Lab Assignment #5 – Introduction to Lidar

1. Logistics

Date assigned:	Monday, November 25, 2024
Date due:	Monday, January 13, 2025 (via Moodle)
Points:	100 points

In this lab, you will be introduced to Lidar data using the open source CloudCompare software. We will look at a classified point cloud of the Potsdam area, and learn how to export the data into different formats for further analysis.

2. Lidar data and software

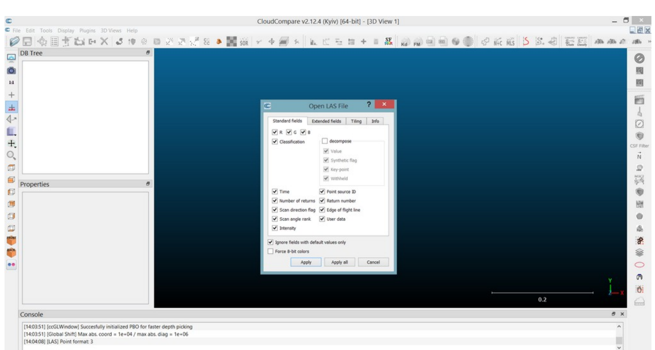
Lidar stands for ‘light detection and ranging’, and involves shooting discrete laser beams at a surface of interest. The reflected energy is then recorded, giving a view of the surface. There are many kinds of lidar data serving many purposes – some you have likely used yourself. For example, the Xbox Kinect uses lidar, as do many cars which are partially self-driving.

There are several types of lidars – from photon counting to full waveform – which are used in many different **spectral bands**. For example, a some lidar can penetrate into shallow waters, and some will not pass much beyond the water’s surface. In this lab, we will use a simple airborne lidar data set over the Campus Golm.

There are several open source packages used for viewing and working with lidar data. Two of the most common are Displaz (https://github.com/c42f/displaz) and CloudCompare (https://www.danielgm.net/cc/). As CloudCompare is easier to install on most systems, we will focus on this lab on **CloudCompare**.

The data we will use today is primarily from an airborne lidar flight in May 2018. A secondary flight using a smaller drone was done on the same day to capture the RGB (color) data to complement the ranging (distance) information from the airborne lidar flight.

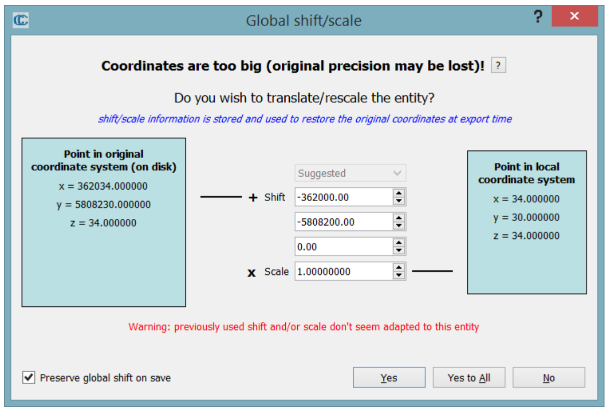
3. Visualizing Lidar Data



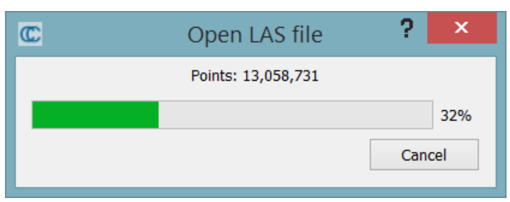
Dialog for loading data into CloudCompare.

There are several different data attributes in our lidar file – not just an xyz (lat/lon/height) coordinate, but also color and other information. Let’s load all of the attributes in and see if we can visualize them.

