Automatic Brain Segmentation for 3D Printing

Phase 2 Report

Due: December 7, 2021

Achira Fernandopulle
Bailey Gano
Leo Musacchia
Nicolas Re
Nathan Carpenter

I pledge my honor that I have abided by the Stevens Honor System

Abstract

The project objective is to design and prototype an automatic process that converts MRI scans into 3D models that can be 3D printed into physical brain models. This report summarizes Phase 3, which includes the revised project statement, finalized technical analysis, engineering design discussion, and alpha prototype description/demonstration. The team created a concrete project pipeline that creates a 3D brain model that meets customer needs and product specifications. The following report explains the pipeline process and alpha prototyping progress of the product. Included is the project plan, indicating the future goals and path that the team will look towards in the remaining three phases.

Table of Contents

Project Statement

Project Overview

One of, if not the most complex and least understood organs in the human body is the brain. This applies to both the doctors who must analyze imaging, diagnose issues, and, sometimes, perform surgery, and the patients who house the brains in question. Such complexity can form a disconnect of understanding between people who have gone through years and years of schooling on this subject and people who have not. The objective of this project is to bridge the gap formed between professional and patient through the use of a brain model personalized to each patient. This model will be an accurate representation to scale of the patient's brain that can be used by a healthcare professional to describe diagnoses, procedures, and any other possible areas of concern in a much more understandable and interactive manner. Another main objective of the prints are to serve as souvenirs given to patients by a practice or made available to people who have access to their scans and want a print of their brain. The pipeline that will be used to produce such a model goes as follows: take the MRI scan of the patient, run it through a segmentation software to form the model, clean up any possible holes or defects that may have occurred, convert that file into one that can be 3D printed, and then print it. By the end of this academic year, the team would like this process to be fully automated and they have made great progress towards achieving this goal throughout Phase 3.

Needs and Specifications

Very little has changed on the front of the needs and specifications of this project. Above all else, the accuracy of the model is the main focus of the group's efforts at this point. In order for the models to be effective in their purpose, i.e. a tool to be used by doctors in order to better explain to their patients all of the intricacies of the brain, they must be as anatomically accurate as possible. Other parameters, such as cost, durability, aesthetics, etc., are also being taken into consideration in order of their relative importance. To better visualize how this group prioritized these parameters, a simple Customer Needs Chart (pictured below) with a numbered scale ranging from 1 to 5, 1 being the least important and 5 being the most important was constructed.

Index	Category	Need	Priority (1-5)
C1	Cost	Model is relatively inexpensive to make	5
D1	Design	Model is realistically sized	3
D2	Design	Model is durable	4
D3	Design	Model is aesthetically pleasing	2
D4	Design	Model is resistant to long term exposure to air	3
P1	Performance	Model is accurate enough to serve as general visual	5
S1	Safety	Model painted with non-toxic material	2

Table 1: Customer Needs Chart

Concept Selection

Since this project is focussed on creating the process of 3D printing a brain model as opposed to designing a physical product, the concept generation is slightly abnormal. This group plans to use primarily open source software with intermittent quality checks to fully automate the entire process of going from an MRI image to a physical model. In order to do this, a general pipeline was created in Phase 1 to outline all of the steps that need to be taken in the process that is being developed. This pipeline will be further analyzed in the *Engineering Analysis* section of this report, but there was some progress made on this front that pertains to the concept selection, primarily the programs that will be used to carry out certain tasks.

The first step in the brain modeling pipeline is to get the MRI images in the proper file format so as to be compatible with the rest of the programs being used. An MRI scan can come in many different formats, but the two main ones are a Nifti file and a DICOM series. The primary difference between these two file types is that a Nifti file is saved as a 3-dimensional image while a DICOM file contains a stack of 2-dimensional images. For this application, the group will be using the MRI images in a DICOM series because it allows for a more detailed segmentation which will in turn create a more accurate model.

Next, the MRI scans have to be uploaded into a segmentation software in order to get rid of any unwanted matter, such as the skull, cerebrospinal fluid, the spinal cord, etc., and clean around the outer surface of the brain. The software that will be used for this part of the process is called FreeSurfer. After the brain has been segmented, it can be placed into a meshing software that will smoothen out the surface and fill in any holes that may have occurred while transitioning between steps. This will be done using a program called Meshlab. After the file goes through the reconstruction and smoothening step, it must then run through a quality certification step to ensure there will be no errors in the surface during the printing process. And finally, the file can be saved into an STL format and sent off to be 3D printed.

Technical Analysis

Model Material and Printer

With the decision to print the brain model using two materials, the team began technical analysis with the material selection. The material for a brain model fitting the team's project statement needs to be durable, relatively stiff and strong, and relatively inexpensive . The team looked at various materials, from gelatin to PLA. After various research on material properties, the team decided to print the brain models using PLA (polylactic acid) filament, as shown in the figure below. Specifically, the gray matter (outer mass of brain) would be printed in some clear/translucent PLA and the white matter (inner mass of brain) being printed in a colored or black PLA filament. PLA filament has a yield strength of 60 MPa, tensile modulus of 3600 MPa, and flexibility strength of 83 MPa. This means that this material is durable, strong, and resistant to long term exposure to air and its surroundings, which fits the customer needs and specifications for this product (needing to be strong and durable).

Figure 1: PLA Filament

Moving forward with this material selection, the team next researched different 3D printing methods and printers that were available to them in the Stevens Proof Lab. The team chose to print the model using polyjet printing, as it creates smooth and detailed prints, can achieve complex shapes (a key component for printing complex models such as a human brain), and has a variety of colors and materials available to print with. With this decision of polyjet printing the team chose to use the Objet 350 Connex 3 printer from the Stevens Proof Lab, as shown in the figure below. This printer has a build volume of 340 x 340 x 200 mm^3, a layer resolution of 0.016mm or 0.03mm, and 5 material families and 140 digital materials to print with. Additionally, this printer has polyjet support materials such as SUP705 (a gel-like photopolymer) and SUP706 (a soluble support), which allow for a clean support removal. Using this printer and it's available support materials will allow the team to make a product that is more accurate and aesthetically pleasing, meeting some of the customer needs and specifications listed in the previous section and allowing the product to become one step closer to meeting the project

statement. Additionally, this printer is more than large enough to be able to print multiple brain models at once, which helps aid the project pipeline in being more time efficient.

