High-Fidelity Mesh Improvement for MRI-Derived Anatomical Models

Project Update 1

Shubham Vikas Mhaske
CSCE 645: Geometric Modeling
Github

October 31, 2025

1 Introduction

This document summarizes the progress for the first project update (due Friday, October 31), as outlined in the project proposal [1]. All stated goals for this milestone have been successfully completed.

The work for this update is divided into three components [1]:

1. **Literature Review:** A thorough review of mesh smoothing and simplification methods.
2. **Data-to-Mesh Pipeline:** The establishment of a reproducible data pipeline to download the BraTS dataset and generate a raw 3D tumor mesh.
3. **Baseline Analysis Suite:** The implementation of a quantitative analysis suite to compute and visualize initial mesh quality metrics.

This report details the implementation of the pipeline and presents the baseline analysis. The full literature review is submitted as a separate document [2].

2 Literature Review Summary

A comprehensive literature review was completed to identify the foundational algorithms for this project [2]. The review confirmed that while Marching Cubes is the standard for isosurface extraction, it produces meshes with significant "staircase" artifacts and poor-quality triangles.

The review identified two key algorithms to address these issues [2]:

- **Taubin Smoothing:** A feature-preserving smoothing operator acclaimed for its ability to remove high-frequency noise while rigorously preserving model volume, a flaw that plagues simpler methods.
- **Quadric Error Metrics (QEM):** An algorithm for mesh simplification that intelligently reduces triangle count by prioritizing the preservation of high-curvature features, essential for maintaining anatomical detail.

These findings validate the project's proposed pipeline [1].

3 Pipeline Implementation

To fulfill the goals for the pipeline and analysis suite, a complete software package was developed using Python and Streamlit [3]. The application (`app.py`) provides a graphical user interface for loading data, processing the mesh, and visualizing the results.

3.1 Data Acquisition

A script (`download_data.py`) was implemented to interface with the Synapse data repository. It uses the `synapseclient` library to log in and download the BraTS 2023 dataset (Synapse ID: `syn64952532`) [3].

3.2 Mesh Generation

The core `process_nifti_to_mesh` function in `app.py` executes the initial "data-to-mesh" pipeline [3]. It performs the following steps:

1. **Load Data:** It reads a `.nii.gz` segmentation file using `nibabel`.
2. **Create Mask:** It processes the segmentation mask by combining the tumor component labels (Label 1, 2, and 4) into a single binary volume, as described in the literature review [2].
3. **Run Marching Cubes:** It uses the `skimage.measure.marching_cubes` algorithm to extract the isosurface, generating the raw vertex and face data [3].

3.3 Quantitative Analysis Suite

The Streamlit application serves as the "quantitative analysis suite" [3]. After generating the raw mesh, it immediately computes and displays the baseline metrics proposed for evaluation [1]:

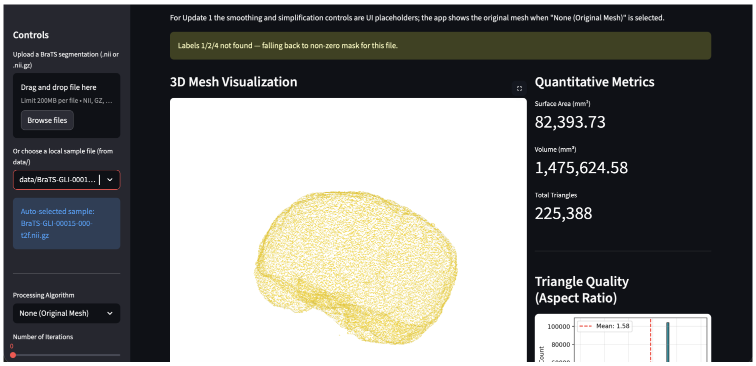
- Surface Area (mm^2)
- Volume (mm^3)
- Total Triangle Count
- Triangle Aspect Ratio Histogram

4 Baseline Analysis Results

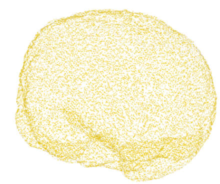
The implemented pipeline successfully generates the raw mesh and its corresponding baseline metrics. These results clearly demonstrate the known issues with raw Marching Cubes output and provide the "before" state for our "before-and-after" comparisons.

