in 3D Slicer¹³ or FSL DTIFIT because the output data was “backwards” and could not be registered with the mesh in SCIRun.

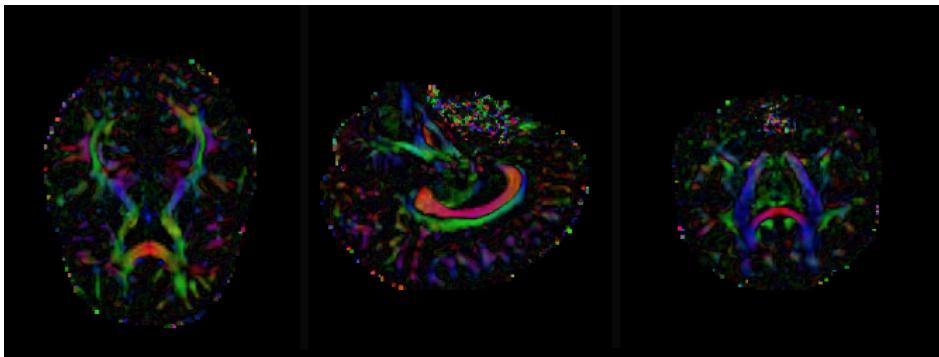


Figure 7: Example of “backwards” data

2.2.4 fMRI

We preprocessed the fMRI data using the 1000 Functional Connectomes (fcon) Project pipeline scripts,¹⁴ which performed anatomical preprocessing, functional preprocessing, registration to the T_1 MRI, segmentation, and nuisance signal regress. The outline pipeline used on this fMRI dataset, specific to the University of Utah, can be found at https://bitbucket.org/UtahBrainNetworks/base_prep, which includes instructions for installation, compilation, and usage.

We opened “rest.nii,” the preprocessed fMRI file, in Matlab using the “load_nii(‘rest.nii’)” function within the NiFTI toolbox¹⁵ after running the fMRI data through the fcon pipeline. We then resized the four-dimensional “img” variable into a two-dimensional variable for use in SCIRun.

2.2.5 EEG

A 60Hz notch filter and its harmonics¹⁶ were applied to the EEG data and output in .edf file format. We calculated the EEG signals matrix using a Matlab script called “edfRead.m.”¹⁷ To run this script, we used: “[hdr, record] = edfread(fname).” The variable ‘record’ contained the EEG signals. We removed the last two rows of the EEG signals matrix as these were control rows for the experiment. We also removed several columns at the beginning and the end of the matrix because these corresponded to taking the EEG net on and off the subject’s head.

2.2.6 Registration

Since the subject did not move in between the T_1 and T_2 MRI, no registration was necessary before segmentation and meshing. We generated the tetrahedral mesh in its own coordinate space from the segmentation, and registered the mesh to the DTI coordinate space with a rigid registration using SCIRun. We registered the fMRI data to the mesh coordinate space with a rigid registration, using SCIRun as well. To register the fMRI data to the DTI coordinate space, the same transformation matrix used to register the mesh can be applied later, if desired. The SCIRun networks for registration are included in Section 2.6.

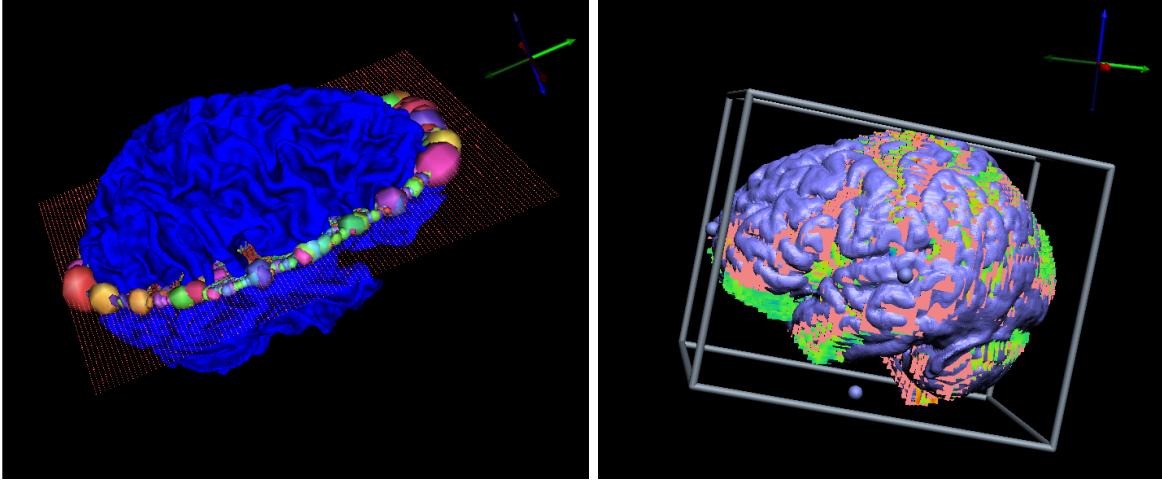


Figure 8: SCIRun manual registrations: mesh to DTI registration (*left*), fMRI to mesh registration (*right*)

2.3 MRI Segmentation of Tissues

Segmentation of the head tissues proved to be the most time-consuming portion of the pipeline. We segmented the head volume into air, cerebral spinal fluid (CSF), white matter, gray matter, skull, sinus, eyes, and scalp. Segmentation of the brain was difficult due to the similar grayscale intensities of different tissues, which means that thresholding the image was not good enough. We did the majority of the segmentation work using Seg3D, a free volume segmentation and processing tool.¹⁸

We initially segmented the brain by inputting a skull stripped T_1 MRI into FSL FAST Segmentation. We skull stripped the T_1 MRI using FSL's brain extraction tool (BET).¹⁹ FSL FAST outputs segmented CSF, white matter, and gray matter layers as well as a bias-corrected T_1 MRI. This method, compared with Freesurfer,²⁰ Statistical Parametric Mapping through Matlab (SPM),²¹ Atlas Based Classification through 3D Slicer,²² and Seg3D methods alone, produced the best initial brain segmentation results for this data.

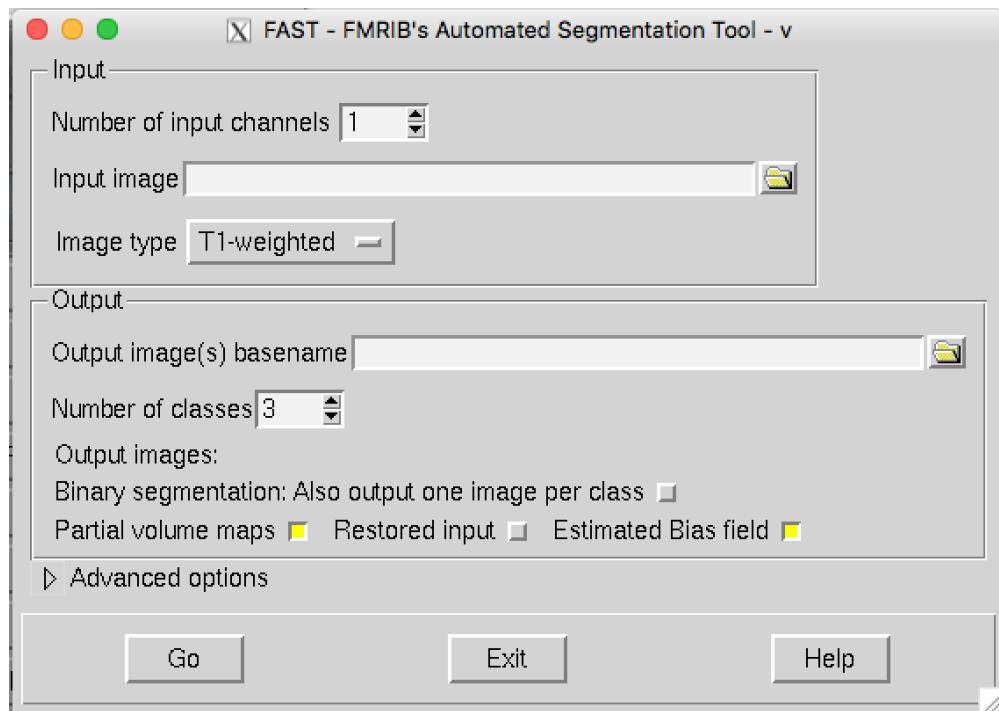


Figure 9: FSL FAST user interface

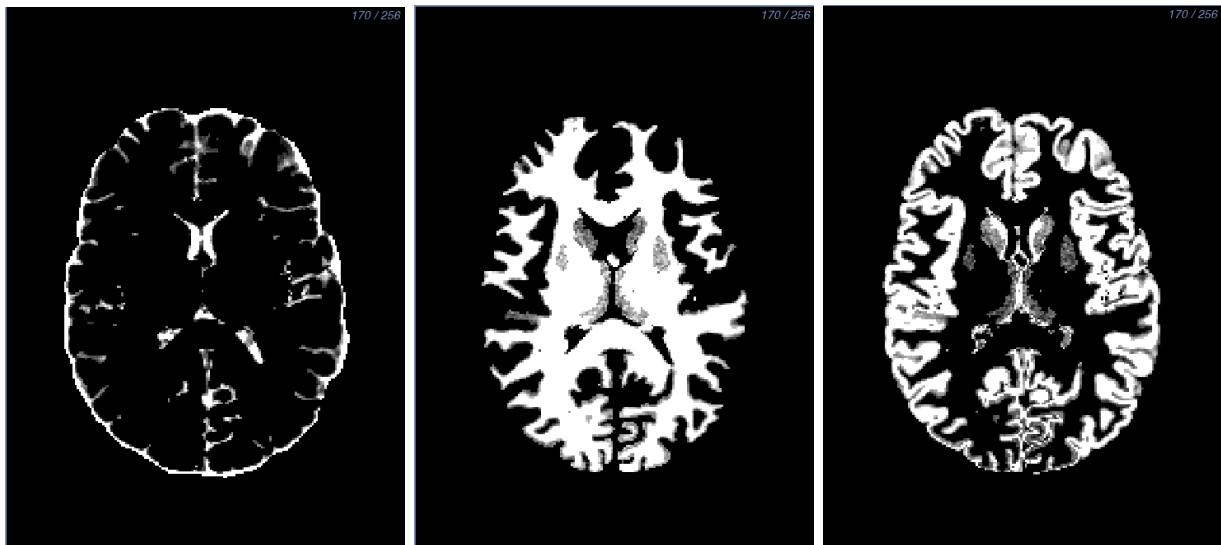


Figure 10: FSL FAST output: CSF (*left*), white matter (*center*), gray matter (*right*)

Although the FSL FAST results were a great improvement compared to the other segmentation software trials, we manually improved the layers to add more detail and to remove any crossover between the layers. We started with the white matter layer because it is the innermost layer. First,

we created a threshold layer FSL FAST output. Then we inspected and manually edited every slice in every direction to add more detail or to clean up noise from FSL FAST.