Figure 2: Objet 350 Connex 3

Cost Analysis

With the software, material, and printer selected, the team was able to perform a cost analysis to determine if their pipeline fit the specification of being relatively inexpensive. The team found that a 2 kg spool of PLA filament costs around 50 dollars, which equates to about 2.5 cents per gram. Because the model will be made using two materials, two spools will be needed, which makes the cost of filament 100 dollars. PLA has a density of around 1.25 g/cm^3 and an adult human brain has a volume of 1260 cm^3. Then using $m = \rho V$, the mass of filament used is found to be around 1575 g, or around 40 dollars of filament. For prototyping, the team will scale the model down for printing to save filament material and reduce printing time. Freesurfer and Meshlab are free software, however, MATLAB requires a license to use. While the team is able to utilize the software for free as Stevens students, this will become unviable in the near future when the team graduates. Therefore, the cost of a license will be considered. A perpetual license (indefinite license) from MATLAB costs around 2150 dollars. Lastly, the printer cost needs to be considered. As with the situation with MATLAB, the team is able to utilize these printers for free using the Proof Lab, but will not be able to in the future. However, the prices are not officially listed, as they are subject to the manufacturer and seller. Additionally, the team would most likely utilize a cheaper printer once they were no longer students. For numerics sake, the cost of dual extruder 3D printers will be considered. The price of these printers range from 800 to 6000 dollars, so an average price of around 3000 dollars will be assumed. With all this in mind, the cost of printing 3D models comes out to be around 40 dollars, with a one time fee of 5150 dollars. The table below helps to visualize and compare the two scenarios of printing while the team are students versus when they are not. This printing cost is extremely reasonable compared to what competitors are selling. For personalized, full-scale brain models, competitors are selling at around 250 dollars per model. The team's model could either be sold much cheaper to

persuade customers, or at a similar but slightly lower price, which would help the team quickly pay off the license and printer fees (assuming they are not students at this time).

	As Students	**No Longer Students**
Filament Cost (per model)	$40	$40
License Cost	$0	$2150
Printer Cost	$0	$3000
Total	$40	$5190 ($5150 one time fee)

Table 2: Cost Analysis for Printing out a Brain Model

Input Gathering - MRI Scans

First in the project pipeline is gathering brain MRI scans to use as a basis for creating the 3D brain model. Analysis on MRI resolutions, scan time, and slice thicknesses need to be done in order to analyze and segment the scans. Medical images (MRI scans), are stacks of 2D images that contain in-plane and out-of-plane resolutions, as shown in the figure below. The team found that MRI scanners typically take 20-60 minutes to complete, and have magnetic field strengths of 1.5T or 3T, reaching resolutions of 1.5 mm x 1.5 mm x 4 mm. Typically, scans are isentropic (equal dimensions in all directions), with dimensions of 3 mm x 3 mm x 3 mm. Additionally, slice thicknesses were found to typically be >2mm.With these dimensions and resolutions found, the team aimed to use MRI scans that are typically around those values, as this will help to meet the project statement of creating an accurate model of the brain. MRI scans with lower resolutions as those stated above will be less accurate, which will in turn cause less accurate segmentation and a less accurate printed model.

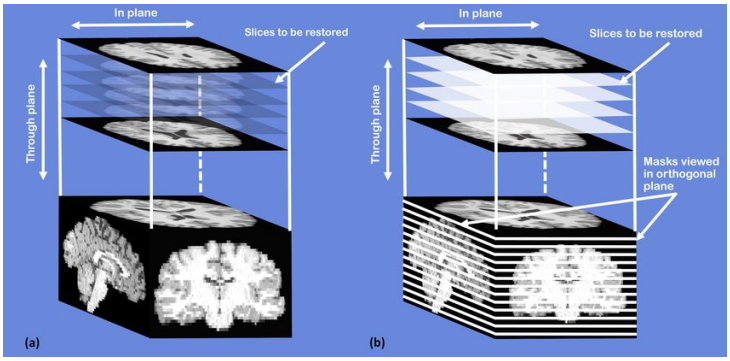


Figure 3: Stack of MRI scans and their dimensions (resolutions)

Engineering Design

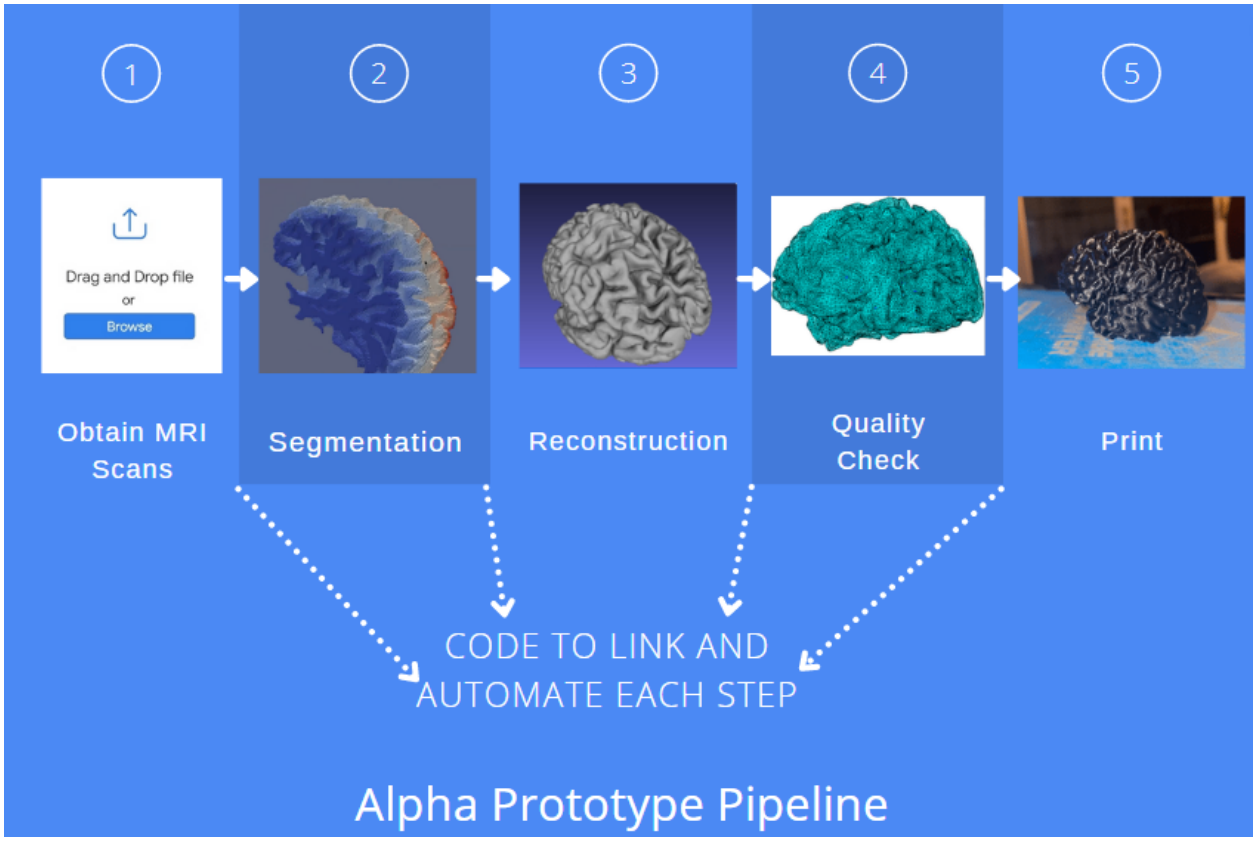


Figure 4: Updated Product Pipeline

The design of the pipeline to accomplish a successful automated process has not changed much since phase 2. Pictured above in figure 4 is an updated version of the overall process design of the group's product. Thus far, the group wanted to demonstrate feasibility and proficiency within the more difficult component steps which were the segmentation, reconstruction, quality check, and print steps. Within the segmentation step the group inputted an MRI scan to see how well the software can segment the slices into a rough model. In the reconstruction step, the group learned how meshlab functions and successfully demonstrated how the surface of the model can be refined and smoothed. In the quality check step, the group was able to put together code in MATLAB that evaluates the surface mesh quality and checks for any points that may cause printing errors. In the print step, the group was able to print a sample model of a brain MRI that had been run through each of these steps. With the successful execution of these highlighted components, the group now knows where the design needs to head towards.