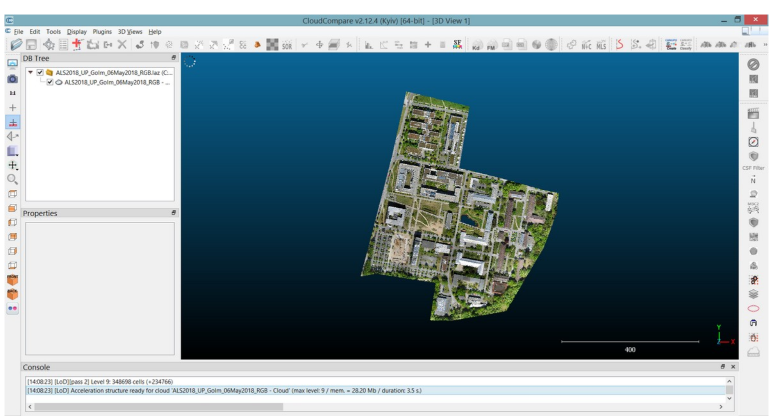
To preserve coordinate accuracy, you might see this dialog pop up:



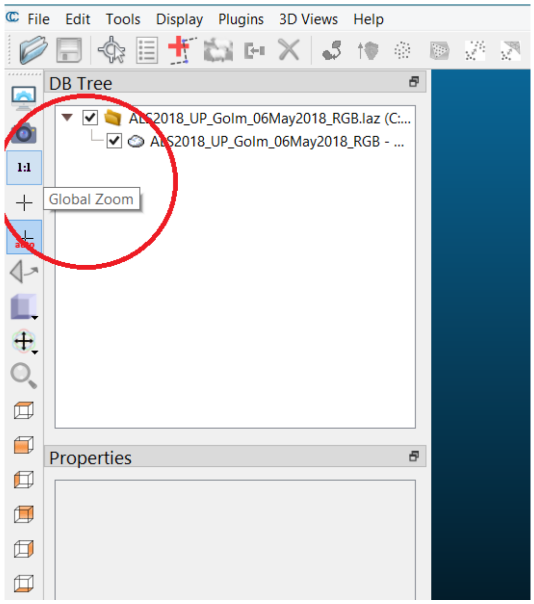
This simply shifts the coordinate system by a fixed amount to keep the numbers displayed in CloudCompare manageable. However, this sometimes gives errors when exporting back to QGIS! To make things simpler on that end, hit ‘No’ here. You should now see the data being loaded – all 13 million points! **NOTE: If you will load in multiple data sets, it is better to click ‘Yes’ here, to make sure that you shift all your data to a common reference in CloudCompare.**



The default view uses *color information* stored in the file to display the lidar points:

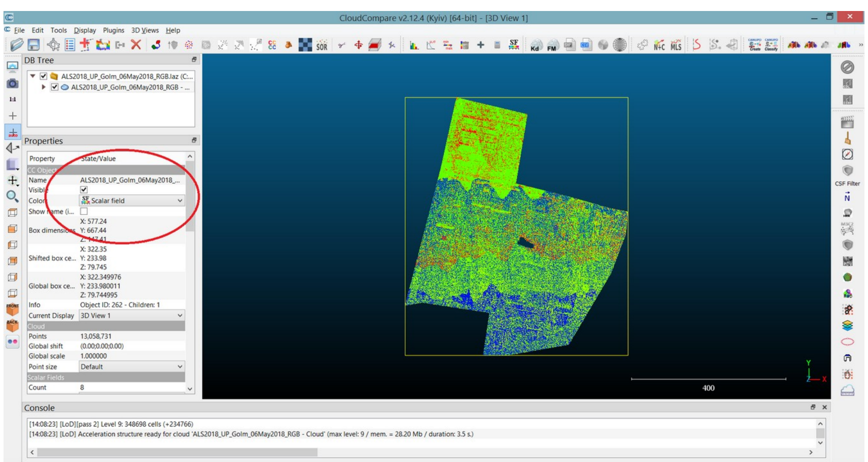


We can move around the point cloud with standard mouse movements – click and drag, use the mouse to scroll to zoom in and out, etc. If you ever get lost/too far zoomed in, you can use the '1:1' button to
zoom back out to the default view:

