

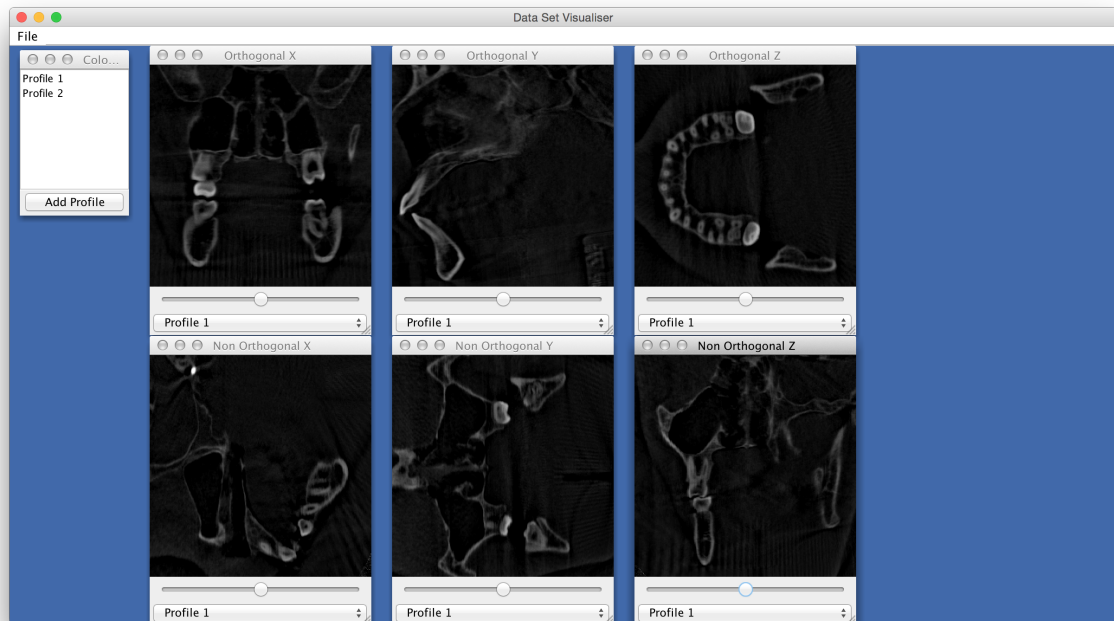
1 Description

1.1 Loading Data

The program created allows the user to select any “.raw” file and visualise the data. Due to limitations in the .raw formats meta data it is easy to decipher the dimensions of the data held within and as such it is necessary for the user to supply this information prior to loading the file. Views Once loaded the user will be presented with 6 visual representations of the data

- 3 orthogonal planes scrolling along the x, y and z axes respectively.
- 3 non-orthogonal planes slicing at angles down the x, y and z axes.

Each view contains two control elements, a slider to iterate through the given axis or angle and a combo box to allow a user to control the view's color profile.



1.2 Color Profiles

A user is able to create color profiles to be used across the views. These profiles work by the user selecting colors for a selected key data values, the view will the interpolate between these color steps to produce the desired color. e.g. consider the following color profile (integer RGB values)

$$0 - 0, 0, 0(\text{Black})$$

$$120 - 255, 0, 0(\text{Red})$$

$$255 - 255, 255, 255(\text{White})$$

There are 128 steps between the first two key steps so each step will calculate its own color based on the following: let m = current data value, c = n 's color, n = first data value in current range, o = the last value in the current range,

$$c = n.\text{color} + (o.\text{color} - n.\text{color} / o - n) * m - n$$

So if $m = 80$: $c = (0, 0, 0) + (2.125, 0, 0) * 80 = 170, 0, 0$ At any point a user can create and use new color profiles to use on the various views.

2 Critical Analysis

2.1 Strengths

There are various strengths in this application:

2.1.1 Simple To Load

Removing the step where a user selects a plane takes away ambiguity that may pop up in a user's misunderstanding of the various axes and planes.

2.1.2 Color Profiles

Initially color steps were saved independently to each view meaning a user couldn't select that profile on a different view, color profiles are view independent.

3 Weaknesses

There are a few issues with the application that could be rectified or ironed out in future iterations.

3.1 Functionality

There are various desired facets that are not present in the final application which were omitted due to time constraints

Loading When a user loads the data files there is no obvious depiction of the file loading, this could lead to the user not thinking that anything is happening.

Single Larger View An early plan for the application was to allow the user to select an active view and see a much larger rendition of it.

Mousewheel Control Allowing the user to control the current slider through the use of the mouse wheel would have been beneficial to allow the user a more intuitive experience.

Zoom Originally planned as part of the single large view, this would have allowed the user to zoom into a given view.

Current Plane Depiction Showing a visual representation of the current plane is a much more simple and easy to understand than text representations.

3.2 Coding

Loops Various points in the code are utilising nested for loops, these could be replaced with single loops to speed up execution.

Floating Points In various points of the application floating point arithmetic is used. Converting the floating point calculations to integer calculations would also allow for faster execution.