One of the next important tasks is to identify what the user interface will look like for someone who is uploading an MRI as an input for the product and how that will be done. The interface might occur on a website or within an app. The next most important aspect is to now

design code that will link each of the steps together and run the file throughout each step of the pipeline to obtain the fully automated process. Each step that has been tested will be elaborated on in the alpha prototype section.

Design Impacts

When developing a product such as this one, there are impacts across multiple domains that should be considered. The first of which is the ethical impacts of this product. As previously mentioned, this product has the potential for multiple types of uses. One use is that a clinic obtains the brain print to use as a tool for explanation or to provide their neuro-patients with a souvenir. Another use is for someone to get a brain print on their own for themselves, or to get a print of a friend's brain as a gift. The main ethical consideration for these types of uses is that an MRI scan is a piece of private personal health information. The only way to legally obtain the prints for our product is to receive some form of consent from the patient. In the case of a clinic being the user of the product, the clinic would have to confirm with the patient that their MRI would be provided to a third party. The clinic would need the patient's consent if the use of the model was explanatory or just as a souvenir. If someone wanted to access our product and get a gift for their friend of a model of that friends brain, that person would have to provide proof of consent from that friend as well. If someone was using our product to print out a model of their own MRI, no consent would be needed as they are the owner of their own personal health information (PHI). Since MRIs are PHI, it must be addressed that in certain situations of use of the group's product there may be some legal requirements.

Another area to consider is the environmental impact of our product. Most of the components of our product occur through software up to the 3D printing. Therefore the main impacts to the environment would occur from the material used for the models. It has been specified that the group would be using polylactic acid (PLA) filament for the models. As it turns out, using this material is a large positive in terms of being environmentally friendly. The carbon emissions which are associated with the production of PLA are 80% lower than that of traditional plastics. PLA is also considered to be biodegradable under specialized composting conditions and will typically break down within twelve weeks. PLA can also be recycled because it can be returned to its original monomer through hydrolysis or a process known as thermal depolymerization. Overall, using PLA for the brain models proves to be beneficial in terms of environmental impact in comparison to other plastics such as polyethylene terephthalate.

One last major area of consideration in terms of impact comes in the form of the societal domain. Physicians, especially those within the discipline of neuroscience, are highly technical individuals. This sometimes leads to the communication between them and their patients to not be so effective. Being so familiar with the jargon and structure of the brain may create a situation where the words used by the physician do not clearly convey the situation to the patient. That's where our product would come in. Having a personalized model that the physician can point to while explaining, and show them what and where something is going on can begin to bridge that

gap between the discrepancies of knowledge between the patient and physician. This can in turn lead to a more satisfied physician knowing that the patient understands what is going on, and a patient who can make more well informed decisions about their own situation.

Due to the novelty of the project, there are no set standards for this project process. The only potential standards are related to the 3D printers being used.

Alpha Prototype

The objective of the team's alpha prototype is to demonstrate the basic function of each of the steps within the pipeline developed for the project. Although it is not a fully automated system as is being planned for, the team wanted to attempt to execute the critical steps throughout the pipeline individually, as well as produce some prints of the brain after going through the steps. The critical steps that were separately evaluated were the segmentation, the reconstruction step, the repair step, and the printing step. Each of these steps were worked through separately and the group successfully demonstrated that the process of printing a brain this way can be done.

Once the MRI image has been obtained, it must be uploaded to Freesurfer. Freesurfer is a segmentation software program that is directly compatible with Mac and Linux operating systems. When using a Windows operating system, a software program called Ubuntu must be installed and utilized. The freesurfer environment must be programmed into Ubuntu. This allows for files to be uploaded and segmented on Freesurfer. To upload a NIFTI file to Freesurfer, the command seen in Figure 5 was used.

```
-------- freesurfer-linux-ubuntu18_x86_64-dev-20211020-b964143 --------
Setting up environment for FreeSurfer/FS-FAST (and FSL)
FREESURFER_HOME   /usr/local/freesurfer/7-dev
FSFAST_HOME       /usr/local/freesurfer/7-dev/fsfast
FSF_OUTPUT_FORMAT nii.gz
SUBJECTS_DIR      /usr/local/freesurfer/7-dev/subjects
MNI_DIR           /usr/local/freesurfer/7-dev/mni
nathan@LAPTOP-FF9FC75L:~$ cd /usr/local/freesurfer/7-dev/subjects
nathan@LAPTOP-FF9FC75L:/usr/local/freesurfer/7-dev/subjects$
nathan@LAPTOP-FF9FC75L:/usr/local/freesurfer/7-dev/subjects$ sudo cp -r <Path to File> /usr/local/freesurfer/7-dev/subjects
```

Figure 5: Command used to Upload NIFTI file to Freesurfer

The NIFTI file was uploaded to Freesurfer, where it can be segmented. This was done through a command known as recon-all. This is the command that was used to create all of the volumes and surfaces. The team found that this is the most critical step in the segmentation process. It takes anywhere between six and twenty four hours to complete. The command is seen below in Figure 6.

```
recon-all -s subj01 -i subj1_anat.nii -all
```

Figure 6: This command segments the MRI image

Next the team opened Freeview which is the Graphical User Interface for Freesurfer. This was done by typing 'freeview' into the command line. In Freeview, the NIFTI file can be accessed. Once the recon-all process is complete, the surfaces and volumes should be visible in the GUI as seen in Figure 7. They can all be verified individually to ensure that the quality is high. If the recon-all process is done correctly, the brain should look like Figure 8.