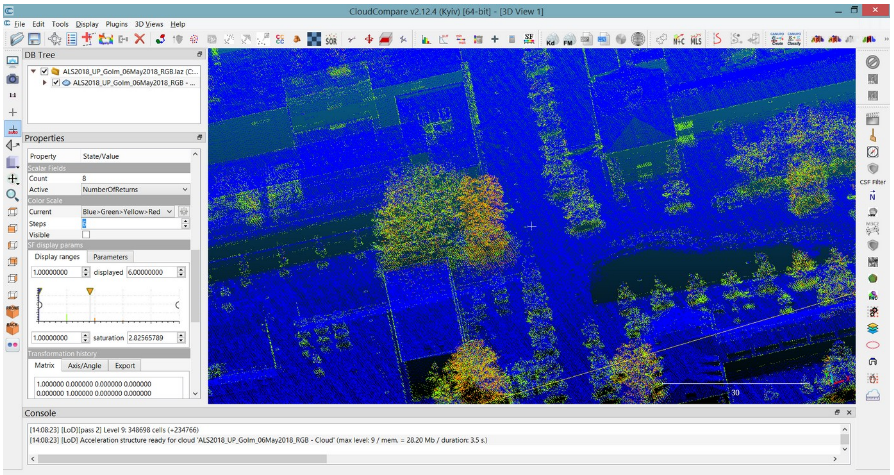


4. Examining Different Attributes

If you change your point-cloud display from 'RGB' to 'Scalar field', you can examine other attributes within the file:



In the case of this file, we have several different attributes to look at. Let's take a look at 'Number of Returns', which is a measure for how many photons were captured over a given location.

