LiDAR Captured Cliff Data Analysis

ROBERTO DYKE, YUKUN LAI, & TRISTRAM HALES

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

INTRODUCTION

With the assistance of Computer Science department, the Earth Sciences hoped to prompt some initial research into the development of a computer-aided system that would help analyse cliff faces in a faster, safer manner.

Field geographers would measure the height of alternating layers of light and dark rock – indicating distinctive weather patterns – thus allowing them to gauge former physical activities in that area.

Due to subtle changes in how the Earth orbits the sun an ice age will occur for 10,000 years, over that time period a dark, fragile layer of sediment (highlighted in green) develops. These are followed by interglacial periods, which last about 100,000 years, and deposit lighter thicker sediment layers due to the longer time period (areas between the green lines).

So a cliff face gives us – effectively – a cross-section of the Earth's glacial-interglacial cycles over millions of years!

Method

To analyse the cliff face we planned on capturing a virtual 3D model of it using a device called a LiDAR scanner. Once captured, the point cloud was then transferred to a computer for refinement and further processing.

We decided to use the programming language C++ to process the point cloud – on account of its adequate speed, and library support.

We primarily used PCL (Point Cloud Library) to help us process the information; the library provided some basic tools for reading, storing, visualising, and measuring parts of the point cloud. We also used the Boost library, which provided a robust implementation of a vital path-finding algorithm.

Development

We started with the very basics of PCL by initially attempting to simply load a point cloud file into a barebones application. Unfortunately the file we were provided with was not in the correct format, so we created a simple program to convert the data to the required structure – further development could now commence.

We then attempted to take the restructured cliff data and compute some local descriptors, which are basics summaries of a point based on the surrounding environment (the neighbouring points). We planned to use these descriptors to identify some of the key features of the cliff face.

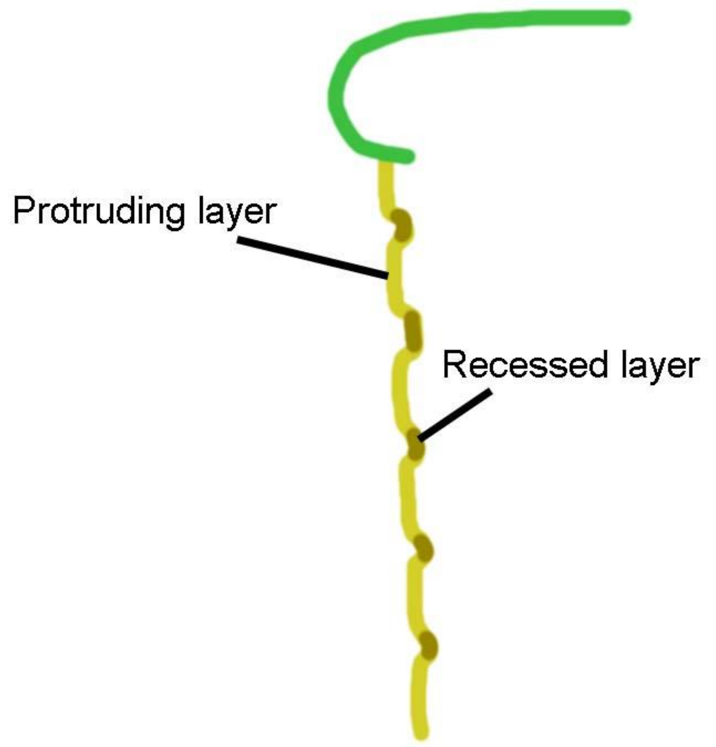
Layer Identification

To allow the computer to be able to identify the individual layers we had to devise a way of describing the layers. This description was split up into two parts, namely: local features and global optimisation. In this case we used local features to describe the characteristics of each individual point. Global optimisation was then used applied to the point cloud's local features to find a set of points that best match our criteria for the characteristics.