Figure 7: Surfaces and Volumes created by segmentation

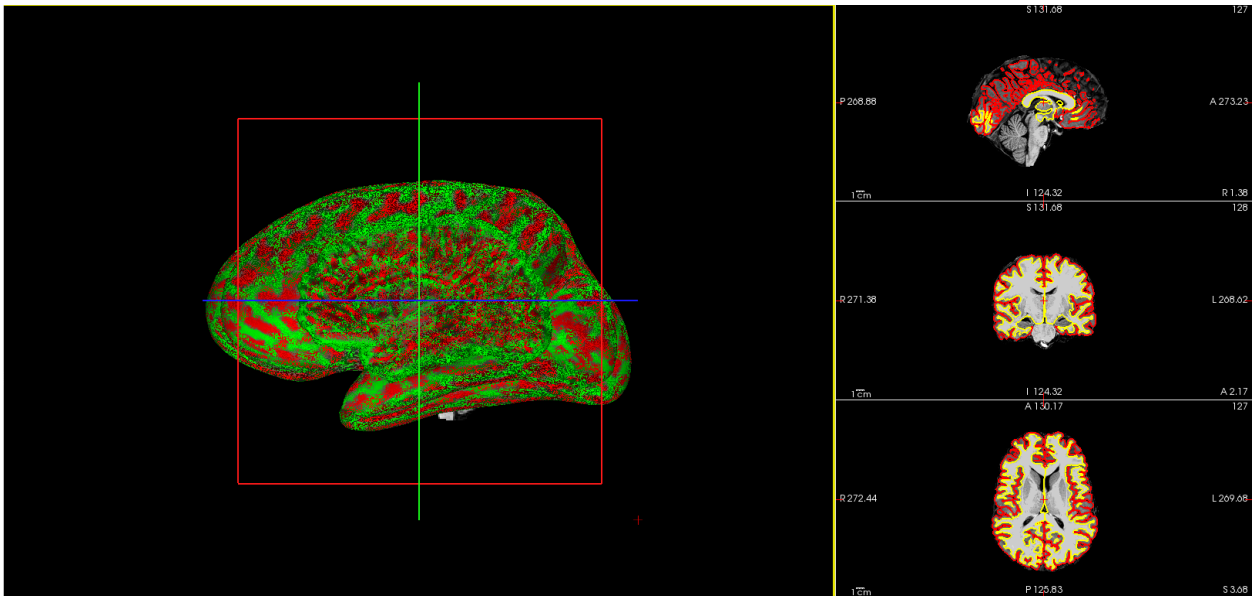


Figure 8 : Inflated view of the model segmentation

The next step in the process was to convert the brain into an .stl file. This step allows us to create a 3D model of the brain. The command that is used for this step is seen below in Figure 9.

```
mris_convert
/usr/local/freesurfer/subjects/mybrain/surf/rh.pial
rh.stl
mris_convert
/usr/local/freesurfer/subjects/mybrain/surf/lh.pial
lh.stl
```

Figure 9: Command that converts segmentation files to .stl format

After the brain is segmented, it must be reconstructed using meshing software, in this case Meshlab, so the file can be smoothed out, edited, and converted into a file that can be 3D printed. The major area of concern on this front is the smoothing out of the surface of the brain. From the segmentation software, the brain model is still more or less a stack of 2-dimensional images, which creates a relatively blocky and low resolution output (further explained in the *Engineering Design* section). In order to combat this, a mesh is created over the surfaces of interest, the outer surface and the grey matter-white matter interface, in order to smooth out the model so that it is more fit for medical and educational applications. A mesh is a clever tool used to smooth out the surface of 3-dimensional computer models without overloading the computer itself. The mesh approximates the shape of the model using polygons, typically triangles, which saves a lot of computing power compared to trying to achieve the exact dimensions. The resolution, and in turn the accuracy, of the model depends on both the number of polygons created in the initial mesh and the distance between adjacent vertices of each polygon. Below is an exaggerated example of what the brain might look like after being run through the segmentation software:

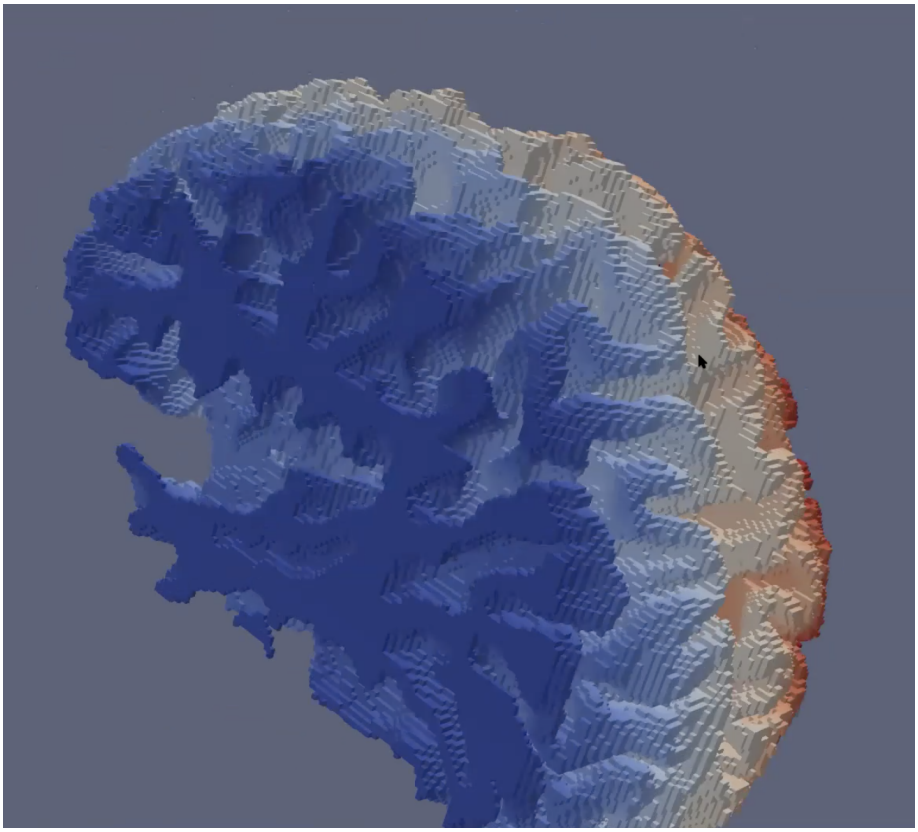


Figure 10: Example Brain Segmentation

Flatten Visible Layers
Flatten all or only the visible layers into a single new mesh.
Transformations are preserved. Existing layers can be optionally deleted

(FilterLayer)

