GEM 511: Advanced GIS for Environmental Management

Paul D. Pickell

2024-01-07

Contents

Welcome

These are the course materials for GEM 511 in the Master of Geomatics for Environmental Management program (MGEM) at the University of British Columbia (UBC). These Open Educational Resources (OER) were developed to foster the Geomatics Community of Practice that is hosted by the Faculty of Forestry at UBC.

These materials are primarily lab assignments that students enrolled in GEM 511 will complete and submit for credit in the program. Note that much of the data referenced are either public datasets or otherwise only available to students enrolled in the course for credit. Deliverables for these assignments are submitted through the UBC learning management system and only students enrolled in the course may submit these assignments for credit.

How to use these resources

Each "chapter" is a standalone lab assignment designed to be completed over one or two weeks.

Students enrolled in GEM 511 will submit all deliverables through the course management system at UBC for credit and should consult the schedule and deadlines posted there. The casual user can still complete the tutorials step-by-step, but the data that are not alreadyh publicly available are not hosted on this website and therefore you will not have access to them.

How to get involved

Because this is an open project, we highly encourage contributions from the community. The content is hosted on our GitHub repository and from there you can open an issue or start a discussion. Feel free to open an issue for any typos, factual discrepancies, bugs, or topics you want to see. We are always looking for great Canadian case studies to share! You can also fork our GitHub repository to explore the source code and take the content offline.

Chapter 1

Spatial interpolation and visualization of LiDAR

Written by Paul Pickell

Lab Overview

The aim of this lab is to use LiDAR data from the University of British Columbia Malcolm Knapp Research Forest (MKRF) to create Digital Elevation Model (DEM) using a variety of spatial interpolation approaches. We will investigate how these methods compare to one another, and explore their strengths and weaknesses. Additionally, we will explore point cloud manipulation and visualization using both QGIS and ArcGIS Pro.

Learning Objectives

- Interpret metadata for a LiDAR acquisition and point cloud
- Manipulate a LiDAR point cloud with a variety of different tools
- Generate and evaluate DEMs from a LiDAR point cloud using different terrain spatial interpolation approaches
- Create maps and 3D visualizations of point clouds and interpolated surfaces

---

Deliverables

Lab report with the following specification:

6 pages maximum PDF including figures, tables and references (3 points). Single-spaced, 12-point Times New Roman font (1 point). All figures and tables should be properly captioned (1 point).

Introduction should address the following questions and requirements (10 points):

- Describe the different spatial interpolation methods that were tested.
- What are the parameters of each corresponding tool in ArcGIS Pro and how do they impact the algorithm?
- Reference to at least 3 peer review sources.

Methods should address the following requirements (10 points):

- Brief study area description and study area map (Note: there is an orthophoto of the research forest that was acquired at the same time as the LiDAR point cloud that you can reference and map).
- Outline all primary steps included in the lab (no need to include exact ArcGIS Pro tool parameters).
- Justify the use of specific methodological choices indicated in the lab.

Results should address the following questions and requirements (15 points):

- Statistics of the LiDAR point cloud in the AOI. What is the true, measured range of elevation by LiDAR within the AOI?
- Table of summary statistics of the binned DEM.
- A panel of maps that symbolize the DEMs produced and the difference rasters that were produced by the comparison to the binning approach.
- Tables that show zonal statistics of the difference rasters across elevation and slope ranges. Which zones has the largest differences?
- Additional maps and/or 3D scenes that illustrate any local observations you made about the magnitude of differences or comparisons between interpolation algorithms

Discussion should address the following questions and requirements (10 points):

- How did each spatial interpolation algorithm perform relative to the binned DEM and the raw point cloud?
- Interpret the results using your observations of the point cloud and other available data, statistics and metadata.
- Strengths and limitations of each spatial interpolation algorithm in this study area.
- Which spatial interpolation algorithm would you recommend the research forest use? Defend and justify your choice.
- Reference to at least 3 peer review sources (can be the same sources as introduction).

