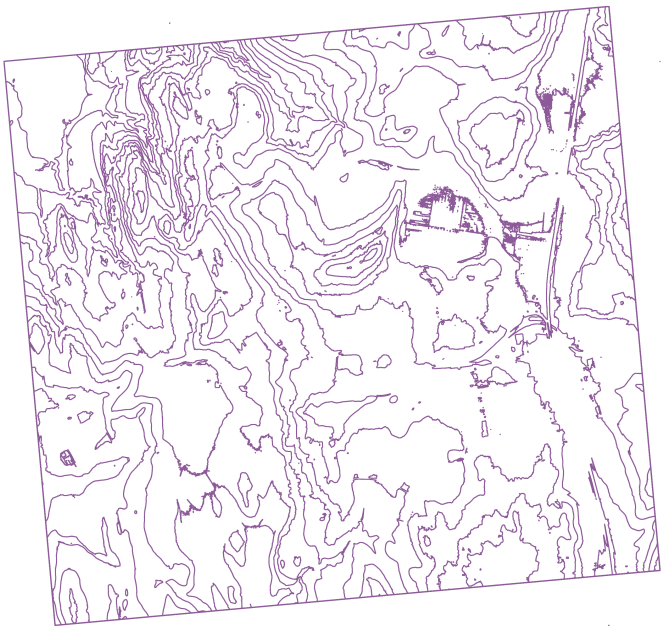
INF205 Project report

Creating contour maps from raster heightmaps

Esther and Trygve 7. May 2024



Introduction The source code is available on https://gitlab.com/Trygve/contour-creator The branch 7-5-2024 contains the source code as it was on the day of the deadline. Becouse of the 90/90 rule we had to focus on the core functionality and unfortunatly we have a few bugs and performance issues left. If we fix these issues they will be available on the main branch. # Functionality Our program uses the marching squares algorithm to create a vector contour map from a raster heightmap (DTM).
It outputs a geojson file that can be read by gis software. https://mapshaper.org/ is a simple website where you can view geojson files.

Instructions on running the program is available in the README.md file # data structures and input/output

The HeightMap class stores the heightmap as a array on the heap. It was not

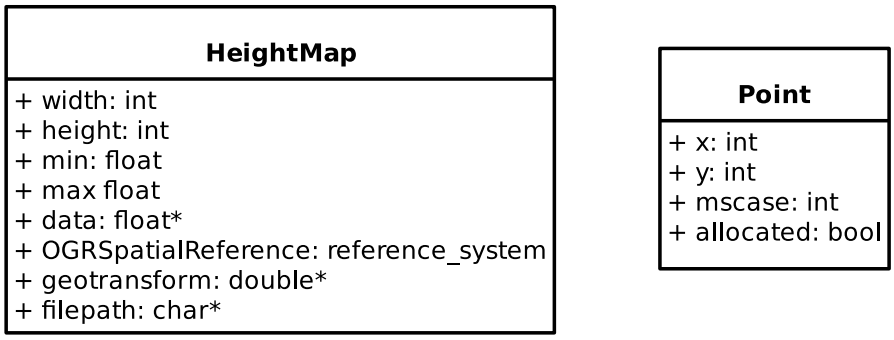


Figure 1: ER diagram

nececarry to copy or move this class, so we did not implement copy or move constructors/assignment operators.

The Point class represents a point in a line. The points are stored in vectors, wich are again stored in a big vector containing all the lines.
The `produce_cellmap` function finds the points for a contour at a given elevation. The `create_lines` uses `produce_cellmap` the get the points and then sorts them into contiguous lines before returning a `std::vector<std::vector<Point>>` thats used by the function `write_output_file` to write them to a geojson file.

Performance

	1	2	3	4	5	Avarage
12 threads	7,03	6,74	6,88	6,86	6,85	6,872
1 thread	21,48	21,27	20,82	20,35	20,81	20,946
reference	4,5	5,61	5,24	4,81	5,02	5,036

	slowdown
ours/gdal	1,36
singlethreaded/multithreaded	3,05

Our program is 1,36 times slower than gdal_contour wich is bad considering ours is multithreaded and gdals is not. On the other hand gdals is has 11 years of development.

The running on 12 threads yields a nice speedup 3,05 times.

The program assumes the whole heightmap can fit in memory, this is not a problem for heightmaps from hoydedata.no as the maximum tile size is about 700mb.

Functionality of special interest

Trygve

I can created a blur method that performs a simple box blur on the whole heigtmap. This is usefull to smooth out noise and create a more readable output. I wanted to implement a gaussion blur, and the code is ready for that, but in the end we became short on time. The blur is activated by passing the `--blur` flag

Esther

Calculated some statistical estimates for the average, variance and standard deviation for the pixels in the image. Also tried to calculate the steepness around each pixel. However, we didn't manage to get the Gdal to work for this. Statistics output is activated by passing the `--stats` flag