Figure 11: Flatten Visible Layers description

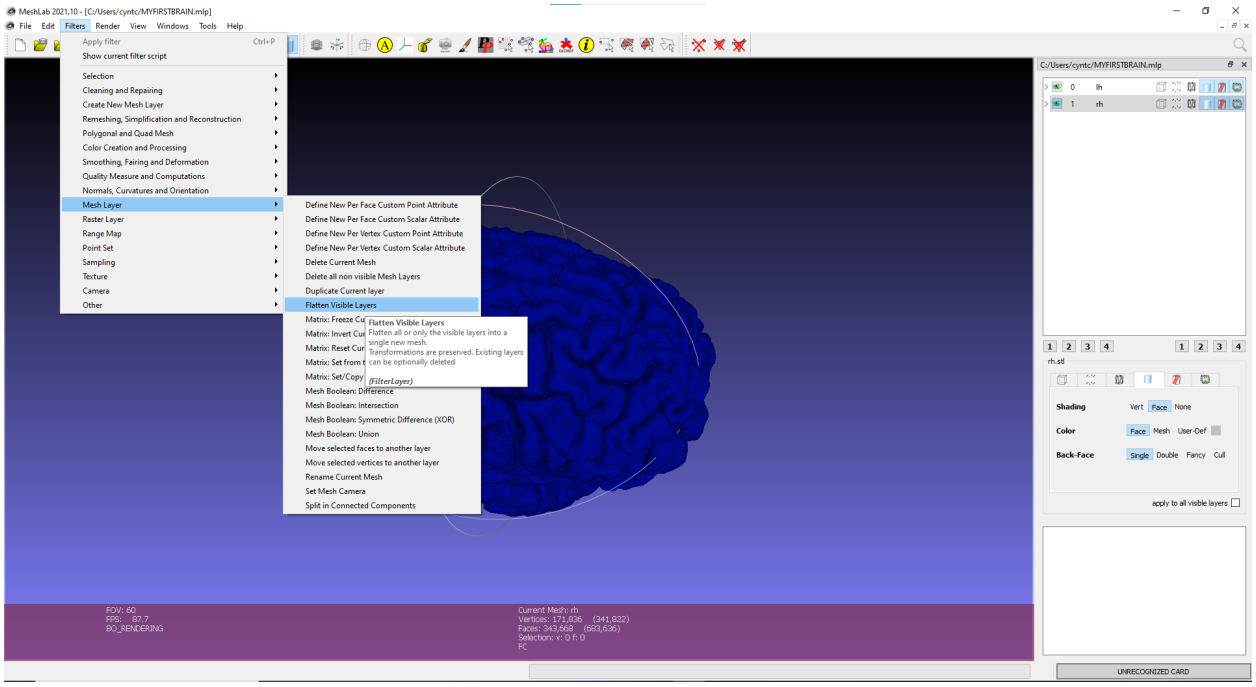


Figure 12: Path to Flatten Visible Layers

In MeshLab there are many different ways that we can ensure that the quality of our model is high. The first step that our group took was to flatten visible layers. This is an important step because it creates a new mesh. Before this step is completed, the model has noticeable layers. This results in a bumpy exterior surface and a low overall quality. Once the visible layers had been flattened, immediately the improvement of the model was apparent. It is important to ensure that this step is done correctly because it directly affects the ability to print the model.

Simplification: Quadric Edge Collapse Decimation
Simplify a mesh using a quadric based edge-collapse strategy. A variant of the well known Garland and Heckbert simplification algorithm with different weighting schemes to better cope with aspect ration andd planar/degenerate quadrics areas.
See:
M. Garland and P. Heckbert.
Surface Simplification Using Quadric Error Metrics (pdf)
In Proceedings of SIGGRAPH 97.

(FilterMeshing)

Figure 13: Quadric Edge Collapse Decimation description

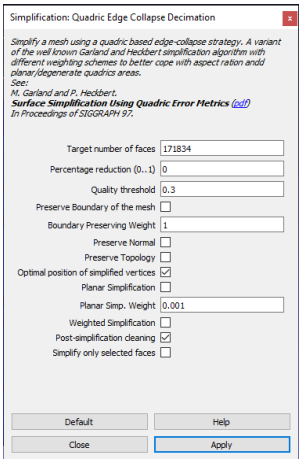
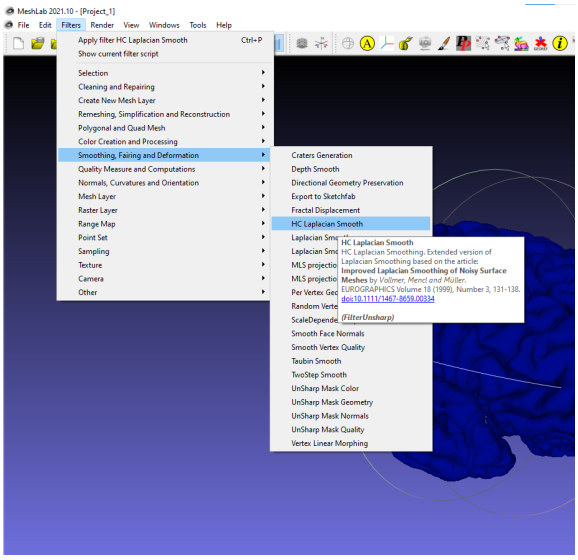


Figure 14: Quadric Edge Collapse Decimation User Interface

Quadric Edge Collapse Decimation is a meshing technique that is used to reduce the number of faces on a 3-Dimensional model. This is important because it simplifies the object and removes all of the unnecessary parts. Using this tool gives us the ability to control the number of vertices that the model has. All of the vertices in the model are connected. This connection is what gives our brain model its structure. Reducing the number of vertices can eliminate any excess material that is inaccurate. In MeshLab, we are given the option to choose the exact number of vertices or to reduce the number of vertices by a certain percentage. Moving forward our team will be looking for an exact number of vertices that will give us the optimal results. MeshLab suggests using 200,000 vertices, but we will continue to explore other possibilities.



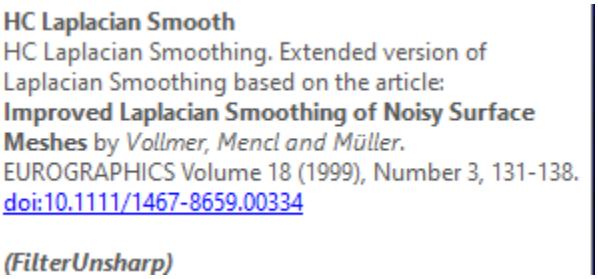


Figure 16: HC Laplacian Smooth description

Figure 15:Path to HC Laplacian Smooth

The HC Laplacian Smooth tool is used to improve the aesthetic quality of the model. It eliminates any surface imperfections. HC Laplacian Smooth is different from Quadric Edge Collapse Decimation because it doesn't alter the structure of the model. HC Laplacian Smooth only affects the surface of the model. It is the last step that should be done before the model is ready to be printed. It is a final improvement to ensure that the model is prepared for print.