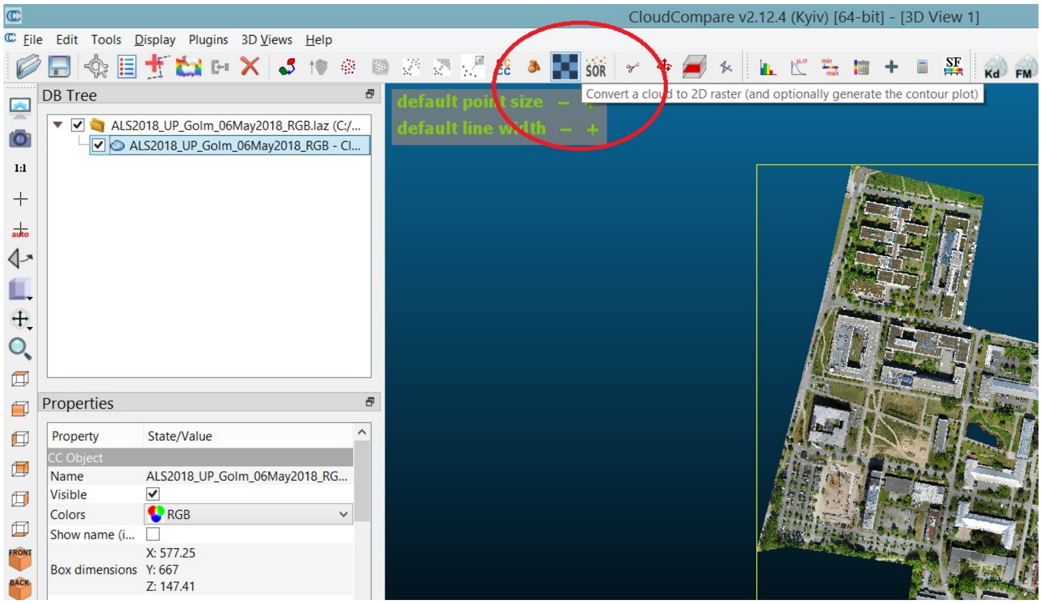


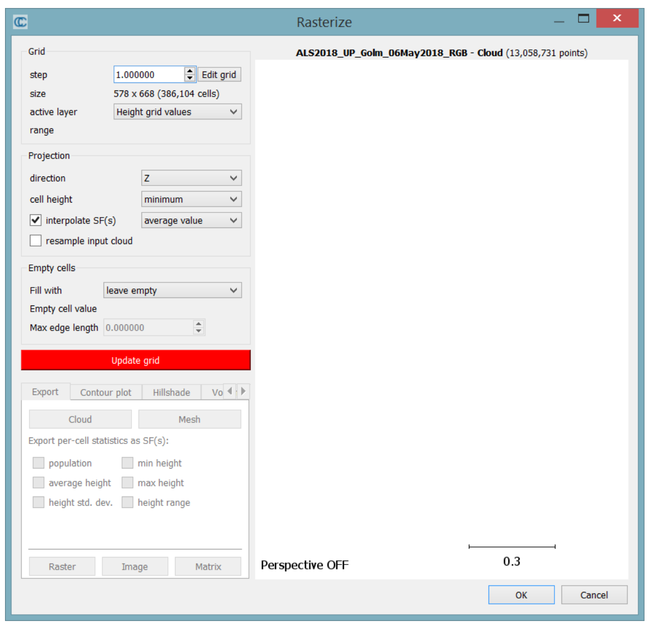
If we change the color scale, it becomes clear that there are many more returns over vegetation than over flat surfaces! Why might that be?

Our lidar file has one additional attribute that was added during post-processing: **Classification**. This is created using a point-cloud classification algorithm (Hint: We'll be looking at image-based classification methods in much more depth in the next few weeks!), and mostly relies on the spatial characteristics of each surface.

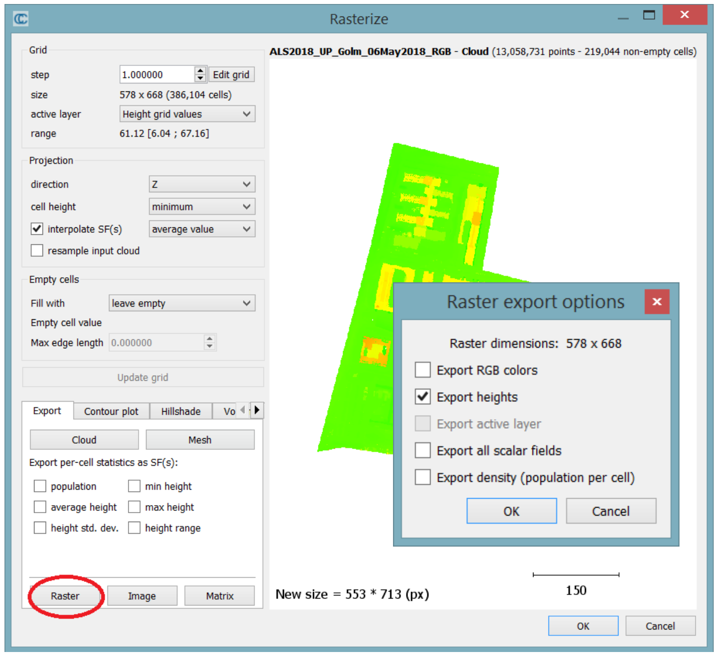
5. Rasterizing Lidar Data

Lidar data has some great advantages over rasterized (gridded) data, but is also significantly more difficult to work with for many simple tasks. One common task for post-processing lidar data is **rasterization**, which is the process of taking the un-gridded point cloud data and creating a simple (flat) raster. Let's try this in CloudCompare:



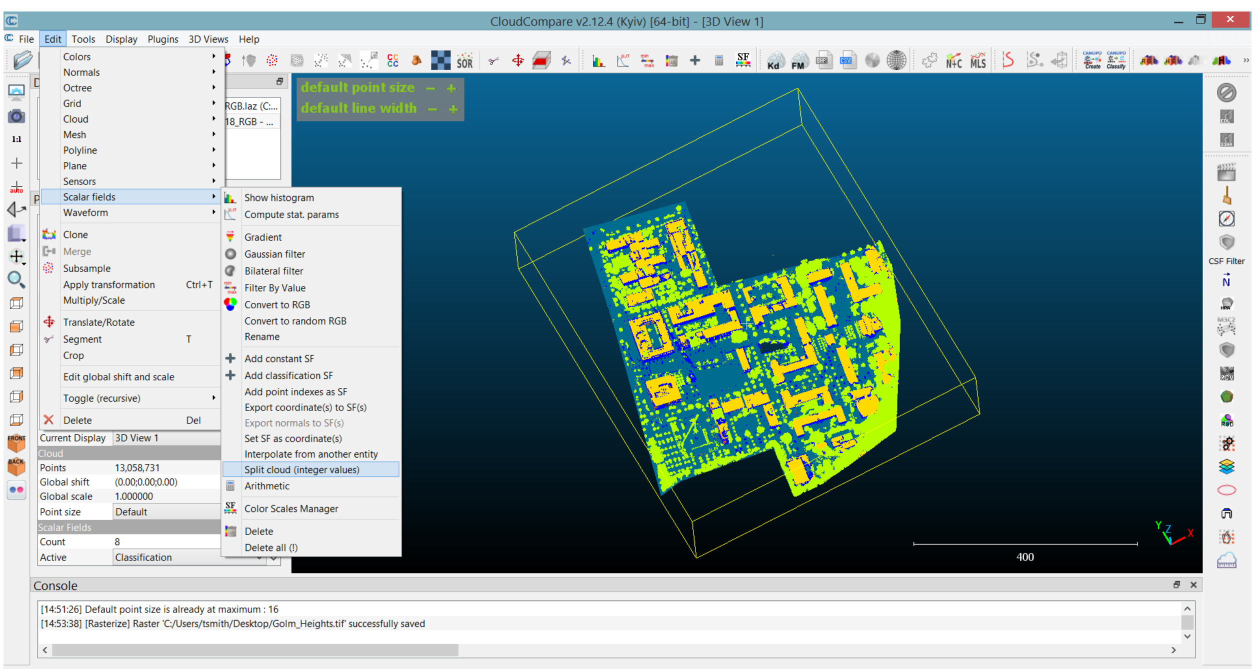


If we hit ‘Update Grid’ we can see what our 1x1 meter grid will look like. We can then choose ‘Raster’ below to export a simple grid of our height values:



This will save a simple raster of all of the heights in our file. This is the equivalent of a Digital Surface Model (not a Terrain Model!).

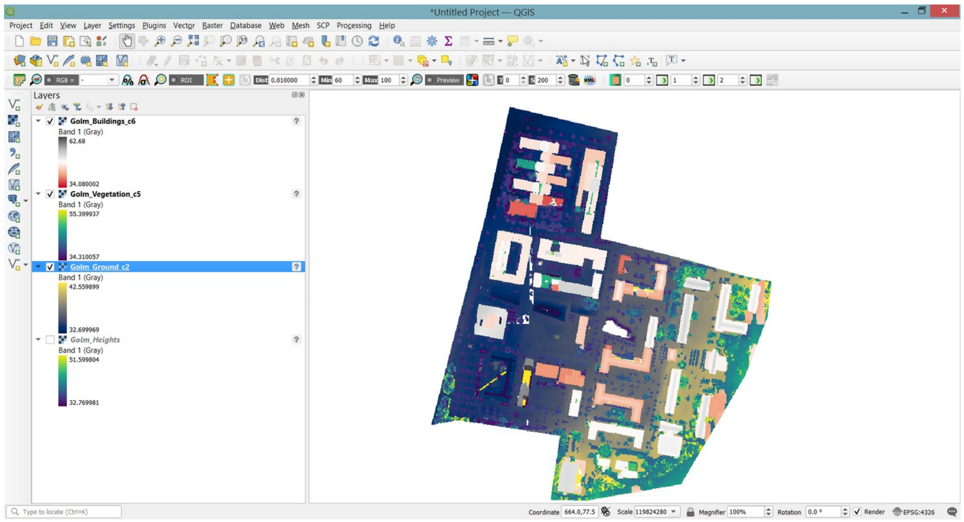
This, however, doesn't make use of our *Classification* field – it many cases you want to find out, for example, how tall your buildings or plants are above the ground. To do this, we first need to *split our point cloud*. We can do this using the 'Split cloud (integer values)' tool:



Try turning those classes on and off again – what does each one represent?