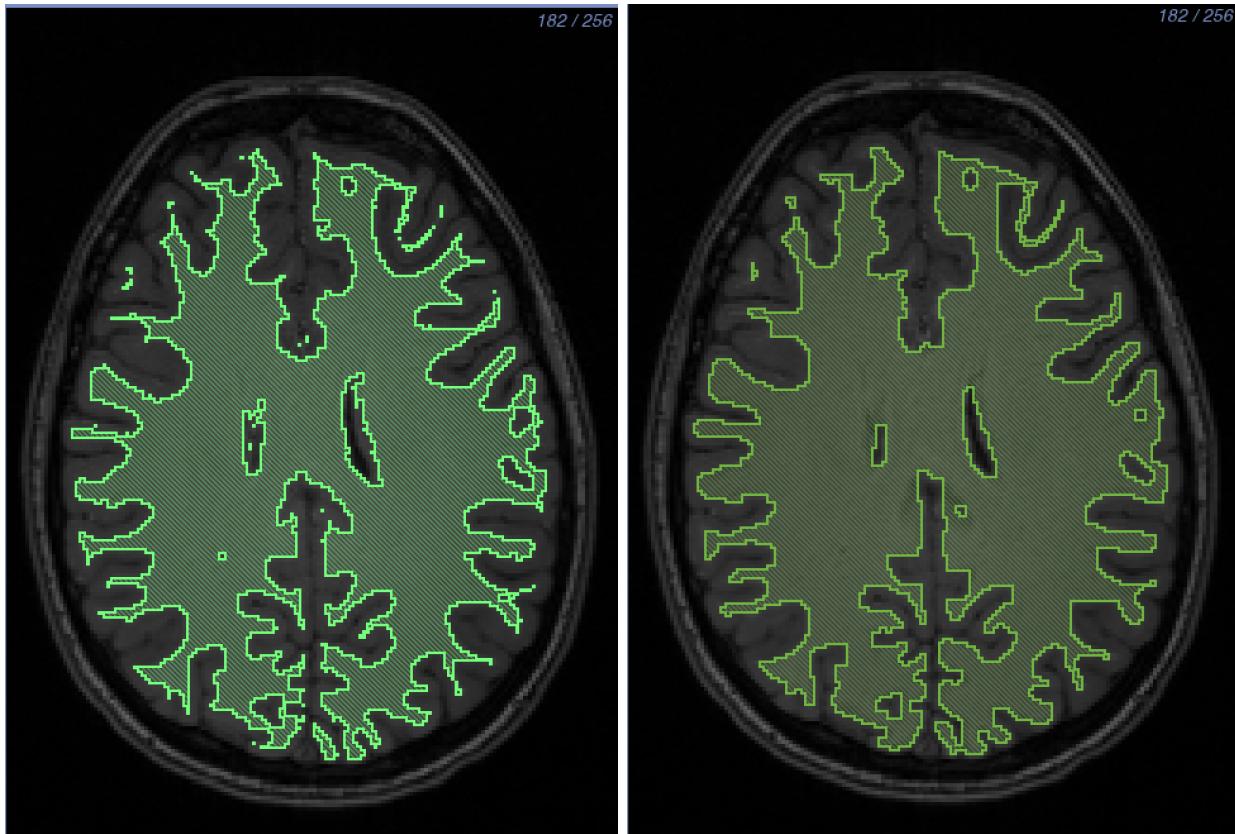


Figure 11: White matter segmentation: Before (*left*) and after (*right*) manual segmentation. The hook feature in the upper right-hand corner is a notable change between the two layers. The layer is more full and has less noise.

After we segmented the white matter, we created a threshold layer from the FSL FAST output for the gray matter. We inspected and manually edited each slice in every direction of the gray matter. We then removed the white matter layer from the gray matter using a Boolean remove mask filter to ensure no overlap between the layers. We manually filled any holes between the two layers. Lastly, we added a gray matter nuclei to the gray matter layer. The thresholding algorithms in Seg3D produced noise around these nuclei because of the similarities of the grayscale values. To fix this noise, we segmented the entire nuclei manually using the paintbrush tool in Seg3D. Then we added the nuclei to the gray matter layer using a Boolean OR mask filter, and removed any overlap from the white matter layer using a Boolean remove mask filter.

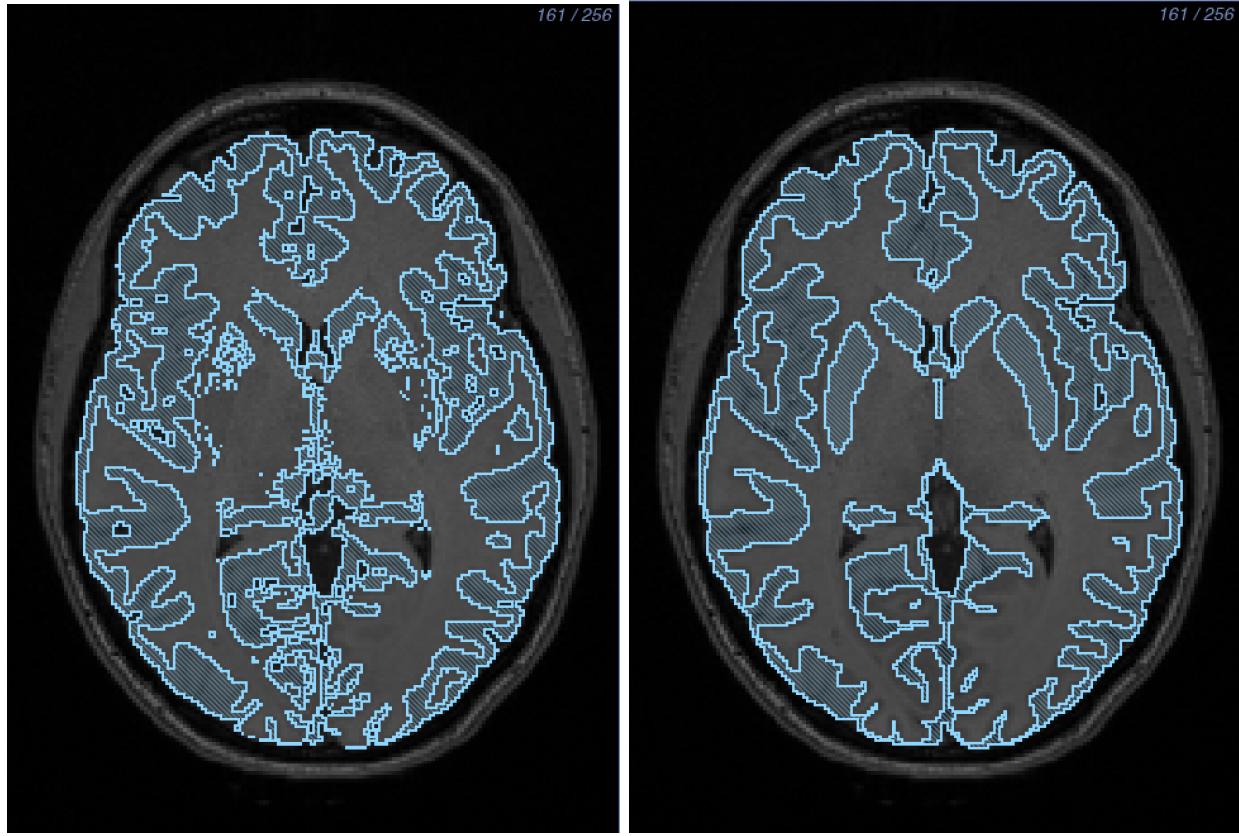


Figure 12: Gray matter segmentation: Before (*left*) and after (*right*) manual segmentation. Gray matter nuclei located in the center of the brain were segmented manually.

After we completed the segmentation of the gray and white matter layers, we made the CSF layer by creating a solid threshold layer for the entire brain and removing the white and gray matter layers using a Boolean remove mask filter. We then checked the white matter, gray matter, and CSF layers for holes, whether on the surface or inside the segmentation between layers. We also performed a quality check on the layers to ensure that they were at least two pixels wide throughout. We chose this check to help create a tetrahedral mesh without holes.

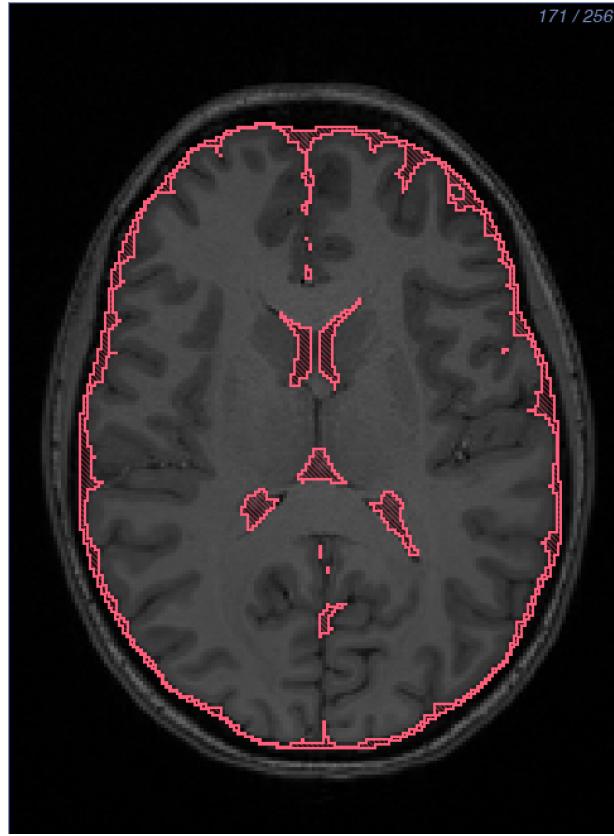


Figure 13: CSF segmentation

The skull and the sinus layers were the most difficult to segment using only an MRI because they both appear black in the MR image, and the subject's data did not include a computed tomography (CT) scan. Our first attempt to create a bone layer applied FSL's skull stripping function by using the BET2 tool to create a skull. We then thresholded the T_1 MRI to create the remainder of the bones in Seg3D and connected the bones to the skull made from FSL. Although this approach gave a decent skull segmentation considering we had only an MR image, our method for segmenting the sinus layer was yet to be determined. As a second method, we estimated the skull from an MR-based synthetic pseudo-CT. We used an improved iterative version of the patch-based method as described by Torrado-Carvajal et al. that takes the T_1 and T_2 images as input and synthesizes the pseudo-CT based on both images, providing more refined and accurate bone boundaries.²³

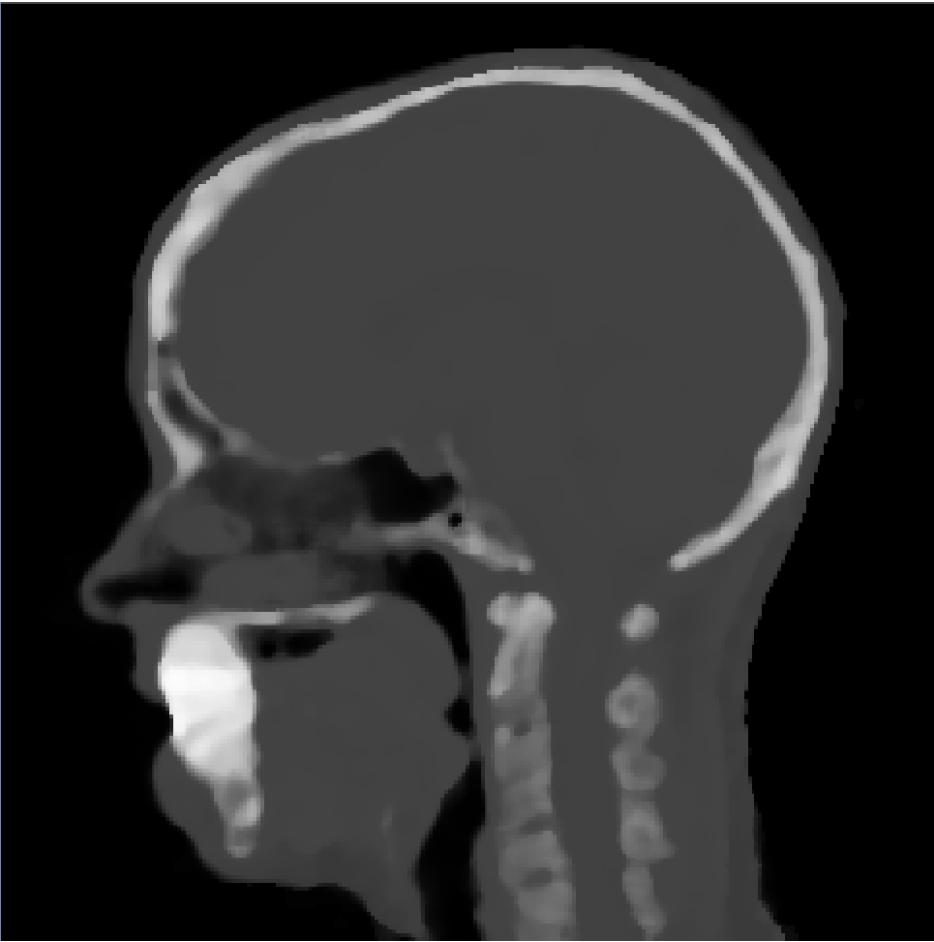


Figure 14: Pseudo-CT scan

The pseudo-CT method was a good starting place for skull segmentation, but we still needed to manually edit the layer. After we applied a median filter with a one-pixel radius and thresholded the skull segmentation, we manually edited each slice in every direction to add detail and smooth. Since the subject had a permanent retainer in her mouth that created all black pixels, we segmented the mouth as solid bone, which was not concerning because the EEG cap used did not cover the subject's mouth. We were able to provide a segmentation of the internal air, including the sinuses, esophagus, and ear canals, from the pseudo-CT image by thresholding the black pixels and then manually editing each slice. We also performed a quality check on both layers to ensure that there were no holes, layer overlap, and that the layers were at least two pixels thick.

