

Pranav Rajan
CS 6635
February 24, 2021

Part 1: Load the Data

Question 1: Visualization of Statistics for 1-D Data

- 1) The number that occurred most frequently was 10.395 which appeared 10 times in the data when using 100 bins for the histogram
- 2) 15 numbers were not used by the class

Question 2: Visualization of 2D Images

- 1) Threshold values: Min - 14, Max - 110. Experimenting with different threshold values, I found that the more land appears/disappears and so does the river.
- 2) Number of points according to information panel: 4658942

Question 3: Explore Data on Polygonal Meshes

- 1) Threshold values: Min - 0.1, Max - 44
- 2) 24 Ventilation slots based upon the image

Question 4: Visualization of 3D Images

See Paraview files

Part 2: Python Script

Question 1: Batch Script

See Paraview Files. Python Script for image 1 is arrow.py. Python script for image 2 is arrow2.py

Question 2: Read and Process Files with Python Scripts

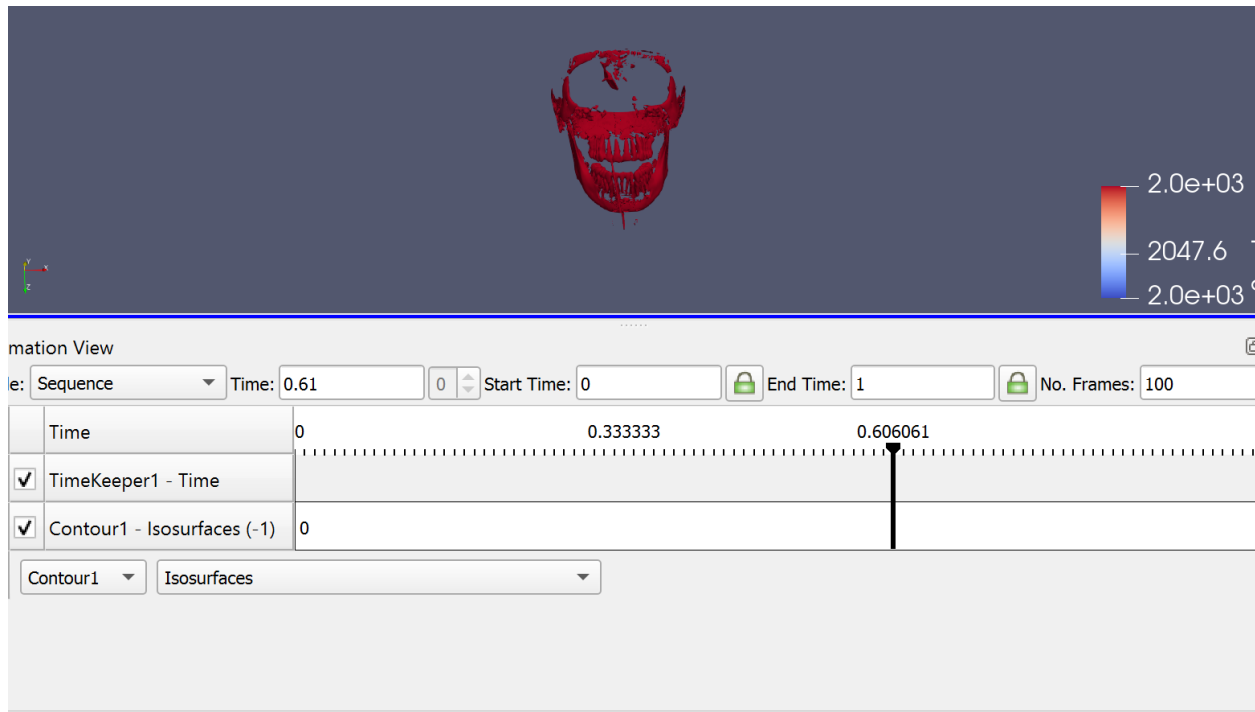
See Paraview Files. Python Scripts in this order for the images: loadData, plotOverline, riverbed3d.