Local Features

Geometry Based Edge Feature

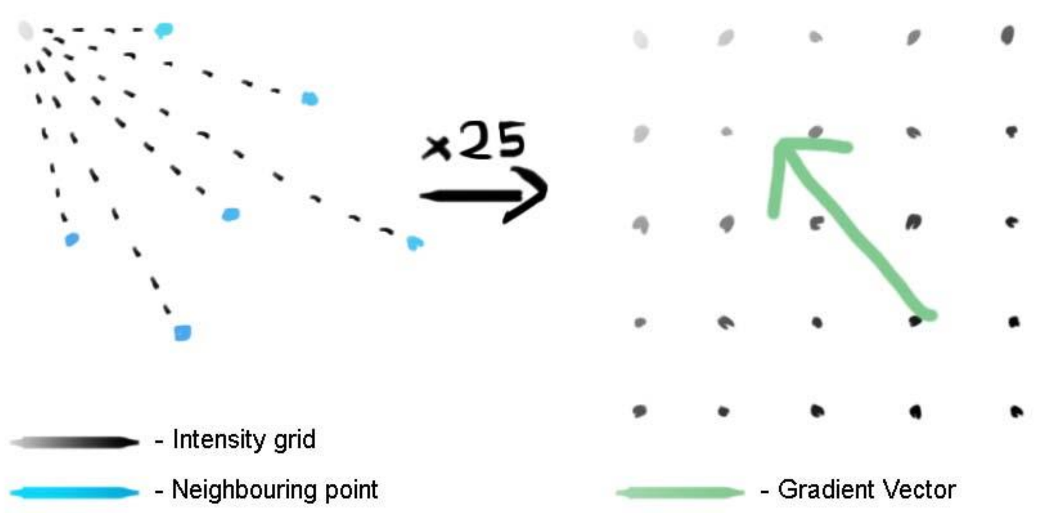
If we were to take a cross section of the cliff face (illustrated to the right) we would find that we could visibly see that the thick layers of sediment are likely to be protruding from the cliff face, while the thinner layers would recess behind the cliff face's regular surface.



By estimating the flatness of the area surrounding each point, we could determine if a point was on an edge between two layers.

To measure the flatness of an area we approximated a flat surface that best fit in with the neighbouring points, we then calculated summed the distance between each of the neighbouring points and the flat surface. A large total distance indicated that the surface was uneven, while a short total distance indicated a flat surface.

Image Based Edge Feature



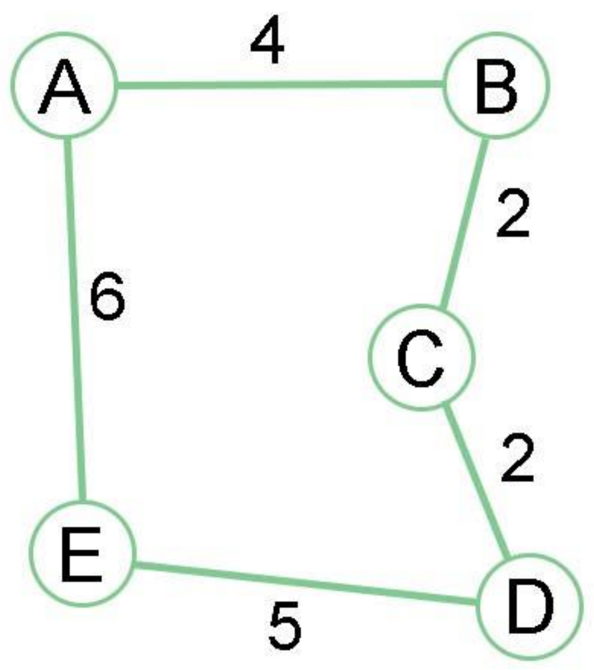
Large changes in gradient were determined to be edges, while small changes across the intensity grid were not.

By identifying the direction and strength of the gradient (a gradual change in contrast between dark to light) we could determine whether or not a point was on an edge as well as the direction the edge went in.

To find the angle and intensity of the gradient of colour we created a 2D canvas of the neighbouring points (as shown to the left). We then identified the intensity gradient by applying convolution masks to the canvas. From the gradient obtained we can calculate a vector for the direction of the gradient.

Global Optimisation

In the case of our problem we were able to present our optimisation problem in the form of a weighted graph, in which the edges represent the individual points in the point cloud and the vertices connections to neighbouring edges.



By using the two local features together we were able to find how well the characteristics of a point on the cliff face matched up with our criteria, we used this coherence value as a representation for cost. We then attempted to find the lowest cost path by jumping from point-to-point between the two red lines. After that we marked the path taken with a green line, producing the following results.

Conclusion

By the end of the 2 months we managed to successfully create the local features and global optimisation algorithms described above. From these we managed to identify most of the prominent edges between sediment layers (as shown earlier).

We unfortunately did not get as high of a success rate at identifying some of the more subtle edges. This was partially due to our use of a low-density point cloud and insufficiently optimised parameters. If we had captured a finer model of the cliff face and more time to develop optimisation tools we may have been able to identify more of the layers.