The eyes, skin, and air layers were the least time consuming to segment. We segmented the eyes simply by thresholding the T_2 MRI. We segmented the skin layer by thresholding the entire head volume and removing all previous layers using a Boolean remove mask filter. We performed a quality check on the skin layer to ensure that it was at least two pixels thick. The important places for the quality check are the bridge of the nose, the bottom of the chin, and the sides of the head. Lastly, we segmented the air layer by thresholding the entire image and removing the solid

skin layer, followed by a check to ensure that the segmentation did not contain any holes between layers after they were removed. This final step is imperative to assure a quality mesh.

Table 2: Segmentation Time

Segmented Tissue	Amount of Work (hrs)
White Matter	40
Gray Matter	20
CSF	4
Skull and Sinus	35
Eyes, Scalp, & Air	8

2.4 Finite Element Mesh Generation

We used our full-head segmentation to generate realistic three-dimensional geometries for use in subsequent finite element simulations. We generated a smooth, linear, subject-specific, boundary-conforming, tetrahedral finite element mesh using Cleaver software²⁴ on a Late 2013 Mac Pro with a 2.7 Ghz 12 Core Intel Xeon E5 processor with 64 GB of RAM and an AMD FirePro graphics card. Cleaver is a multi material meshing package that produces structured meshes of tetrahedral elements with guaranteed minimum element angles, resulting in quality meshes that require fewer computational resources. We made a very high-resolution mesh without holes using the parameters listed in Table 3.

Table 3: Clever Settings (High Resolution)

scaling factor	0.6
size multiplier	1.0
lipschitz	0.2
padding	0
element sizing method	adaptive

Along with these parameters, indicator functions had to be input into Cleaver. We created indicator functions by calculating inverted distance maps of each layer in the full-head segmentation in Seg3D. To reduce the size of the mesh, we first generated a new mesh changing only the scaling factor parameter to 1.0 from the parameters in Table 3. We exported the computing sizing field from Cleaver and manipulated it in SCIRun by changing how quickly the elements increased in size. We input the changed sizing field back into Cleaver with the same indicator functions and successfully cleaved a new, smaller mesh without holes.

2.5 Mathematical Modeling

We used the head mesh, with associated inhomogeneous and anisotropic regions, as a volume conductor to solve the following boundary value problem:

$$\nabla \cdot \sigma \nabla \Phi = -I_V \quad \text{in } \Omega, \quad (1)$$

where Φ is the electrostatic potential, σ is the electrical conductivity tensor, and I_V is the current per unit volume defined within the solution domain, Ω . Equation (1) was solved for Φ with a known description of I_V and the Neumann boundary condition:

$$\sigma \nabla \Phi \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_T, \quad (2)$$

which says that the normal component of the electric field is zero on the surface interfacing with air (here denoted by Γ_T). The brain and surrounding tissue and skull were discretized using dipoles for the current source. We calculated the electrical field within the brain and then projected the field onto the surface of the scalp.²⁵

2.5.1 Electrical Conductivity Preparation

All electrical conductivities were homogeneous for each tissue with the exception of the white matter when using tensor data. The isotropic conductivities²⁶ that we used are shown in Table 4.

Table 4: Isotropic Tissue Conductivity

Tissue Type	Isotropic Conductivity (S/m)
White Matter	0.1429
Gray Matter	0.3333
Cerebrospinal Fluid (CSF)	1.79
Skull	0.001
Skin	0.4346
Sinus	1e-6
Eyes	0.5051

When we added the DTI tensor data, we used two approaches to convert the tensor data to conductivities. The first was scaling the data:²⁷

$$\sigma_{aniso} = \frac{\sigma_{iso}}{\sqrt[3]{d_1 d_2 d_3}} D \quad (3)$$

where D is the diffusion data, d_i is the i th eigenvalue of D , and σ_{iso} is the white matter isotropic conductivity. The second method gave the white matter a fixed ratio of conductivity:

$$\sigma_{aniso} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ W \end{bmatrix}, W = \begin{bmatrix} \sigma_{iso} \\ \frac{\sigma_{iso}}{10} \\ \frac{\sigma_{iso}}{10} \\ \frac{\sigma_{iso}}{10} \end{bmatrix} \quad (4)$$

where v_i is the i th eigenvector of D , W is the white matter ratio vector, and the ratio is 10 : 1.

2.5.2 Numerical Methods

We computed solutions to Equation 1 using finite element methods. By applying Green's divergence theorem to Equation 1, we generated the following weak formulation:

$$\int ((\bar{\sigma}_e + \bar{\sigma}_i) \nabla \phi_e) \cdot \nabla \psi(\bar{x}) d\bar{x} = - \int (\bar{\sigma}_i \nabla V_m) \cdot \nabla \psi(\bar{x}) d\bar{x}, \quad \forall \psi \in \Omega \quad (5)$$

where Ω is the linear, finite element mesh, and ψ represents the finite element basis functions characterized by local hat functions associated with mesh nodes. By applying this formulation to the finite dimensional mesh, we reduced Equation 5 to a system of linear equations:

$$A\phi_e = -RV_m \quad (6)$$

where A and R represent stiffness matrices defined by $A_{j,k} = \langle \nabla\psi_j, (\bar{\sigma}_e + \bar{\sigma}_i)\nabla\psi_k \rangle_\Omega$ and $R_{j,k} = \langle \nabla\psi_j, \bar{\sigma}_i\nabla\psi_k \rangle_\Omega$, and ϕ_e and V_m represent extracellular and transmembrane potentials, respectively.²⁸

We used SCIRun, the open-source problem-solving environment, to apply parameters and to solve Equation 6 numerically. Within the SCIRun environment, we applied isotropic and anisotropic conductivity tensors to the tetrahedral mesh as well as to inhomogeneous regions. We defined initial and boundary conditions and generated border regions in order to compute potentials by way of a conjugate gradient method with a Jacobi preconditioner.

2.6 Simulations and Visualizations

We ran all simulations and visualizations in the SCIRun problem-solving environment. All networks are shown in Figures 28 - 32.

2.6.1 Forward Problem

Solving systems from a known source to the EEG electrodes, described in Section 2.5, is known as a forward problem. The opposite action, solving systems from the EEG data to a known source, is an inverse problem. Inverse problems are commonly used for subjects with neurological disorders. In this project, we built SCIRun networks to solve forward problems with known sources and to write a lead field matrix for inverse problem networks in the future. We solved the forward problems with an isotropic and anisotropic system, which included DTI data.

The necessary inputs for the isotropic case were the tetrahedral head mesh, isotropic conductivities, the head segmentation, the physical electrode locations, and dipole sources. The physical electrode locations and dipole sources were part of the EEG dataset. We removed the fiducials from the electrodes, and the dipole sources were those chosen by the user. The dataset contained 4800 dipoles and 256 electrodes after the fiducials were removed. We registered the mesh to the head segmentation using a rigid registration. After we cut the flat tetrahedra out of the mesh, we mapped the conductivities to their respective tissues. We registered the electrodes and dipoles to the head mesh using the same transform. We solved the system with the mapped data and the chosen dipole sources. We then mapped the solution onto the mesh and the electrodes for visualization. We also included streamlines and isolines in the visualization.

For the anisotropic case, the network was largely the same with the exception of the scaled diffusion tensor dataset used as white matter conductivities. In addition, the DTI to mesh transform was needed as input. We registered the head mesh, electrodes, and dipoles to the DTI space with a rigid registration; the head segmentation was not needed.

2.6.2 fMRI

We visualized the fMRI data one step at a time using the two-dimensional matrix described in Section 2.2.4 as input and viewed one column at a time. We set each column onto a lattice volume,

rotated it 180 degrees, smoothed, thresholded, and clipped the volume for easier rigid registration. We registered the fMRI to the tetrahedral mesh using the bounding boxes and manual registration. After registration, we mapped the smoothed fMRI data onto the mesh using a mapping matrix with a linear interpolation basis.

2.6.3 EEG

We visualized the EEG data on the physical electrode locations, which we registered to the mesh space with a rigid registration after removing the fiducials from the electrode dataset. We used the filtered and cut EEG matrix as input and represented each time step as a column. We then placed the electrodes onto the mesh and mapped the EEG data onto the electrodes one time step at a time.

3 Results

All SCIRun networks used to generate results will be included in an open-source dataset for future use and editing.

3.1 Segmentation

For this project, we segmented an entire head into the eight detailed layers listed in Section 2.3. We used this mesh to create an inhomogeneous three-dimensional tetrahedral mesh.

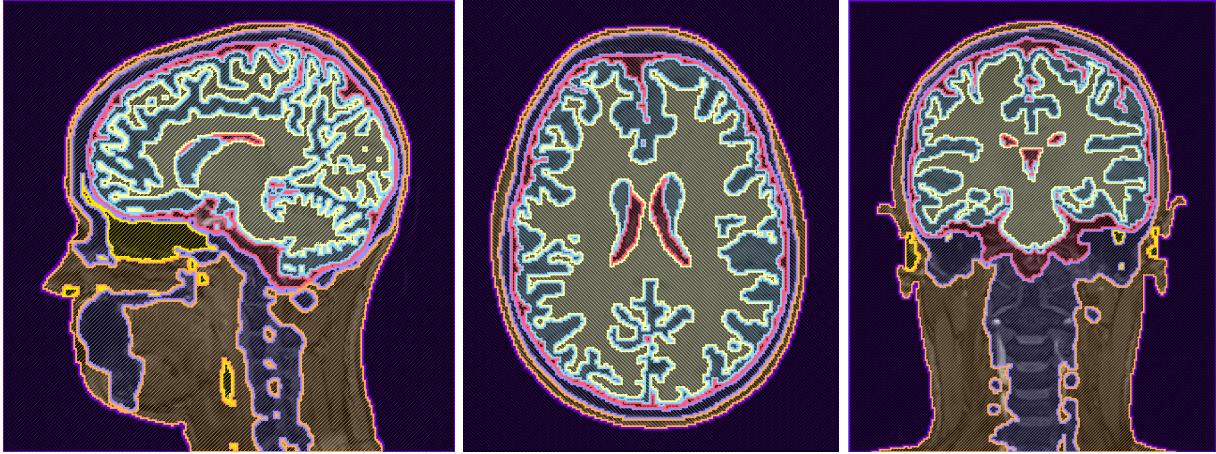


Figure 15: A high-resolution, eight-layer, full head segmentation made with Seg3D

Since the dataset did not include a CT scan, the task of segmenting the skull and the sinus layers seemed daunting. As described in Section 2.3, our first attempt at a bone segmentation was to combine a skull made from FSL²⁹ with bones thresholded using Seg3D. Although we had a bone segmentation, the sinus segmentation wasn't completed from this segmentation since it would have to be created manually.