Part 3: Time-Dependent Isosurface (CS 6635)

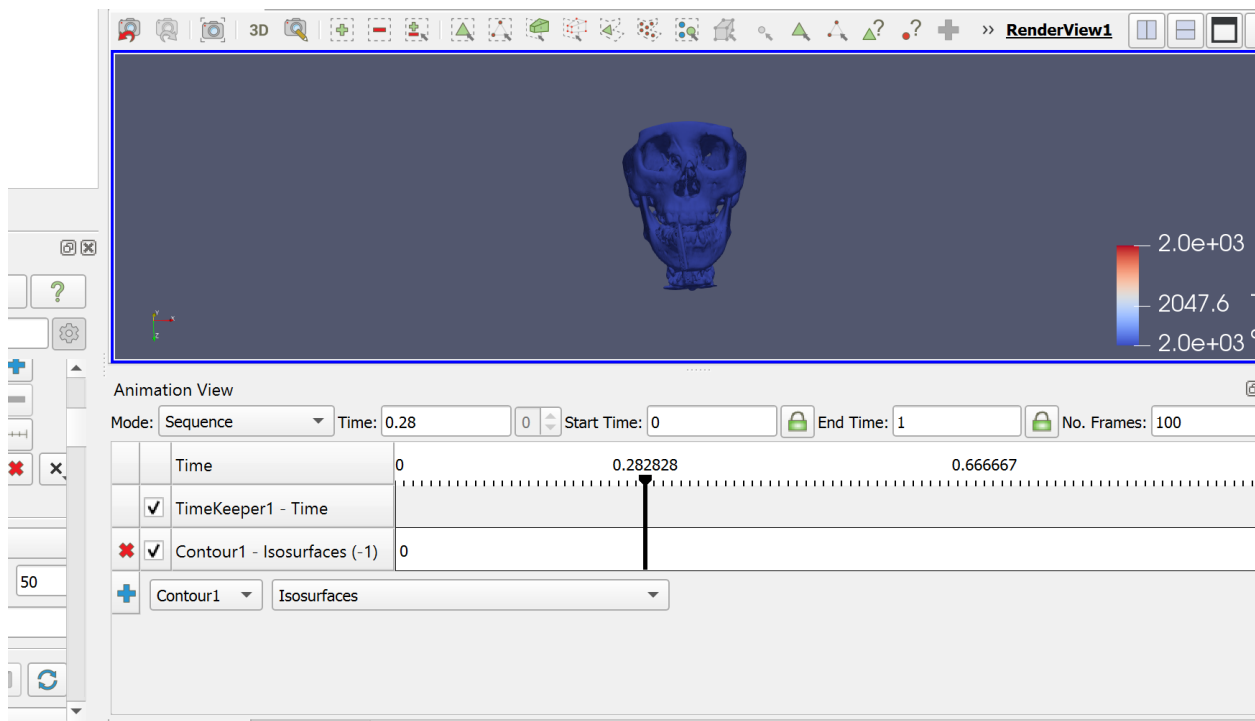
Question 1: Isosurface Animation

Skin to Skull range - 455-1158 Isovalues

Spine Image - Isosurface Values: 2481.818



Teeth Image: Isosurface Values 1158



Question 2: Reading Questions

- 1) Time varying data is data that changes over time such as that in applications such as physics, engineering and other scientific applications. One important aspect for visual data analysis with time varying data is that "...the visualization software must be able to compute animate, and track desired features at an interactive speed" (Hanwei Shen, Visualization Handbook). The marching cubes algorithm while fast for isosurfaces can be quite computationally expensive.
- 2) The need for the hierarchical index tree data structure is to reduce the overhead for the amount of disk I/O operations for accessing indices of the data at different time steps when there is not enough memory to store the animation
- 3) Given a temporal hierarchical index tree, traverse the tree and return a set of nodes that contain the information needed to visualize the data at a particular time step.