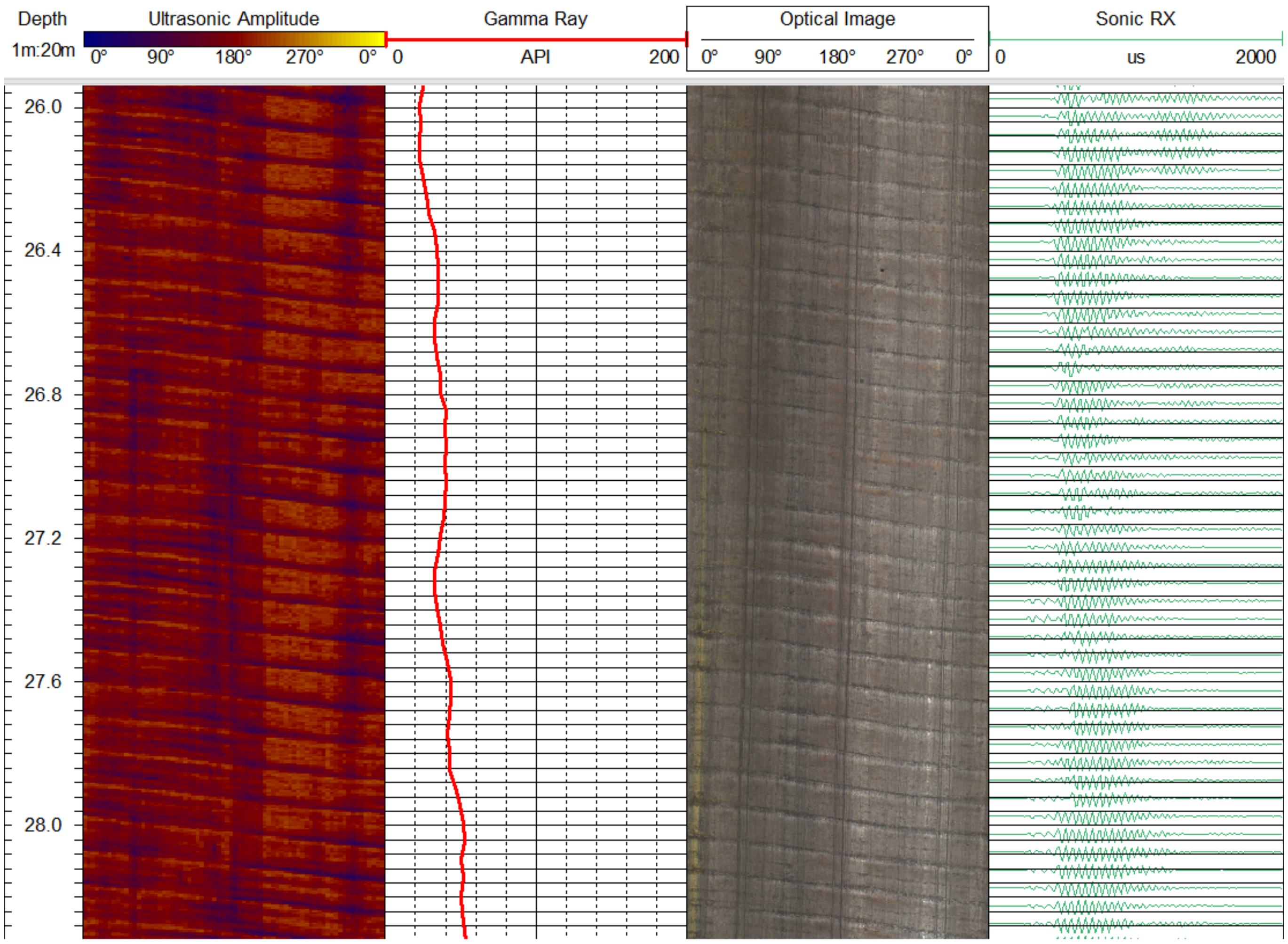
Background

Logging data takes many different forms. Below are a few examples of datasets that will be described.



Reference Axis

To the very left, a depth scale is shown. When logging data is presented, the reference axis is the vertical axis rather than the typical horizontal axis.