The skull/bone and sinus segmentations we made from the pseudo-CT scan were smooth, connected segmentations that fit well into the entire head segmentation.

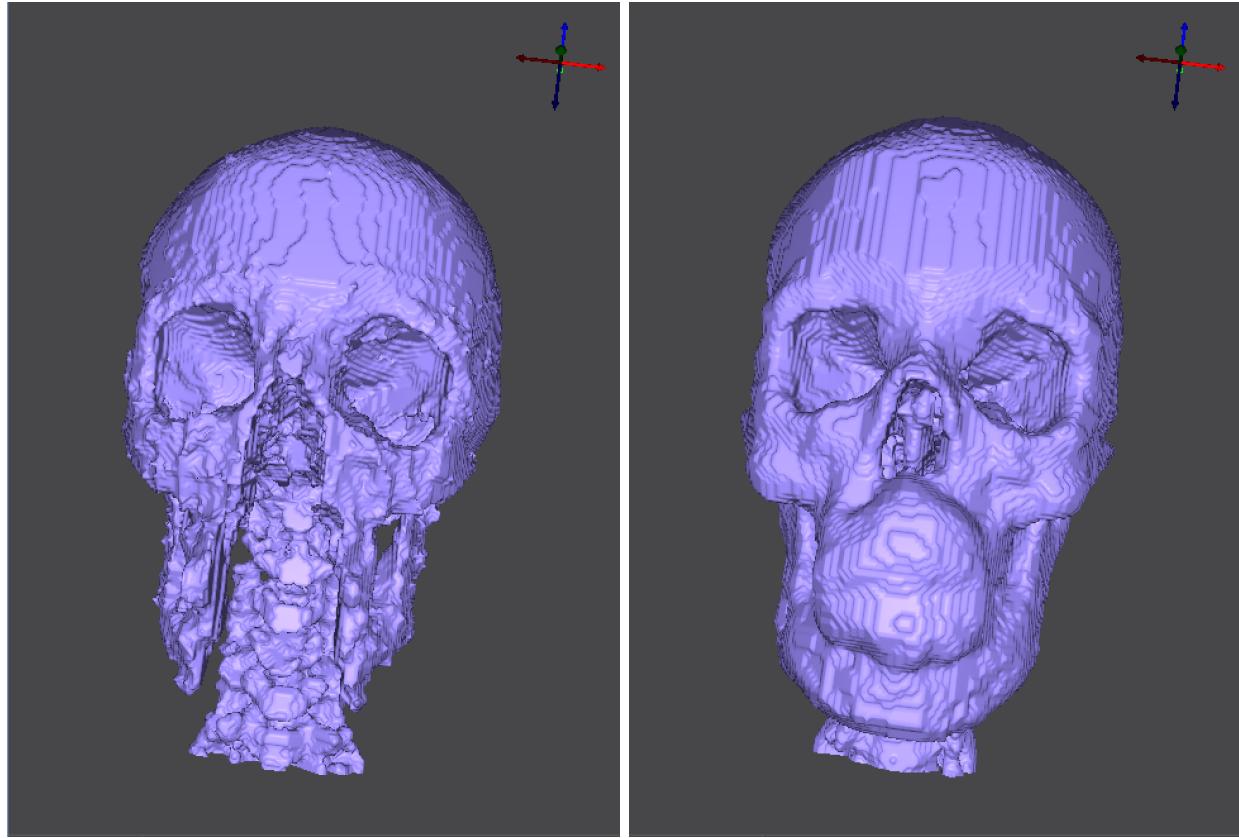


Figure 16: Skull segmentation comparison: Created with BET and thresholding (*left*) and with pseudo-CT (*right*). Both segmentations were made using Seg3D.

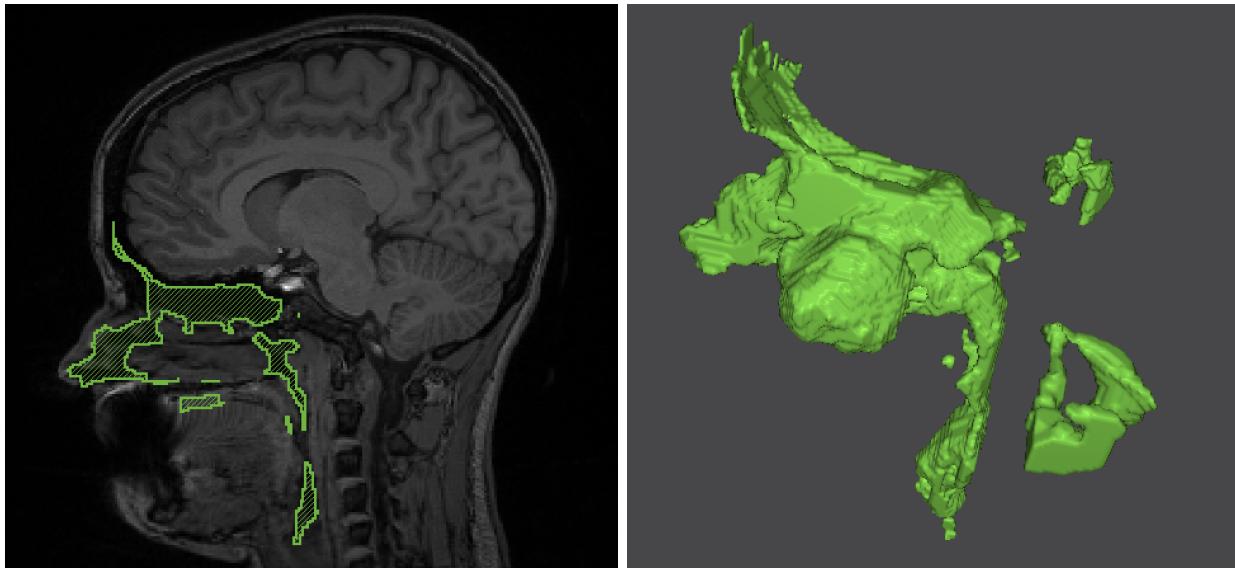


Figure 17: Sinus segmentation

During MRI imaging, the subject was on her back, which caused the brain to shift to the back of the head, resulting in thin segmented sections on the back of the head. There were also thin sections on the side of the subject's head, the bridge of the nose, and the bottom of the chin. We made these sections at least two pixels thick to ensure a mesh without holes.



Figure 18: Thin segmentation sections: side of the head (*left*), bridge of the nose (*middle*), bottom of the chin (*right*)

3.2 Finite Element Meshes

The highest resolution mesh we generated with the settings listed in Section 2.4 had 60.2 million elements and 10.3 million nodes. This mesh was large because of the complexity of the segmentation, including small features, thin sections, and several layers touching at once. The simulations performed slowly when using this mesh due to its size and required at least 32GB of RAM.

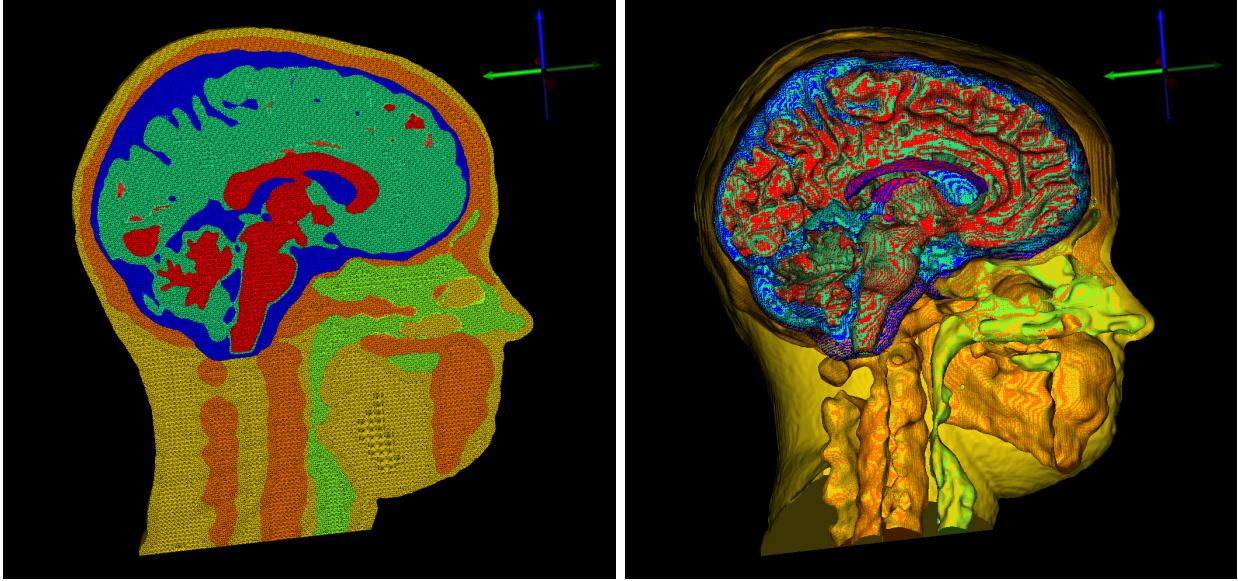


Figure 19: 60.2 M element mesh: tetrahedral mesh (*left*), surface mesh (*right*)

We attempted to generate smaller meshes to be able to run simulations more efficiently, but many of the meshes contained holes. After we manually changed the sizing field described in Section 2.4, we generated a mesh with 15.7 million elements and 2.7 million nodes that contained no holes. However, this mesh contained one flat tetrahedra, which we later removed in a SCIRun network. This issue is currently being investigated by Cleaver software developers.

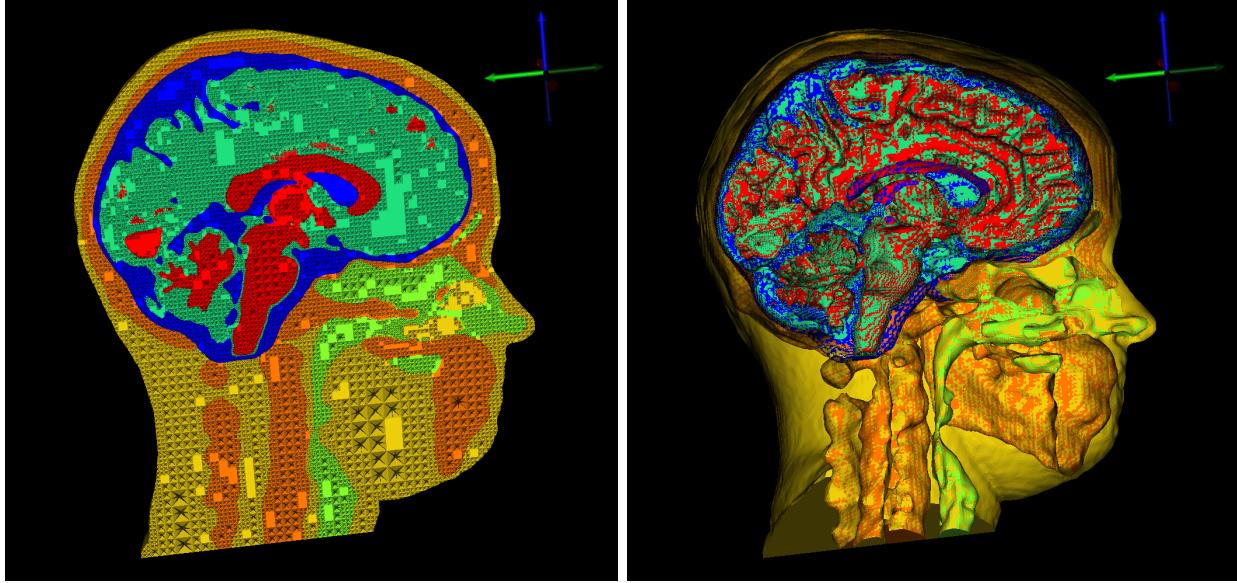


Figure 20: 15.7 M element mesh: tetrahedral mesh (*left*), surface mesh (*right*)

3.3 Forward Problem

3.3.1 Isotropic

An isotropic, inhomogeneous head model is expected to have spherical propagation of electrical signals. We generated three-dimensional streamlines as well as isolines to visualize this propagation and to compare isotropic and anisotropic conductivity. The simulations showed spherical propagation and acceptable registration of electrodes and dipoles to the mesh space.