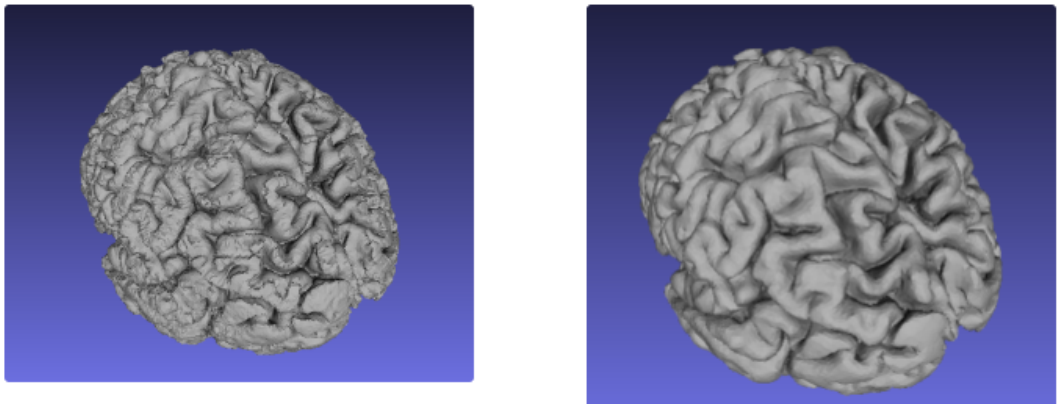


Figure 17: Before (left) and after (right) MeshLab alteration

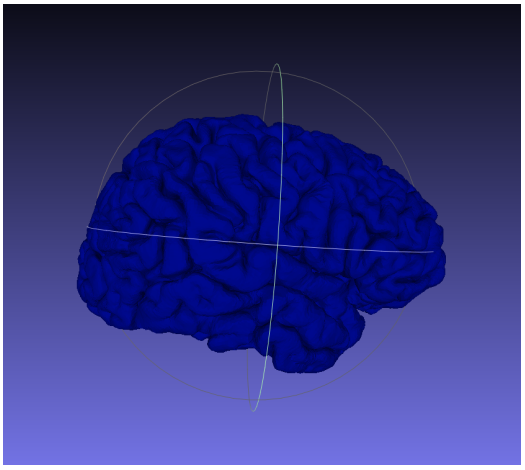


Figure 18: Finalized model

After the stl file has been developed the file will then be run through a quality check in Matlab. This is used to make sure the model meets all the requirements to be printed accurately. The first stage of the code is used to upload the image of the brain into the software. This can be seen in Figure 19.

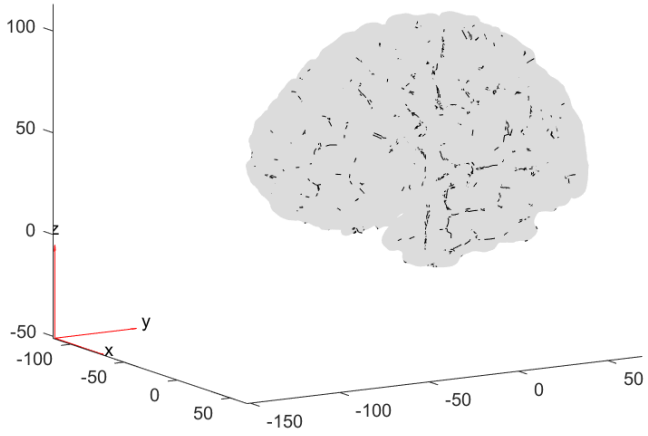


Figure 19: Image of Brain model in Matlab

After the file is in Matlab a mesh is then created. This is important for when the different checks are going to be applied such as checking for holes, adding or deleting vertices, or applying further smoothing functions on the mode. Before any of these quality checks can be completed the mesh that is added to the brain model needs to be up to the standards as well. Looking at the code that is provided in Figure 20, a mesh with a minimum and maximum edge length of 0.5 to 20 is generated and the aspect ratio of each triangle is found.

```
StructuralModel = createpde('structural','static-solid');
brain = importGeometry(StructuralModel,'Brain.stl');
pdegplot(StructuralModel,'EdgeLabels','off')

mesh = generateMesh(StructuralModel,'Hmax',20, ...
                                          'Hmin',0.5);

Q = meshQuality(mesh);
elemIDs = find(Q < 0.3);

figure
pdemesh(mesh,'FaceAlpha',0.5)
hold on
pdemesh(mesh.Nodes,mesh.Elements(:,elemIDs), ...
        'FaceColor','blue','EdgeColor','blue')
```

Figure 20: Mesh Quality check Algorithm

The aspect ratio ranges from a value of 0 to 1 with 1 representing the optimal shape for that element. In order to make sure that the mesh is of good quality a filter is created to highlight all elements that fall below a ratio of 0.3 in blue. An example of a model with highlighted elements can be seen in Figure 21 below.

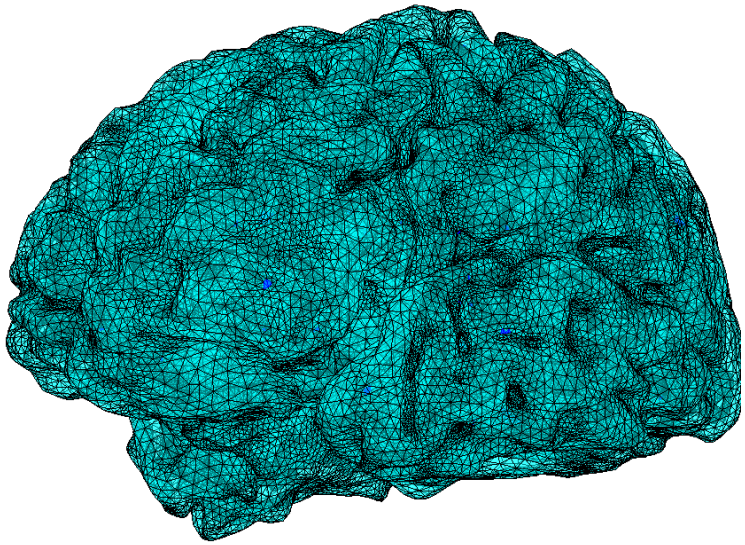


Figure 21: Brain Model with mesh elements highlighted in blue

In order to fix this the minimum and maximum values of the mesh can be altered until it passes the test. For the case of the group brain model the values were 0.5 to 20. Once this is completed the other checks can be done.