Figure 1 shows the visual baseline results. The full application dashboard is shown in Figure 1a, displaying the rendered mesh alongside the calculated metrics (e.g., 225,388 triangles). A close-up of the wireframe mesh is provided in Figure 1b, which clearly shows the "staircase" artifacts.

Figure 2 presents the baseline quantitative analysis of the mesh's triangle quality [1]. The dashboard view (Figure 2a) and the close-up (Figure 2b) show the distribution of aspect ratios, with a mean of 1.58. The presence of distinct, non-ideal peaks confirms the poor quality of the initial mesh elements, which the subsequent smoothing pipeline will be designed to improve.

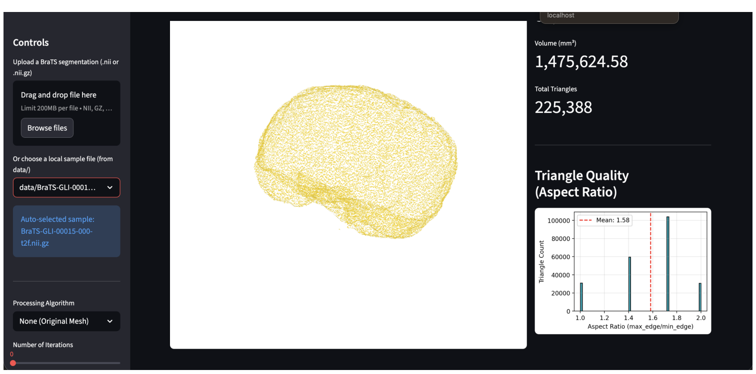


(a) Full dashboard view of the running application.

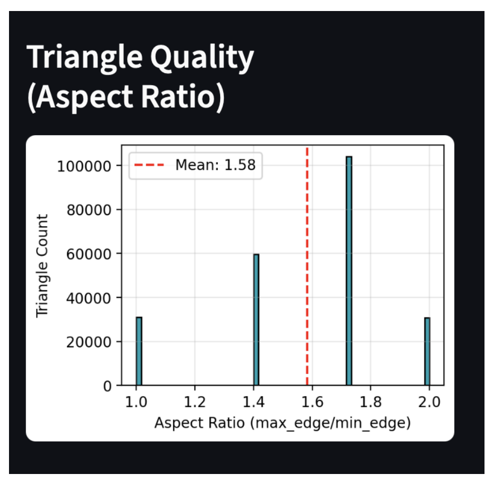


(b) Close-up of the raw mesh, highlighting "staircase" artifacts.

Figure 1: Visual baseline of the raw Marching Cubes mesh generated by the pipeline [3].



(a) Dashboard view highlighting the quantitative metrics.



(b) Close-up of the "Triangle Quality (Aspect Ratio)" histogram.

Figure 2: Quantitative baseline analysis of the raw mesh's triangle quality.

5 Plan for Second Update

With the baseline pipeline and analysis suite complete, work will now proceed toward the goals for the second update (due Monday, November 24) [1]. The immediate next steps are:

- Complete the implementation and testing of a baseline **Laplacian smoothing** operator for comparison.
- Complete the core implementation of the advanced **Taubin smoothing** algorithm.
- Produce a **comparative analysis** of the two smoothing methods, rigorously evaluating volume preservation and noise reduction using the established metrics.

References

[1] S. V. Mhaske, "High-Fidelity Mesh Improvement for MRI-Derived Anatomical Models: Project Proposal," CSCE 645: Geometric Modeling, Oct 2025.

[2] S. V. Mhaske, "High-Fidelity Mesh Improvement for MRI-Derived Anatomical Models: A Literature Review," CSCE 645: Geometric Modeling, Oct 31, 2025.

[3] S. V. Mhaske, "Project Update 1: Python Source Code," CSCE 645: Geometric Modeling, Oct 31, 2025. (Files: `app.py`, `download_data.py`, `requirements.txt`).