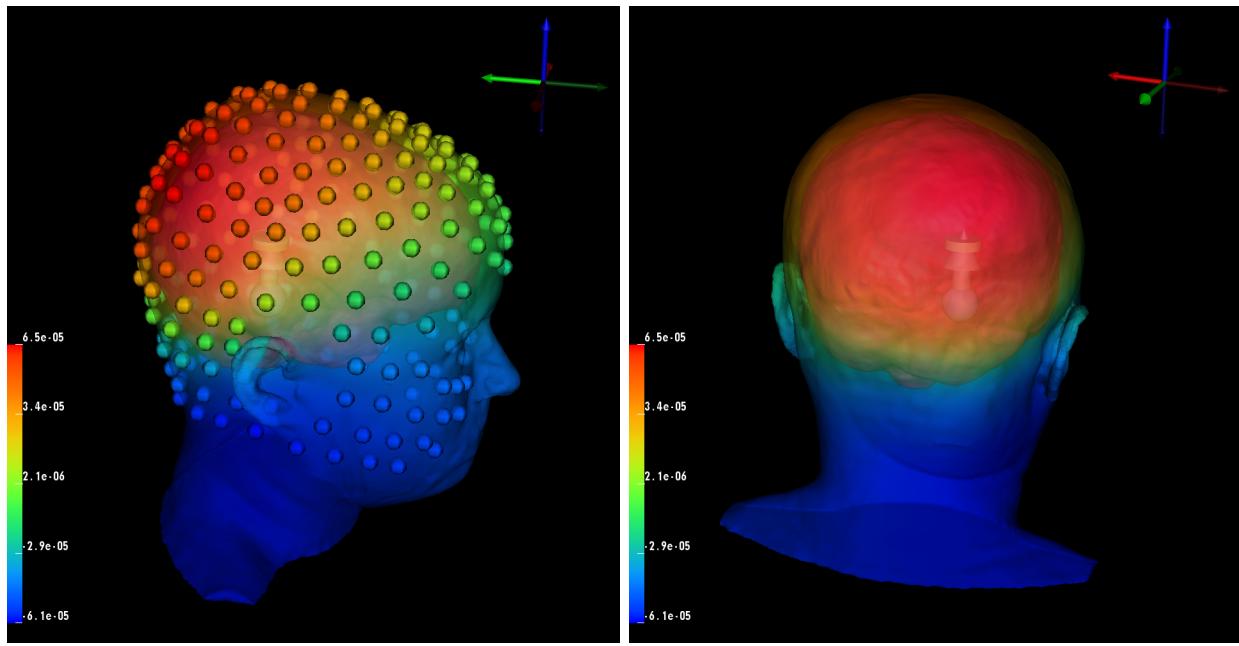


Figure 21: Isotropic forward problem solution with dipole source and data mapped onto the head surface and electrodes

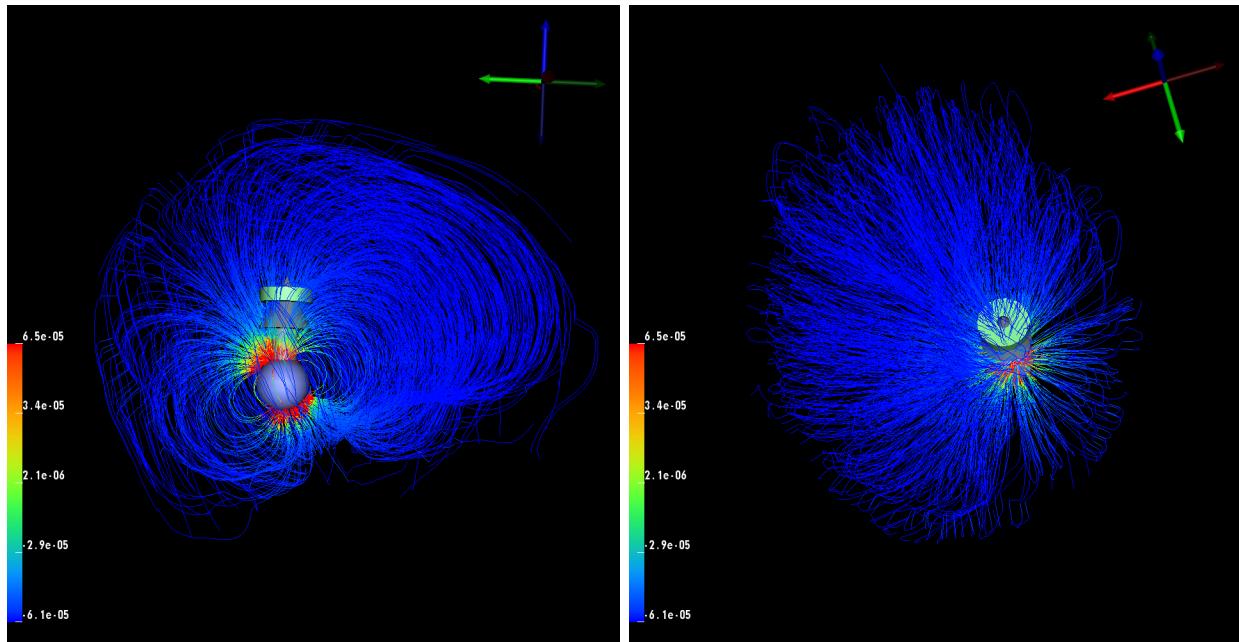


Figure 22: Isotropic streamlines with dipole source

3.3.2 Anisotropic

The expectation for an anisotropic, inhomogeneous head model is to have nonspherical propagation, which can be seen with the streamlines and isolines visualizations. As discussed in Section 2.5.1, two methods are used for scaling diffusion tensor data. We built the SCIRun network with the option to choose either scaling method. The simulations showed nonspherical propagation and acceptable registration of the electrodes, dipoles, and mesh into the diffusion tensor space.

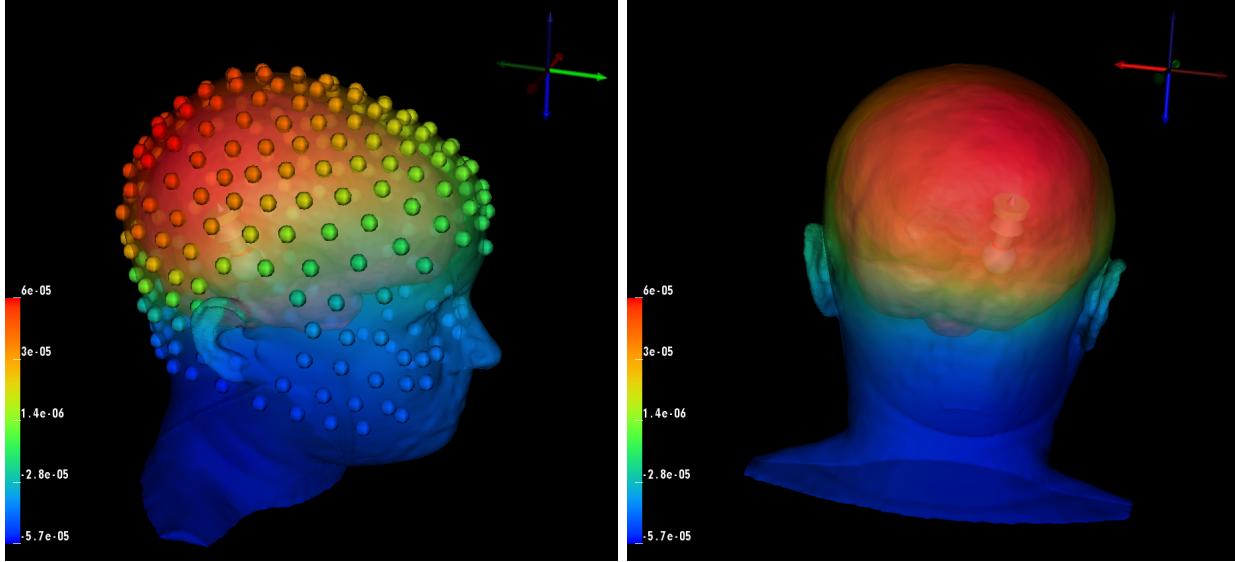


Figure 23: Anisotropic forward problem solution with dipole source and data mapped onto the head surface and electrodes