Next the brain model, now with a quality mesh, is to be checked for holes and missing mass that might be created by alternating file types throughout the pipeline process. To accomplish this, the team will implement a method that looks at each edge of the model and checks whether each has two triangles from the mesh in common. If there is a discrepancy, that means that there is a hole or gap that needs to be filled. An example of the hole finding algorithm can be seen below.

```
>> % MAKE a face/vertices structure
tmpvol = zeros(20,20,20);       % Empty voxel volume
tmpvol(8:12,8:12,5:15) = 1;     % Turn some voxels on
fv = isosurface(tmpvol, 0.99);

% REMOVE a face
fv.faces(30,:) = []; % Remove a face!

% TEST for gaps or holes
edges = sort(cat(1, fv.faces(:,1:2), fv.faces(:,2:3), fv.faces(:,[3 1])),2);
[unqEdges, ~, edgeNos] = unique(edges,'rows');
if size(edges,1) == size(unqEdges,1)*2
    % Every edge is used twice... consistent with a closed manifold mesh
    disp('No problem!')
else
    badEdgesMask = hist(edgeNos, 1:max(edgeNos))~=2;
    badEdgeNos = edgeNos(badEdgesMask);
    badNodeNos = edges(badEdgeNos,:);
    badFaceNos = find(sum(ismember(fv.faces, badNodeNos),2)>=2);
end
```

Figure 22: Hole finding algorithm

Following the identification of holes and imperfections, the stl needs to be modified in order to fill in the holes and imperfections that were identified and create a more accurate and smooth model. The team is utilizIng an stl tools package in MATLAB that contains four key functions:stlGetVerts(), stlAddVerts(), stlDelVerts(), and stlSlimVerts(), which can be seen in Appendices A1-A3. The stlGetVerts() function returns vertices that are open or closed, depending on what the user specifies. The vertices that meet the function parameters are then listed as an array of vertices and faces. The stlAddVerts() function adds new vertices to an stl mesh, with the input of the function being an array of faces and vertices, likely derived from the stlGetVerts() function. This function can be utilized to add mass to the brain model and fill in the holes and imperfections that the user specifies. The stlDelVerts() function does the opposite of the stlAddVerts function, in that it deletes vertices specified by an array of faces and vertices. This function is important to delete extraneous mass and help smoothen out the brain model. Lastly, the stlSlimVerts() function finds and removes duplicate vertices in surface meshes. This is an important function as it helps to smoothen out the mesh and additionally helps to reduce the vertex count of the surface mesh. The input of this function are two lists, one being the list of total lists and the other being vertex lists that define each triangle face. The figure below shows a test of these functions on a sphere, where the vertices below z=0 are deleted, leaving a large hole. The stlGetVerts() function is then used to create a lists of vertices that are open(the hole) and the stlAddVerts() function is then used to close the hole. These tools are all imperative in order to improve the quality and accuracy of the model, helping to meet the specifications of creating an accurate and quality model.

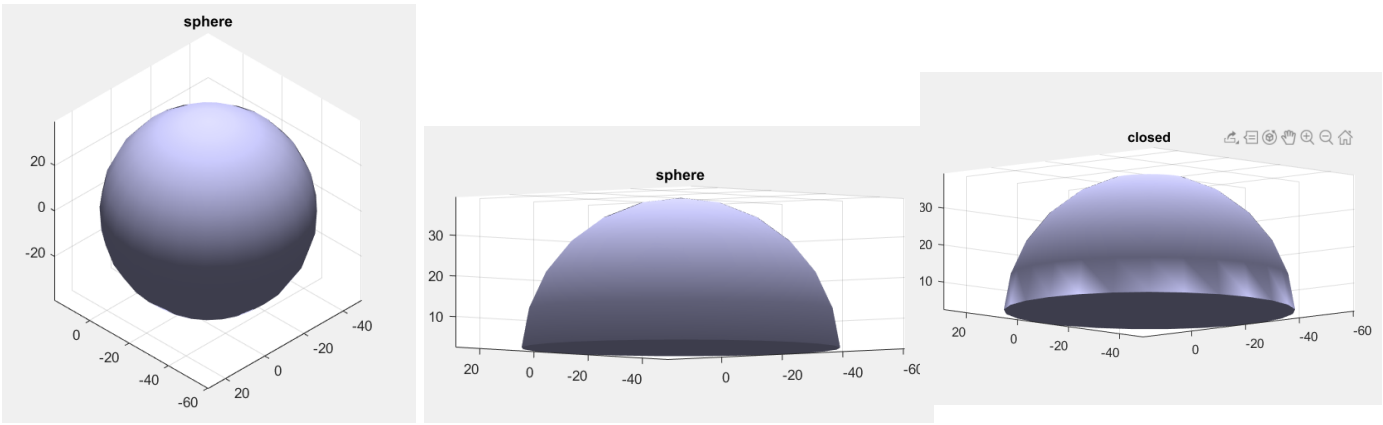


Figure 23: Example of stl modification functions used on a sphere

Lastly, the modified and smoothened stl model is ready to be 3D printed. The stl model is sliced using a slicing software (the Proof Lab will choose a slicing software of their choice). This will create a gcode file, which is a file that tells the printer which movements to make to create the model. Finally, the file is input into the 3D printer and after a number of hours, the brain model is printed and ready to be used. Below is a test model of a brain (scaled down to 25% size)

the team printed using the prototyped method. Due to complications with the proof lab staff, the model was not able to be printed using the Objet Connex with clear PLA at the present time of this report. However, the model will be printed with these parameters in the near future. To accommodate, a Flashforge Finder with black PLA filament was used as shown below.

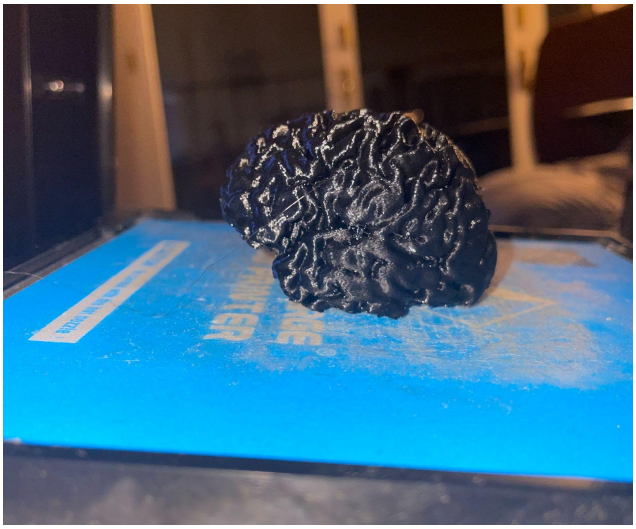


Figure 24: Printed model of brain using black PLA

Project Planning

While there is no sponsor for the project, due to the licenses and printers provided by Stevens, the only cost on the team budget is for PLA filament. As found in the cost analysis, the cost of 2kg of PLA filament is around $50. Due to the model being created with two materials, two spools of filament will be needed, which brings the cost to $100. However, the team found 1kg of PLA from Micro Center to be around $15 for black and 19 for white, which would bring the cost to $68 rather than $100. Even without 3D printers available from the school, one of the team members has a personal 3D printer, which would still make the 3D printer cost invalid.

The Bill of Materials below is a living document to track what the team will be purchasing in the remaining phases of the project.

