

Nearly Raw Raster Data Visualization

Rick Steinmetz¹

University of Illinois

ABSTRACT

Given a Nearly Raw Raster Data (NRRD) file, provide a website that allows users to view the dataset in 3-dimensional space. The user should be intuitively able to navigate the image to see different viewpoints. Given CT scan data, allow a color map to be applied that lets the user identify different types of tissue (skeleton, tissue/muscle, skin, etc.)

1 INTRODUCTION

This fall my father fell off a ladder while doing home repairs and fractured his ankle. This gave me some inspiration when thinking about the data that was captured as part of his CT scan to see if there was a way to visualize the captured data for common users without needing any equipment or software that would be available in a hospital setting. I began to investigate if something like this existed that I could build off of.

2 RUNNING THE VISUALIZATION

In order to run the visualization, unzip the provided zip file and run your favorite http server on it. Personally, I installed the “**http-server**” package with **npm**. You can run an “**npm i**” in the project folder and the dependency will be installed using the “**package-lock.json**” file. The other option is to install it globally with “**npm i http-server -g**”. Any http server should be sufficient to run the code. I tested in the Chrome browser. I also have a copy hosted on my [github.io](#) page².

3 NRRD

I needed to look into how a CT scan is stored. A CT scan is most commonly stored in a Digital Imaging and Communications in Medicine (DICOM) file format. I wondered what the main difference between the DICOM and NRRD format of the sample file I found are. There are a few benefits to using the NRRD format over DICOM.

3.1 Anonymous

The DICOM file format contains personally identifiable information (PII) about the patient. This prevents these files from being shared and also explains why the hospital would not provide me with the dataset, as it would violate HIPPA.

3.2 Single File

Another benefit to NRRD files is that the data can be stored in a single file. DICOM data is stored in multiple files which for my use case simplified the sharing and loading of the dataset.

4 OBTAINING THE DATA

Unfortunately, due to these HIPPA regulations, I was unable to obtain the raw CT scan data of my father’s injury. The best the hospital could do was get me static images which didn’t help me visualize it in a way that I wanted.

Searching the internet, I found the website³ of Pavol Klacansky: a Graduate Research Assistant at the University of Utah. On this

website there is aggregated data of Open Scientific Visualization Datasets, which was a great starting point to work from. One of the datasets provided is from Phillips Research, Hamburg, Germany and has data form a rotational c-arm x-ray scan of a human foot containing both tissue and bone in the dataset. This was exactly what I needed for this project. It provided me with an analog for my father’s injury that I could visualize the type of data that the doctors could see when diagnosing my father.

The base functionality is viewing the NRRD file in 3-dimensional space and modifying the iso value to see the change in the data that is displayed.

5 RENDERING NRRD FILES

Pavol Klacansky provides a **dvr.js** package that is a surface and volume renderer using OpenGL 2.0 and also available to the public domain which allows modifications to this package. Using some JavaScript we load the NRRD file from disk and use his package to display the visualization.

5.1 EXISTING VISUALIZATION

Looking at this dataset on his webpage (Figure 1), I wanted to look at ways to improve what he had already built. He has a very basic representation of an NRRD file where you could move the scan in three dimensions and zoom in and out. He also has a slider that you can adjust to change the iso value that is being displayed. However, the entire surface was one color and made it difficult to distinguish between the different types of elements.

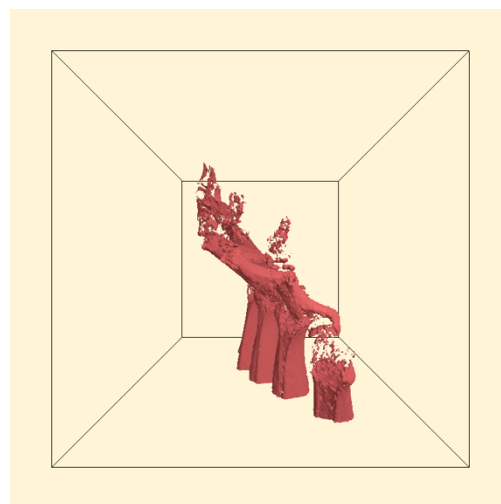


Figure 1: 3-dimensional rendering of the tissue.