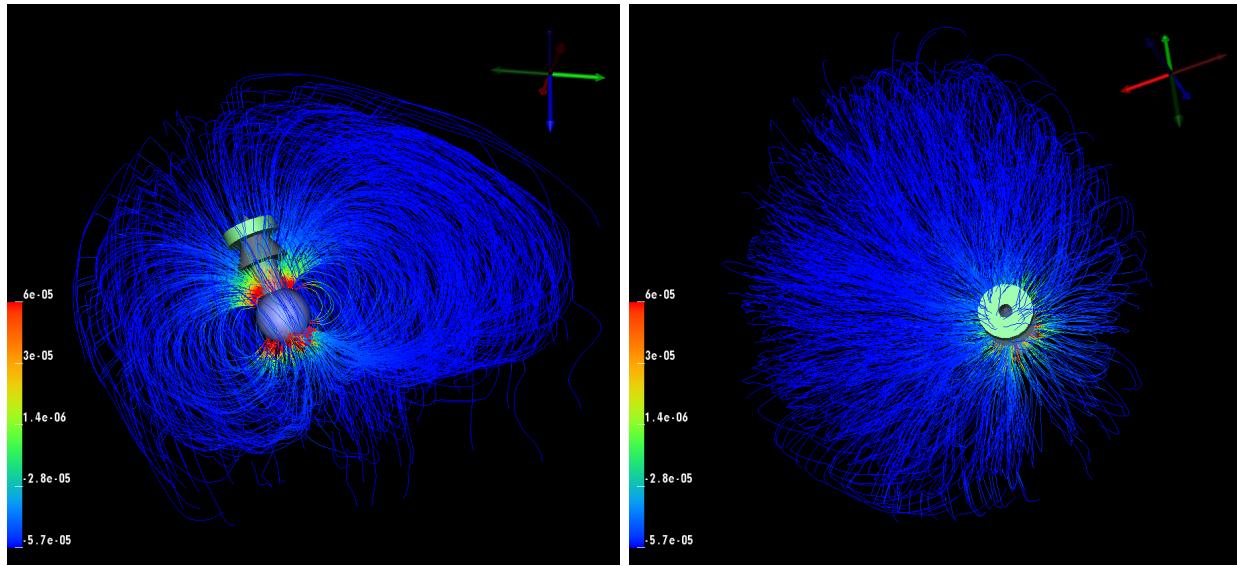


Figure 24: Anisotropic streamlines visualization with dipole source

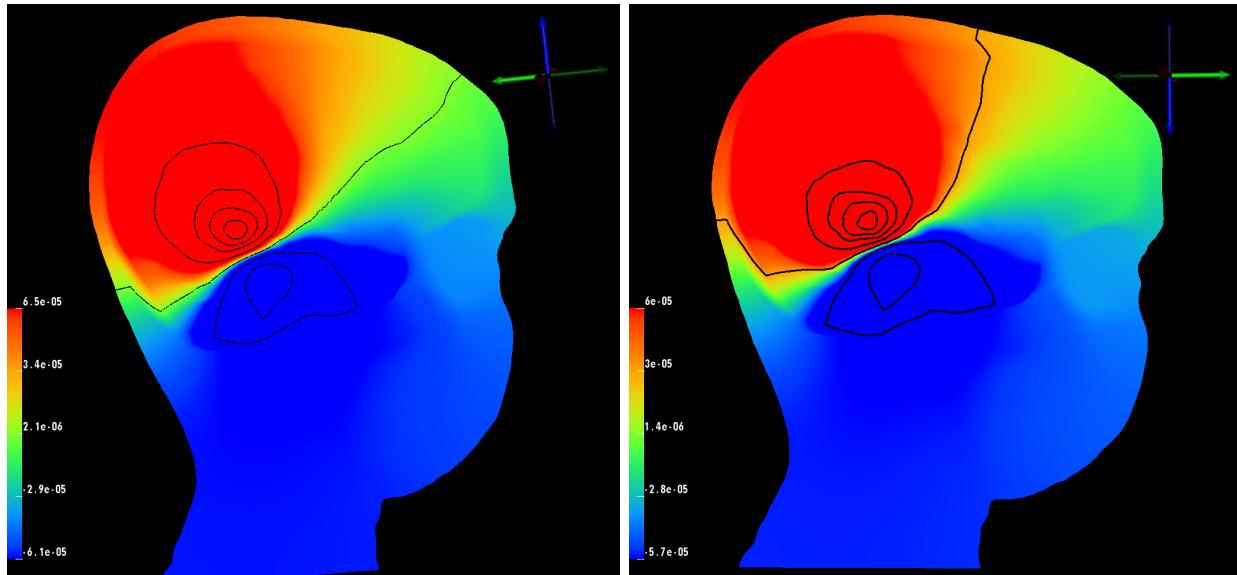


Figure 25: Isolines comparison: isotropic white matter conductivity (*left*), anisotropic white matter conductivity (*right*)

3.4 fMRI Visualization

fMRI data was a novel datatype for SCIRun. We successfully mapped and visualized the fMRI data onto the cortical surface with a rigid and manual registration to the mesh coordinate space.

This mapping network allows for future use of fMRI data in simulations using the SCIRun software package.

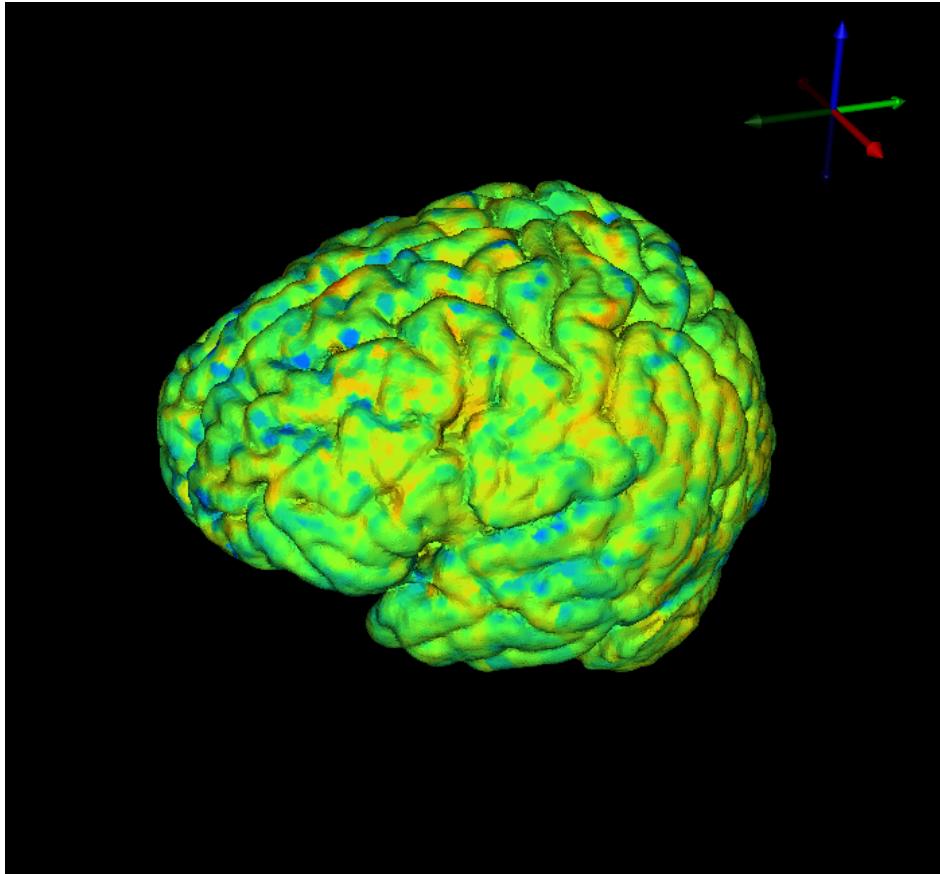


Figure 26: fMRI data mapped onto cortical surface mesh

3.5 EEG Visualization

When using EEG data, the particular application dictates if further processing, filtering, or cutting of the data is necessary. This EEG dataset contained “bad” electrodes, specifically around the eyes, which was possibly due to the blinking or rolling of the subject’s eyes. The bad leads will be fixed with further specific processing or trilinear interpolation.

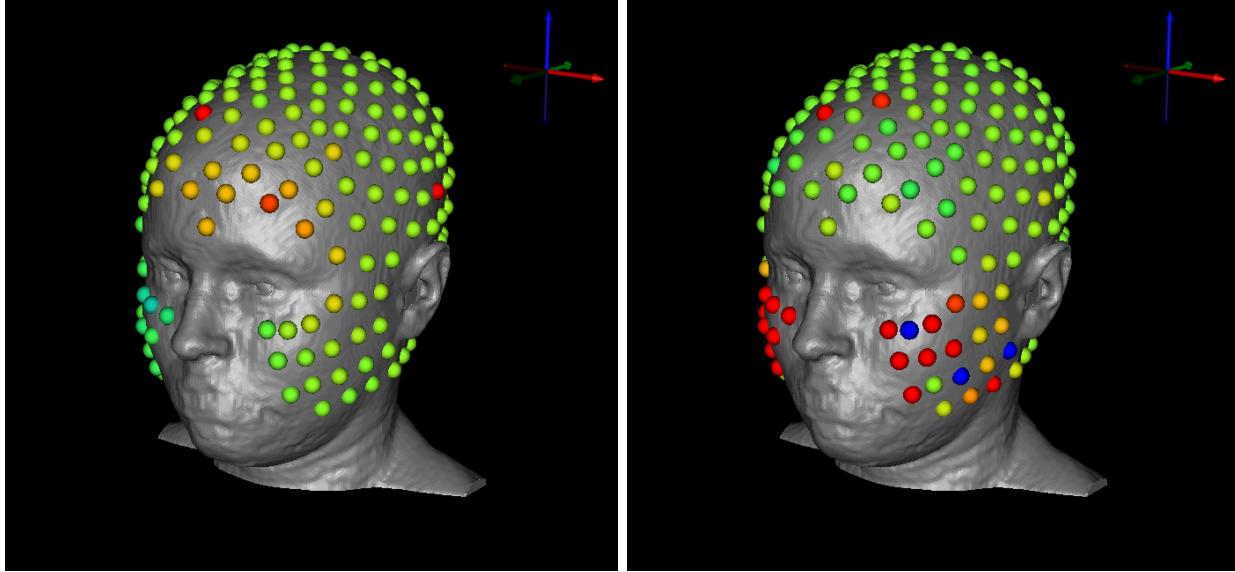


Figure 27: EEG signal visualization. Examples of “bad” electrodes that need further processing for specific applications

4 Conclusion

In this paper we have outlined a comprehensive pipeline to build an inhomogeneous, anisotropic head and brain model based on human data of all image modalities for use in electroencephalography with an emphasis on forward and inverse problem research as well as visualizations of functional MRI data and EEG data. Along with the pipeline, we have released the human data as open-source to enable other scientists to have a starting point for further research. Building a model can be time consuming and may hinder further important research. This model will allow scientists to have a straightforward path to building their own model and give them access to this model because it is the product of a high-resolution, multi-image dataset. The open-source project will be available in parts for those who want to use only specific aspects of the project.

Further investigations based on this pipeline include finding more appropriate decimation algorithms for three-dimensional tetrahedral finite element meshes to further reduce the mesh size; more exact sinus and skull segmentation methods to improve the appearance and accuracy of these layers; and more robust registration techniques, which will provide a better transformation matrix for moving images to DTI coordinate space, especially for fMRI data. Additions to this dataset could include more methods to incorporate functional MRI data into source localization simulations and more specific processing of EEG data for different applications, which will result in better visualizations.

5 Acknowledgements

Imaging and electroencephalography for this project were done at Electrical Geodesics Inc. at the University of Oregon. This work could not have been completed without the help of Jess Tate and Brett Burton. This project was supported by the National Institute of General Medical Sciences of the National Institutes of Health under grant number P41 GM103545-18.

References

- [1] R. Brette and A. Destexhe. *Handbook of Neural Activity Measurement*. Cambridge University Press, 2012.
- [2] SCI Institute, 2016. SCIRun: A Scientific Computing Problem Solving Environment, Scientific Computing and Imaging Institute (SCI), Download from: <http://www.scirun.org>.
- [3] J.P. Mugler and J.R. Brookeman. Three-dimensional magnetization-prepared rapid gradient-echo imaging (3d mp rage). *Magn. Reson. Med.*, 15(1):152–157, July 1990.
- [4] M.P. Lichy, B.M. Wietek, J.P. Mugler, W. Horger, M.I. Menzel, A. Anastasiadis, K. Siegmann, T. Niemeyer, A. Konigsrainer, B. Kiefer, F. Schink, C.D. Claussen, and H.P. Schlemmer. Magnetic resonance imaging of the body trunk using a single-slab, three-dimensional, t2-weighted turbo-spin-echo sequence with high sampling efficiency (space) for high spatial resolution imaqing: initial clinical experiences. *Invest. Radiol.*, 40(12):754–760, December 2005.
- [5] T.G. Reese, O. Heid, R.M. Weisskoff, and V.J. Wedeen. Reduction of eddy-current-induced distortion in diffusion mri using a twice-refocused spin echo. *Magn. Reson. Med.*, 49(1):177–182, 2003.
- [6] J. Juntu, J. Sijbers, D.V. Dyck, and J. Gielen. Bias field correction for mri images. In: Kurzyński M., Puchała E., Woźniak M., żołnierk A. (eds) *Computer Recognition Systems. Advances in Soft Computing*, vol 30. Springer, Berlin, Heidelberg, (543-551), 2005.
- [7] Y. Zang, M. Brady, and S. Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE Trans Med Imag*, 20(1):45–57, 2001.
- [8] J.L.R. Anderson, S. Skare, and J. Ashburner. How to correct susceptibility distortions in spin-echo echo-planar images; applications to diffusion tensor imaging. *NeuroImage*, 20(2):870–888, 2003.
- [9] 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.
- [10] X. Li, P.S. Morgan, J. Ashburner, J.C. Smith, and C. Rorden. The first step for neuroimaging data analysis: Dicom to nifti conversion. *Journal of Neuroscience Methods*, 264:47–56, 2016.
- [11] FSL. Fsl dtifit. <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FDT/UserGuide> Section 2.4.