In general, logging data will be referenced against depth and/or time. Some information about these references follows:

- Usually there will be both depth and time references, but sometimes you may have logs with only a depth or only a time reference.
- For this exercise, it can be assumed that times are seconds since the start of the log, and not date/time/timezone representations.
- For this exercise, it can be assumed that depths are in metres.
- Sometimes the sampling period is constant in depth (for example 0.05, 0.10, 0.15, 0.20, ... m), but often it is not.
- Sometimes the sampling period is constant in time (0.01, 0.02, 0.03, 0.04, 0.05, ... s), but often it is not.
- Very rarely will the data be sampled constantly in both depth and time (this requires running the tool at a constant velocity).

- Reference times will always be strictly increasing.
- Reference depths will be piecewise monotonic increasing/decreasing. It is possible for a time sampled log to have sections where the depth is increasing followed by sections where it is decreasing, and even sections where the depth is static.

Log 1 – Ultrasonic Amplitude

The leftmost log is an amplitude image from an ultrasonic imaging tool.

The horizontal axis represents an angle azimuthally around the borehole, and thus the left and right edges are the same angle (0° and 360° represent the same azimuth).

Each row in the image is referenced to a particular depth/time, and thus the log can be plotted both against depth, or against time.

There is a fixed number of columns, and the columns are evenly spaced (in this case 0.6° apart).

Values for each grid point (point with coordinates [depth/time, azimuth]) are floating point numbers in arbitrary units. To display them, we map the values to a colour via a gradient.

Log 2 – Gamma Ray

The second log is a log of a single numeric value for a given depth/time, in this case the background gamma ray activity measured by a gamma ray tool.

The horizontal axis represents the activity level, and a regular line chart is created that plots depth/time vs. activity level. The gamma activity level is a floating point value and in units of API.

Log 3 – Optical Image

The third log is an azimuthal image log like log 1, but rather than floating point values it has RGB tuples at 8 bits per channel representing a colour for each point. Some images can have high dynamic range and may be 10 or 12 bits per channel.

Log 4 – Sonic RX

The fourth log is an acoustic waveform from a sonic tool.

The horizontal axis in this case represents time. The values are floating point values, but rather than being mapped to a colour via a gradient, the data is plotted as individual waveform plots at the depth/time they were sampled at.

For sonic data, there is no guarantee that each waveform has the same number of points, or a fixed sampling rate, or is even sampled at the same times.

Tasks

Data Structure

You will have noticed that each of the example logs above share similarities – they're all arrays of various types of data with various structures (evenly gridded, unevenly gridded, single dimensional, ungridded, ...). The first task is to define data structures that can represent the types of data encapsulated in these logs. Note, it isn't the visualisation you're trying to represent, but the actual data in the logs themselves.

The data structure(s) you arrive at should allow for:

- Memory mapping of data
- Allow rapid access to any part of the data
- Be fast to search for particular depth or time ranges
- Encapsulate units, names and axes
- Be extensible such that in the future it can be extended to more value types, gridding and higher dimensionality

Please code these as C-style structs in a header file with sufficient comments and/or documentation to describe each member of the struct, what it represents and how it can be used.

Interface

Let's hide a bunch of these structs behind an interface. Please declare two interfaces:

- A C FFI that allows read access to the data structure(s)
- Classes/module in a language of your choice that wraps the C FFI and presents a nice, high level abstraction to the data structure(s)

Note, you do not need to write the implementation of these interfaces, just their declaration. The interface should allow:

- Loading of data
- Access to data values, including subsets of the full dataset
- Access to metadata (units, names, etc)
- Rapid seeking/searching through data
- Anything else you might think is useful for manipulation of these types of datasets

Resampling

Assume you have a log like log 1 (the ultrasonic amplitude). The log was sampled in time at varying velocities, and in multiple directions.

Write a method/class in a compiled (I'll accept bytecode for the JVM and the .NET CIL as being compiled) language of your choice that takes a dataset and resamples a subset of it into a form that can be used for display against a depth axis.

The output should:

- Have floating point values (the gradient colour mapping happens later)
- Be evenly gridded
- Be resampled to a reasonable number of points that can be displayed on a modern monitor
- Prioritise display of the most recently acquired data (as in if there is multiple samples for the same depth, the more recently acquired data should have a higher z order/be in front).

The inputs to this implementation should be:

- The dataset
- The depth and azimuth range of interest
- The number of points in each dimension to resample to
- Anything else that might be useful...

Outputs

Please send through a compressed Git repository containing all of the code, notes, documentation, tests and other artifacts you produced while doing these tasks.