6 IMPROVEMENTS

I decided on four improvements to apply to his rendering algorithm to enhance the viewing experience.

6.1 Contrast Improvements

The background color that the original project chose is a light cream color. Given that I wanted to attempt to modify the visualization to display different types of elements, I needed a darker display to contrast more with the colors that I would be using for the elements. This was a simple enough change to change the background color of the website using Cascading Style Sheets. Given that I updated the background to black, I also wanted to keep the bounding box of the visualization, so I updated that color scheme to white to stand out from the now black background.

6.2 Isovalue Clamping

The isovalues inside of the NRRD file are stored in the range from 0 to 1. Looking at the rendered image, I made the decision to clamp the possible values for the user to select from to be in the range from 0.05 to 0.5. The reason I decided to clamp these values is if you look at isovalues below 0.05, there is too much noise (Figure 2) that appears on the visualization which makes it difficult to see details.

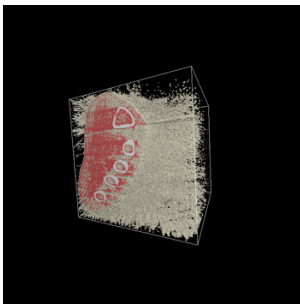


Figure 2 : isovalue set to 0.00

At 0.00 you can see a lot of extra noise in the image around the edges of the bounding box. If I clamp the possible values to 0.05 that noise is eliminated (Figure 3). The same goes for having an isovalue above 0.5. Once you start getting into higher isovalues the amount of bone that is displayed gets very small and it is difficult to see any patterns/trends. I found the sweet spot for getting the most out of the visualization was to clamp the top end of the isovalue to be 0.05. This shows a little extra noise on the edges of the 3D visualization but preserves the detail on the part of the visualization that I care about most.

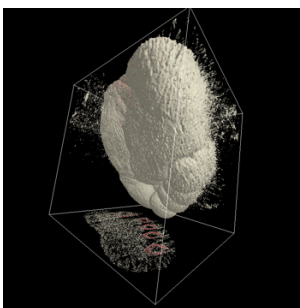


Figure 3: isovalue set to 0.05

If we scale the isovalue to be too high, in this example, greater than 0.50, you start to lose a lot of the detail of the image which is not ideal. Due to this, I clamped the top end value to 0.50. (Figure 3)

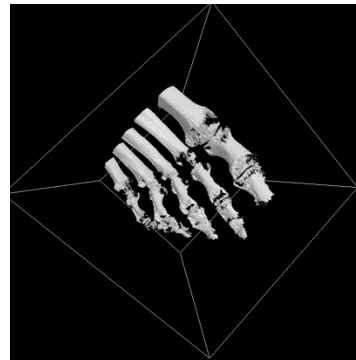


Figure 4: isovalue set to 0.50

6.3 Color Map

I wanted to apply a color map to distinguish between different isovalue levels and apply three different colors to as described in Table 1.

Type	isovalue Range	Red	Green	Blue
Bone	0.00 – 0.15	1.00000	1.00000	1.00000
Tissue/Muscle	0.15 – 0.50	0.07800	0.13000	0.13000
Skin	0.50 – 1.00	0.80000	0.76078	0.59608

Table 1: RGB Color Map

6.4 Updating Code

The package contains code to render both volume and surface visualizations. Given the goal of rendering CT scan data, I did not need the volume rendering. I went through the code based and cleaned up the code and removed the volume rendering logic which reduced the code size by about 20%. This was also done as a learning experience on WebGL 2.0 and how to render the visualization inside of a webpage. I also modified the code that loads the NRRD file from disk to follow the newer ES6 JavaScript standard.

7 SUMMARY

Overall, this was a fun project to work on and be able to see the data that a medical professional would want to see. This idea made me more interested in taking the Machine Learning for Healthcare class that is now being offered at the University of Illinois. I'm hoping that I can continue to use what I learned doing this project in the future!

8 REFERENCES

- [1] ras11@illinois.edu
- [2] <https://rsteinmetz.github.io/cs-519/>
- [3] <https://klacansky.com/open-sci-vis-datasets/>