- [12] P.A. Yushkevich, J. Piven, H.C. Hazlett, R.G. Smith, S. Ho, J.C. Gee, and G. Gerig. User-guided 3d active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *NeuroImage*, 31(3):1116–28, July 2006.
- [13] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J-C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F.M. Fennessy, M. Sonka, J. Buatti, S.R. Aylward, J.V. Miller, S. Pieper, and R. Kikinis. 3d slicer as an image computing platform for the quantitative imaging network. *Magn. Reson. Imaging*, 30(9):1323–31, November 2012.
- [14] NITRC. 1000 functional connectomes project. http://fcon_1000.projects.nitrc.org/.
- [15] Jimmy Shen. Tools for nifti and analyze image, Jan 2014. <https://www.mathworks.com/matlabcentral/fileexchange/8797-tools-for-nifti-and-analyze-image>.
- [16] U. Aydin, J. Vorwerk, P. Kupper, M. Heers, H. Kugel, A. Galka, L. Hamid, J. Wellmer, C. Kellinghaus, S. Rampp, and C.H. Wolter. Combining eeg and meg for the reconstruction of epileptic activity using a calibrated realistic volume conductor model. *PLOS ONE*, 9(3), March 2014.
- [17] Brett Shoelson. edfread, September 2016. <https://www.mathworks.com/matlabcentral/fileexchange/31900-edfread>.
- [18] CIBC, 2016. Seg3D: Volumetric Image Segmentation and Visualization. Scientific Computing and Imaging Institute (SCI), Download from: <http://www.seg3d.org>.
- [19] S.M. Smith. Fast robust automated brain extraction. *Human Brain Mapping*, 17(3):143–155, November 2002.
- [20] A.M. Dale, B. Fischl, and M.I. Sereno. Cortical surface-based analysis. i. segmentation and surface reconstruction. *NeuroImage*, 9:179–194, 1999.
- [21] Wellcome Trust Centre for Neuroimaging. Statistical parametric mapping, May 2016. <https://www.mathworks.com/matlabcentral/fileexchange/56963-statistical-parametric-mapping>.
- [22] Koen Van Leemput, Frederik Maes, Dirk Vandermeulen, and Paul Suetens. Automated model-based tissue classification of mr images of the brain. *IEEE transactions on medical imaging*, 18(10):897–908, 1999.
- [23] A. Torrado-Carvajal, J.L. Herranz, E. Alcain, A.S. Montemayor, L. Garcia-Cañamaque, J.A. Hernandez-Tamames, and ... & N. Malpica. Fast patch-based pseudo-ct synthesis from t1-weighted mr images for pet/mr attenuation correction in brain studies. *Journal of Nuclear Medicine*, 57(1):136–143, 2016.
- [24] CIBC, 2016. Cleaver: A MultiMaterial Tetrahedral Meshing Library and Application. Scientific Computing and Imaging Institute (SCI), Download from: <http://www.sci.utah.edu/cibc/software.html>.
- [25] C.R. Johnson. *Computational Methods and Software for Bioelectric Field Problems*, volume 1. CRC Press, 4th edition, 2015.

- [26] M. Rullmann, A. Anwander, M. Dannhauer, S.K. Warfield, F.H. Duffy, and C.H. Wolters. Eeg source analysis of epileptiform activity using a 1 mm anisotropic hexahedra finite element head model. *NeuroImage*, 44:399–410, 2009.
- [27] D. Gullmar, J. Haueisen, and J. R. Reichenbach. Influence of anisotropic electrical conductivity in white matter tissue on the eeg/meg forward and inverse solution. a high-resolution whole head simulation study. *NeuroImage*, 51:145–163, 2010.
- [28] D. Wang, R.M. Kirby, R.S. MacLeod, and C.R. Johnson. Inverse electrocardiographic source localization of ischemia: An optimization framework and finite element soltuion. *Journal of Computational Science*, 2013.
- [29] M. Jenkinson, M. Pechaud, and S. Smith. Bet2: Mr-based estimation of brain, skull and scalp surfaces. *n Eleventh Annual Meeting of the Organization for Human Brain Mapping*, 2005.