We can export those classified points as their own grid using the same procedure. Let's take the ground, vegetation, and building layers each as a separate raster file. We can then open all four images in QGIS.

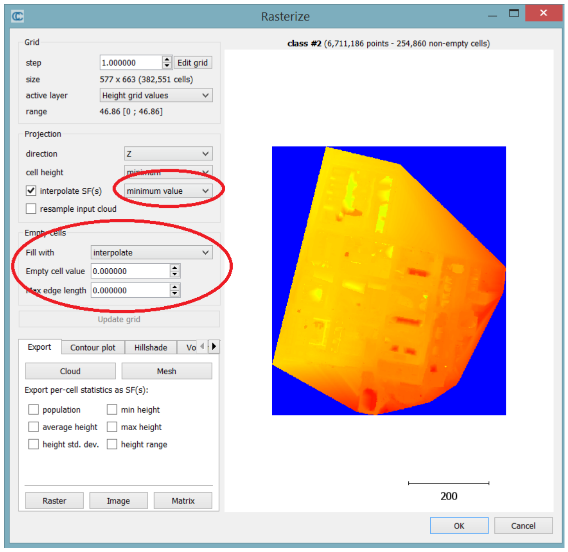
My output looks like this:



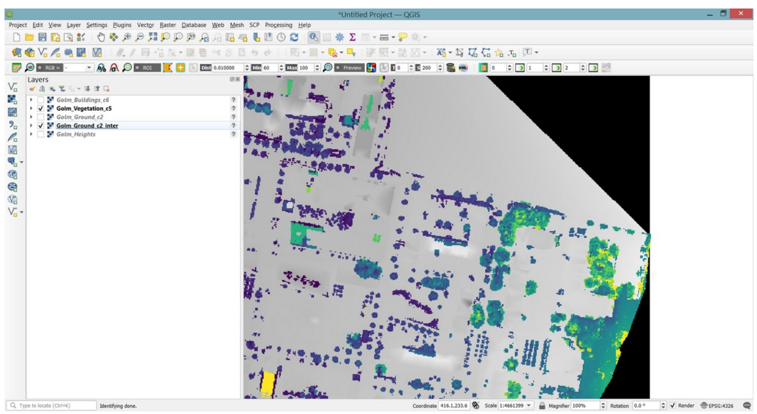
Let's do one more step – create an interpolated rough ground surface with which to compare our vegetation and building heights. If we want to accurately measure the height of our vegetation above

the ground, we need to know the height of the ground! There are many sophisticated ways to do this, which will be further explored in the advanced lidar course (Summer Semester). We can do a really quick and dirty calculation for our area, however, using a simple **interpolation**.

As before, let's export our *ground class*, but this time we want to fill in the holes with interpolation:



We can now look at vegetation heights above the ground, rather than only looking at the rasterized vegetation data!