BOM Item Number	Description	Vendor	Cost (each)	Quantity	Purchased	Received
1	Black PLA Filament (1 kg)	Micro Center	$15	2	No	No
2	Natural PLA Filament (1kg)	Micro Center	$19	2	No	No

Table 3: Current Bill of Materials

In Phase 1, the team was able to define project objectives, perform state-of-the-art review, identify customer needs and specifications, brainstorm conceptual ideas, develop initial concept designs, and identify areas for technical analysis. In Phase 2, the team was able to finalize concept selection and designs, perform technical analysis, and develop a project website. In Phase 3, the team was able to develop detailed engineering design and update product specifications, finalize technical analysis and undergo alpha prototyping of critical sub-systems. Below is the team's Gantt chart for Phase 3, visualizing the distribution of time and tasks during this phase.

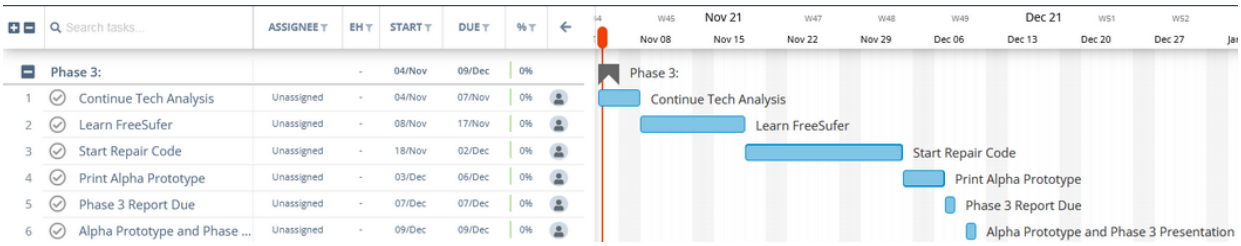


Figure 25: Phase 3 Gantt Chart

Moving forward the team plans on implementing the functions they learned in phase three into code that will work on brain models, specifically referring to the stl modification section of the pipeline. Up until now, the team was learning different functions they could use to modify the brain stl to smoothen and more accurately represent the model. Now that these functions have been learned, the team will need to implement them into their final code that will be used on the brain stl to a satisfactory degree. Additionally, a major tasks for the team moving forward is making the process automatic and connecting the different steps of the pipeline so that there only needs to be human intervention at the very beginning and end. This will involve finding an interface that will connect all parts of the code together into one, coherent program. Lastly, the team will need to figure out a method that will collect mri scans and send them through the project pipeline, presumably an app or web interface. Overall, the team has made substantial progress with the alpha prototype, which in turn has opened the pathway to move onto bigger tasks such as those listed above.

References

https://www.biopak.com/sg/resources/what-is-pla
https://www.makeshaper.com/3d-printer-filament-price-cost/
https://www.mathworks.com/matlabcentral/fileexchange/51200-stltools
https://www.mathworks.com/matlabcentral/answers/27886-find-holes-and-gaps-in-stl-files
https://www.microcenter.com/product/611537/inland-175mm-natural-pla-3d-printer-filament-1kg-spool-(22-lbs)
https://www.microcenter.com/product/486462/285mm_Black_PLA_3D_Printer_Filament_-_1kg_Spool_(22_lbs)?storeID=075

Appendix

A1- stlGetVerts() Function

```
function list = stlGetVerts(v, f, mode)
%GETVERTS returns the vertices that are 'opened' or 'closed' depending on
%the 'mode'. An 'open' vertice is the one that defines an open side. An
%open side is the one that only takes part of one triangle
%V is the Nx3 array of vertices
%F is the Mx3 array of faces
%MODE can be 'opened' or 'closed' depending of the kind of vertices to list
%LIST is the list of 'opened' or 'closed' vertices
sides = sort([[f(:,1) f(:,2)]; ...
   [f(:,2) f(:,3)]; ...
   [f(:,3) f(:,1)]],2);
[C,ia,ic] = unique(sides,'rows');
ind_all = sort(ic); % open and closed sides
ind_rep = find(diff(ind_all) == 0);
ind_cls = ind_all(ind_rep); % closed sides
sides_cls = C(ind_cls,:);
ind_rep = [ind_rep; ind_rep+1];
ind_opn = ind_all;
ind_opn(ind_rep) = []; % open sides
sides_opn = C(ind_opn,:);
switch mode,
   case'opened',
       list = v(unique(sides_opn(:)),:);
   case 'closed',
       list = v(unique(sides_cls(:)),:);
   otherwise,
       error('getVerts:InvalidMode','The ''mode'' valid values are ''opened''
or ''closed''');
end
```

A2 - stlAddVerts() Function

```
function [vnew, fnew] = stlAddVerts(v, f, list)
%STLADDVERTS adds new vertices (and consequently, new faces) to a STL object
%V is the Nx3 array of vertices
%F is the Mx3 array of faces
%LIST is the list of vertices to be added to the object
%VNEW is the new array of vertices
%FNEW is the new array of faces
% triangulation just with the slice
faces = delaunay(list(:,1),list(:,2)); % calculate new faces
% update object
nvert = length(v); % number of original vertices
v = [v; list]; % update vertices with the ones in the list
f = [f; faces+nvert]; % update faces with the new ones
[vnew,fnew] = stlSlimVerts(v,f); % clear repeated vertices
```

A3- stlDelVerts() Function

```
function [vnew, fnew] = stlDelVerts(v, f, list)
%STLDELVERT removes a list of vertices from STL files
%V is the Nx3 array of vertices
%F is the Mx3 array of faces
%LIST are the vertices (rows) to delete, where length(LIST) < N
%VNEW is the new array of vertices
%FNEW is the new array of faces
% find (on the global set) the position (rows) of the vertices to be deleted
[~,vdel] = ismember(list,v,'rows');
% delete vertices and get new tags
vnew = v;
tags = 1:length(v);
vnew(vdel,:) = [];
tags(vdel) = [];
% delete faces
fnew = f;
[fdel,~] = find(ismember(f,vdel)); % find the position (rows) of the faces to
delete
fnew(fdel,:) = [];
% rename faces, as some of the vertices have been deleted
flist = reshape(fnew,numel(fnew),1);
[~,ind] = ismember(flist,tags);
fnew = reshape(ind,numel(flist)/3,3);
```

A* - stlSlimVerts() Function

```
function [vnew, fnew]= stlSlimVerts(v, f)
% PATCHSLIM removes duplicate vertices in surface meshes.
% This function finds and removes duplicate vertices.
% USAGE: [v, f]=patchslim(v, f)
% Where v is the vertex list and f is the face list specifying vertex
% connectivity.
% v contains the vertices for all triangles [3*n x 3].
% f contains the vertex lists defining each triangle face [n x 3].
% This will reduce the size of typical v matrix by about a factor of 6.
% For more information see:
%http://www.esmonde-white.com/home/diversions/matlab-program-for-loading-stl-fi
les
% Francis Esmonde-White, May 2010
if ~exist('v','var')
   error('The vertex list (v) must be specified.');
end
if ~exist('f','var')
   error('The vertex connectivity of the triangle faces (f) must be
specified.');
end
[vnew, indexm, indexn] =  unique(v, 'rows');
fnew = indexn(f);
```