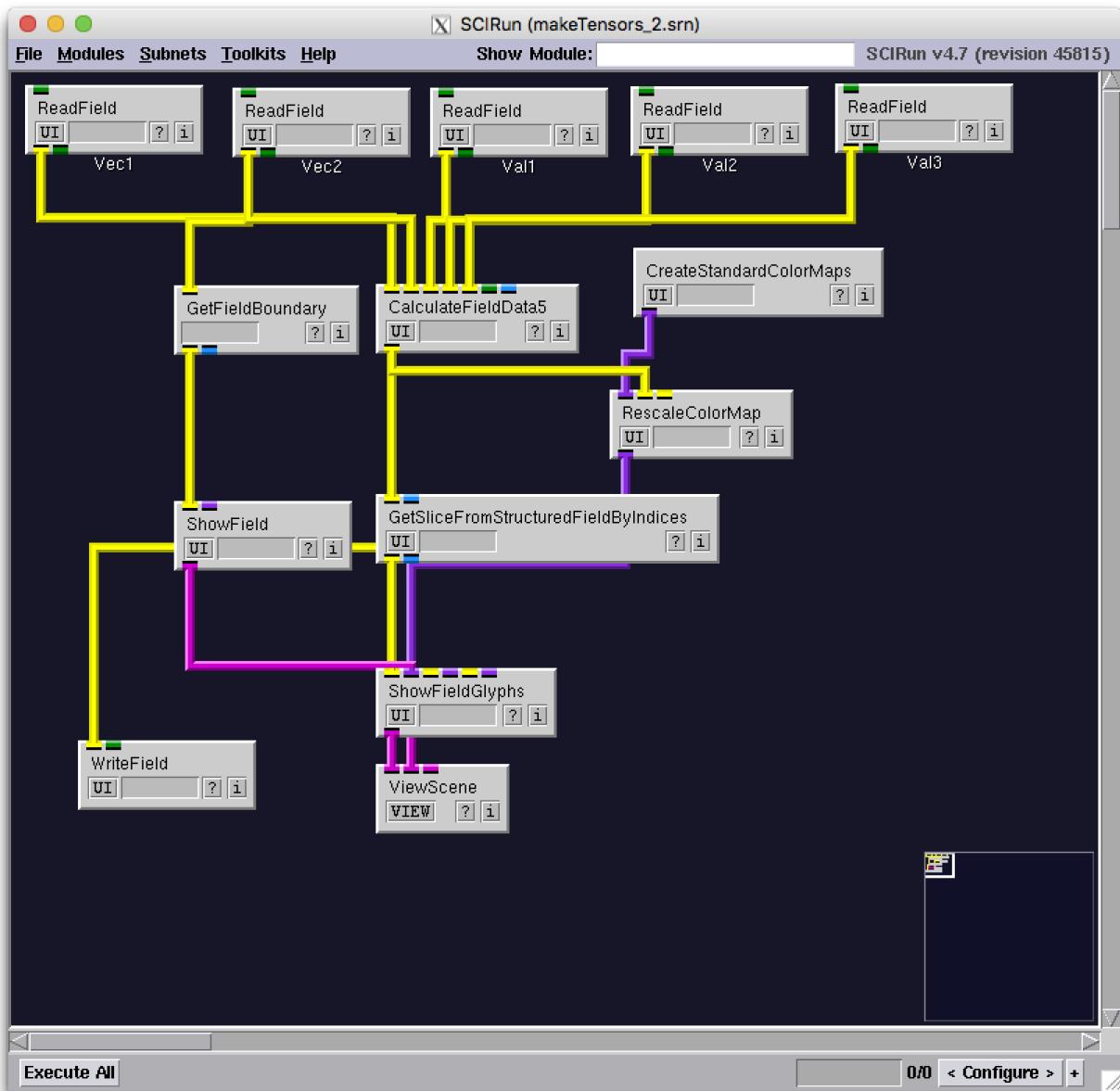


Figure 28: SCIRun network to build diffusion tensor dataset

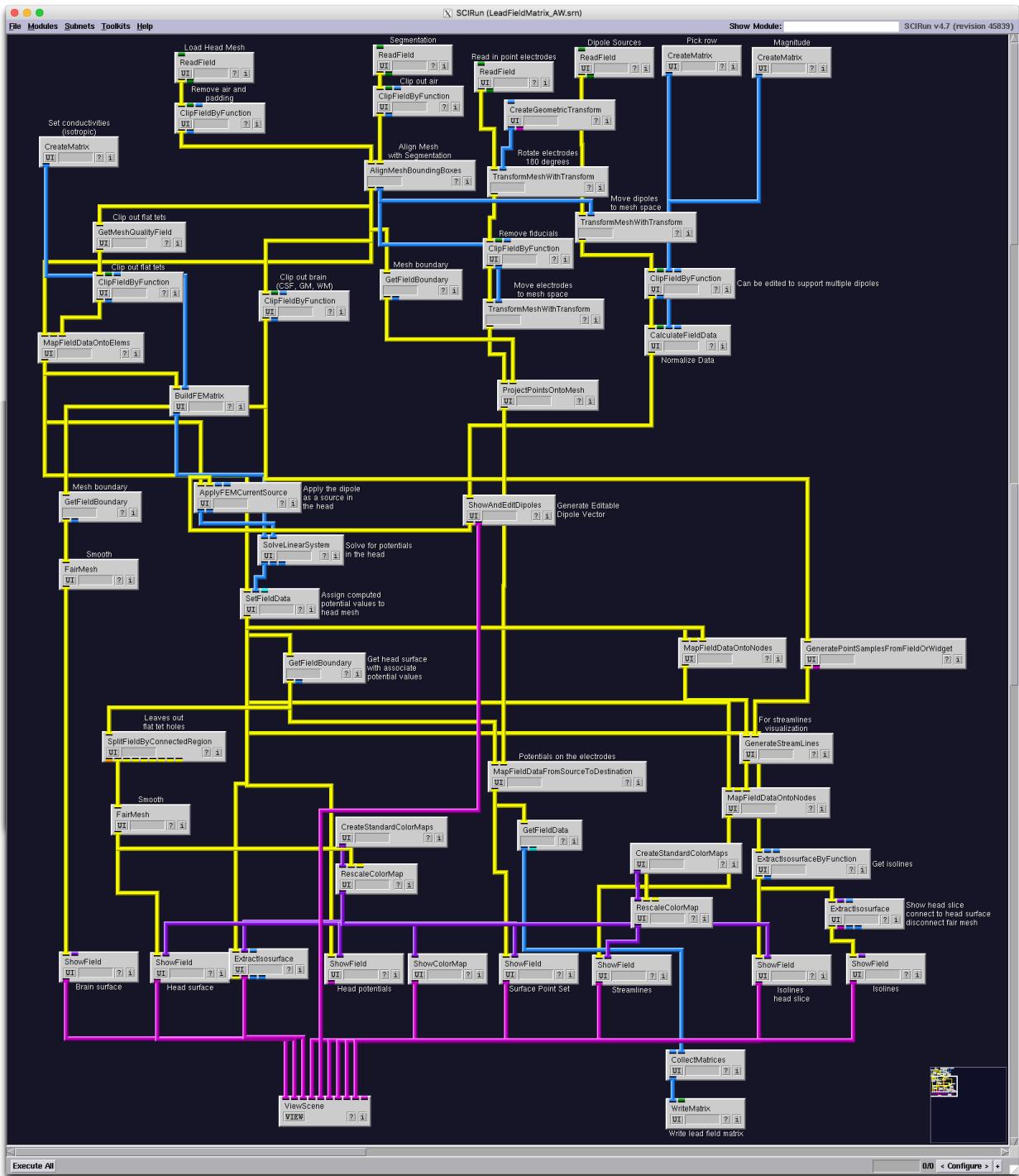


Figure 29: SCIRun network for isotropic forward problem

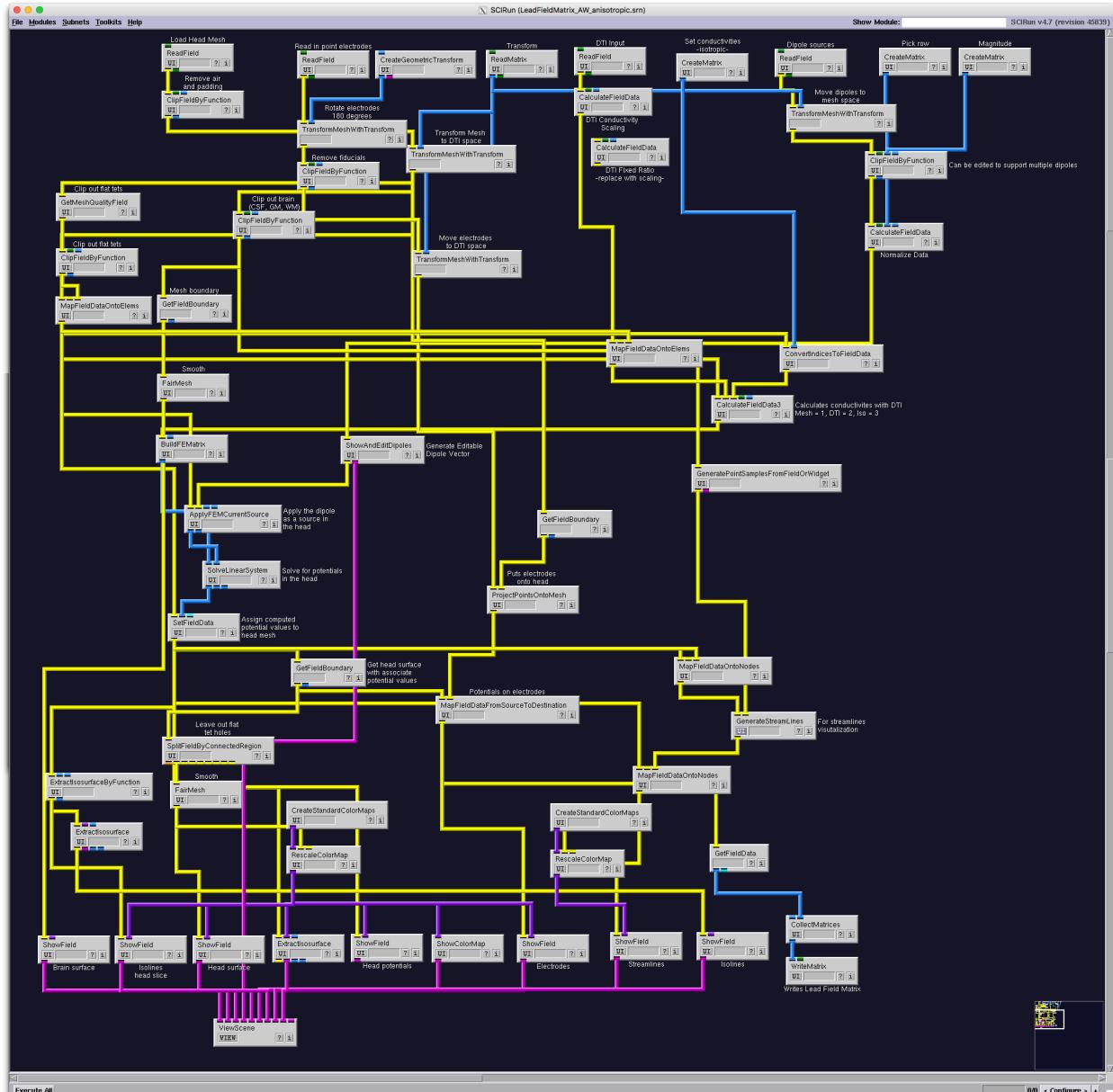


Figure 30: SCIRun network for anisotropic forward problem

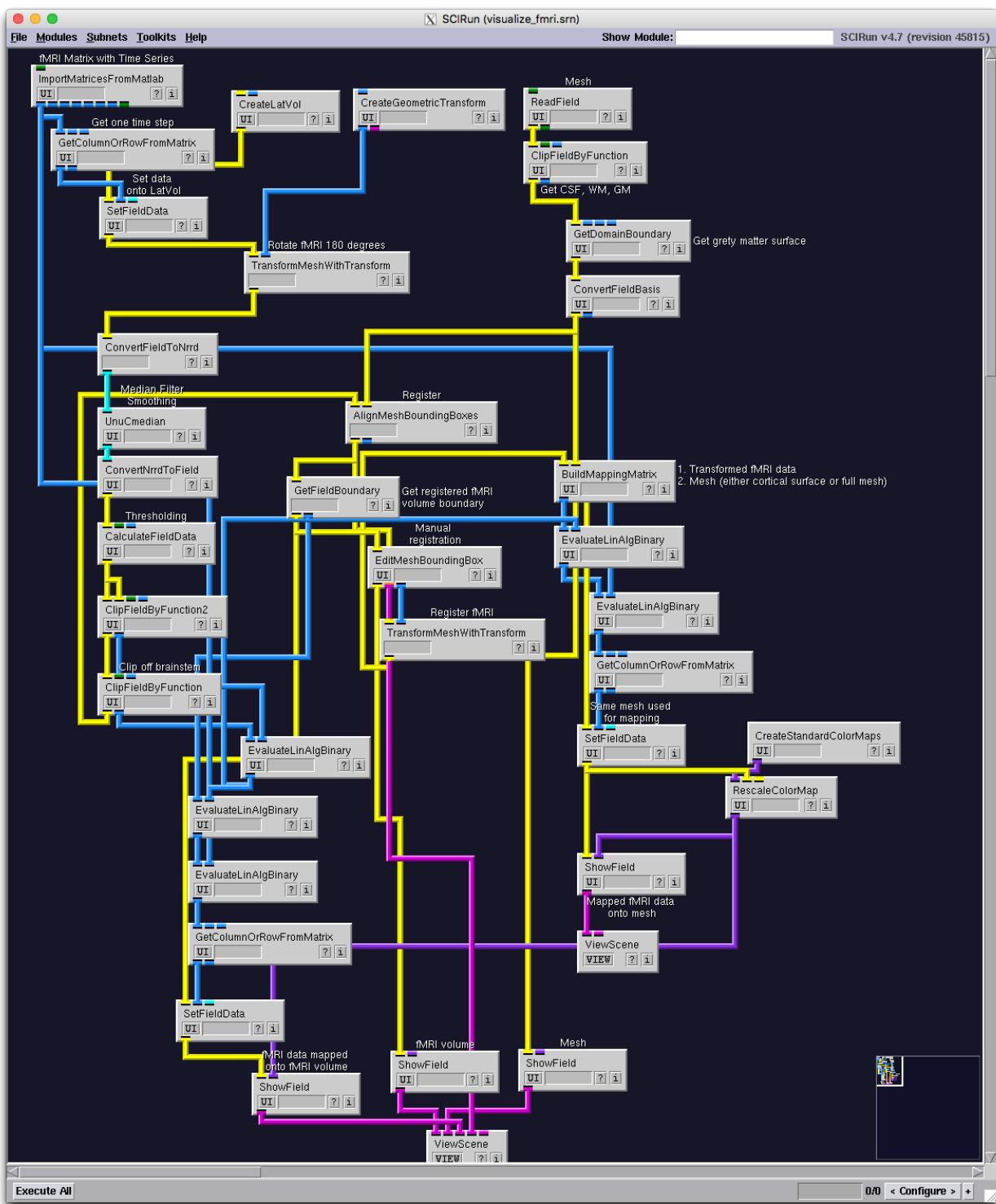


Figure 31: SCIRun network for fMRI visualization

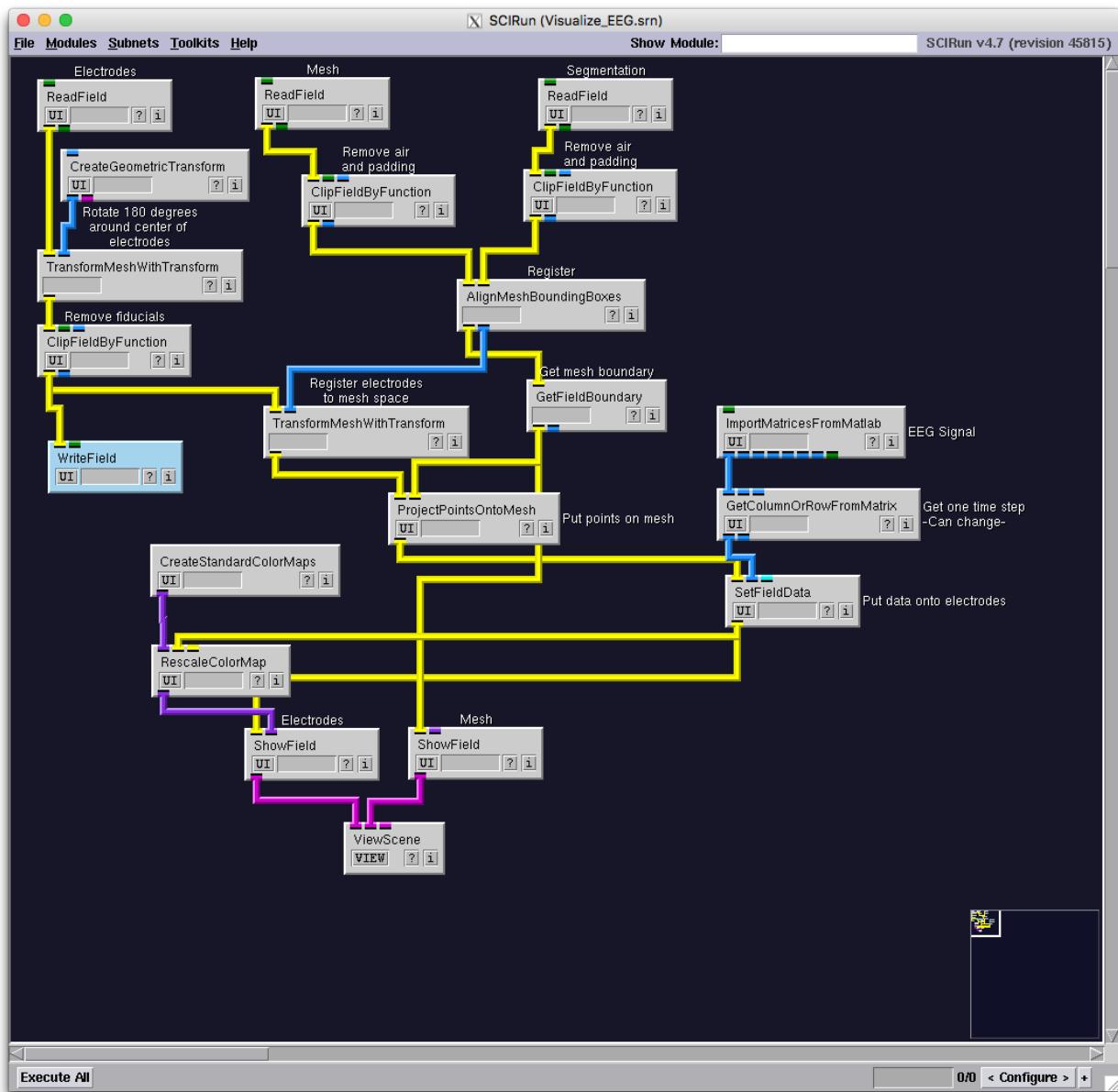


Figure 32: SCIRun network for EEG visualization