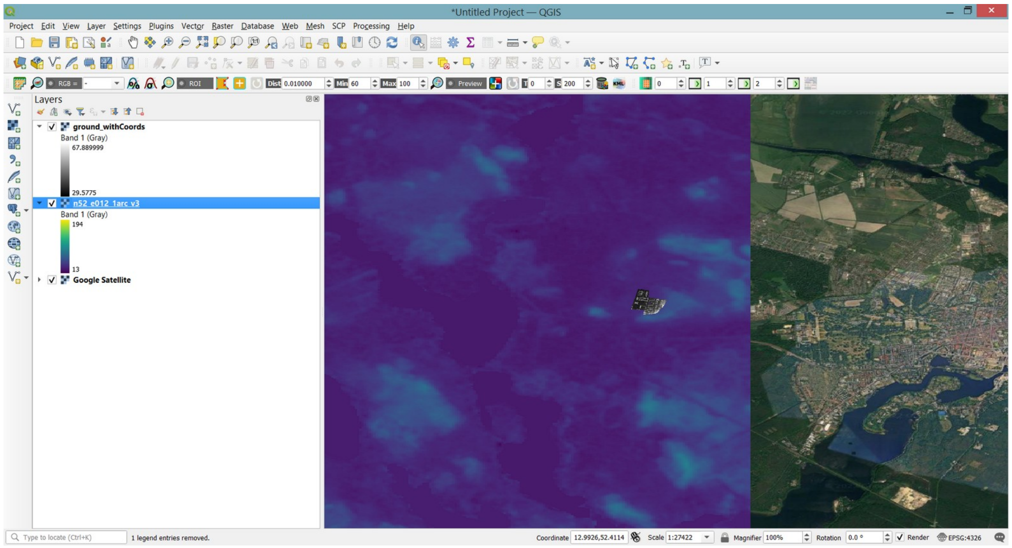
Question 1: Create a map in QGIS showing vegetation heights above the interpolated ground surface. Where is the tallest vegetation? Where are there clear errors? (25 points).

Question 2: Create a map in QGIS showing building heights above the interpolated ground surface. Do these building heights seem accurate? Where are they better or worse? (25 points).

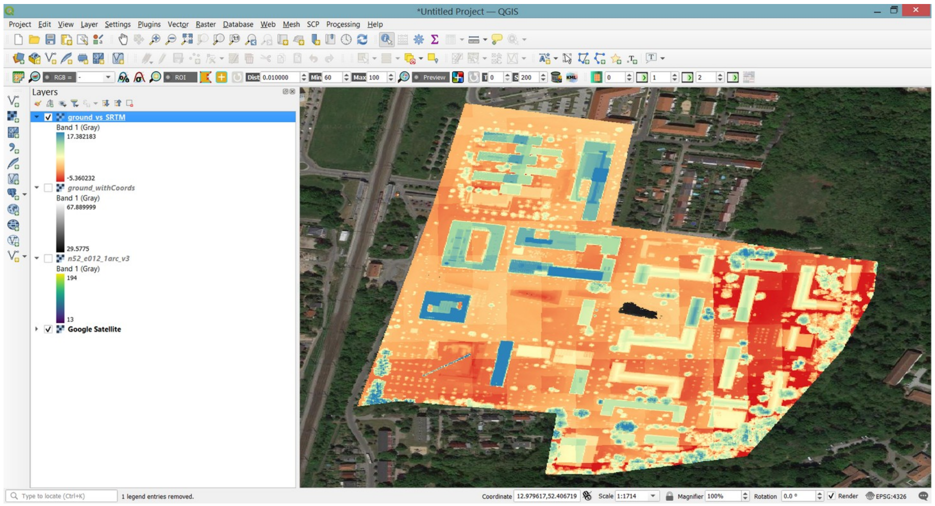
Question 3: Repeat your analysis using a larger grid spacing – how do the estimated vegetation heights change? Please provide a map output here. (25 points).

6. Comparison to Global-Scale DEM Data

Using either SciHub or EarthExplorer, we can find a high-resolution DEM of the Golm area. I downloaded the SRTM 1-arcsecond DEM (which is also available on Moodle).



It is clear that these data are at a very different scale! Let’s do a direct comparison – we can use the Raster Calculator to create a difference between our lidar data and the large-scale DEM. Depending on your interpolation scheme, it might look something like this:



<u>Question 4:</u> Why do you think the Lidar and SRTM data are so different? Which data set is more trustworthy? Do the errors get bigger or smaller if you use a larger grid (e.g., 10 meters?) Please provide maps showing the differences between your terrain models at different grid resolutions, with useful colorbars, etc. (25 points).