Data

We will be working with LiDAR data collected over the UBC Malcolm Knapp Research Forest (MKRF) in Maple Ridge, British Columbia. These data are publicly available from the Master of Geomatics for Environmental Management (MGEM) Data Store and the instructions for accessing these data are given in the tasks below.

Task 1: Preprocess LiDAR data in PDAL

Point cloud data are large. For example, the point cloud collection that we will be working with contains 1,671,233,402 points! Typically, we should not interact with point cloud data in a desktop environment until we have to. Graphical user interfaces like QGIS and ArcGIS Pro introduce a large amount of computational overhead when working with point cloud data and these software are more suited for visualizing the data rather than processing them. Point cloud data are much more commonly hosted on remote servers nowadays, in cloud-optimized formats, and available for on-demand and query-ready streaming.

In this task, we will explore large LiDAR acquisitions that have been collected at the University of British Columbia (UBC) Malcolm Knapp Research Forest (MKRF). LiDAR collections are typically tiled to reduce the overhead with transacting with individual files. We are only going to process a handful of tiles, but we need to first grab the right tiles for our area of interest (AOI).

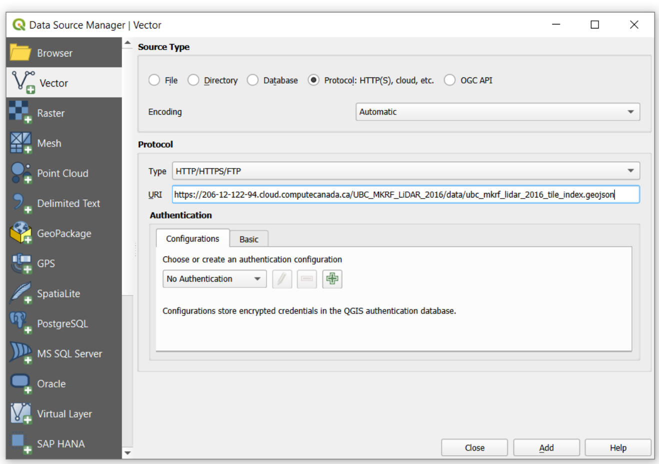
Step 1: Navigate to the MGEM Data Store and inspect the UBC MKRF 2016 LiDAR collection: https://206-12-122-94.cloud.computecanada.ca/UBC_MKRF_LiDAR_2016/

At the top, you will see some generic metadata for the collection. Below that, you will see a web map showing the tiles. If you click on one of the tiles, it gives the direct download universal resource locator (URL) and file size. If you scroll down, all the tiles are listed in a table along with a metadata file that provides some more specific information about each individual tile-file.

Step 2: Click to open one of the metadata txt files in your browser. This is json-formatted metadata for the associated LiDAR tile. Scrolling through it you will find summary statistics over a number of different dimensions. What attributes describe these LiDAR data?

Suppose that we have some AOI within the research forest that we need to retrieve the LiDAR data from. How could we figure out which tiles we need without downloading all of them from the server? In most cases, you should have a polygon tile index available to assist you with this task. The tile index is another form of metadata, albeit spatial metadata.

Step 3: Right-click on the **tile_index.geojson** file at the top of the file listing and select "Copy Link". Open QGIS and click the "Open Data Source Manager" button. On the left of the Data Source Manager dialogue, select "Vector", toggle on "Protocol: HTTP(S), cloud, etc", then paste the URL you copied into the "URI" field (uniform resource identifier). "Add" the layer to your map view then "Close" the dialogue and inspect the result.



A nice feature about QGIS is that it supports the ability to read any openly-specified geospatial file directly from a remote source. ArcGIS Pro only allows you to read some sources published on compatible remote databases (e.g., ArcGIS enterprise geodatabase or PostgreSQL) or from layers published on ArcGIS Online.

Now that we have spatial tile metadata, we can perform spatial intersection to find the right tiles. Our AOI is going to be the following longitude-latitude bounding box:

Lower left (LL): -122.55275, 49.32325

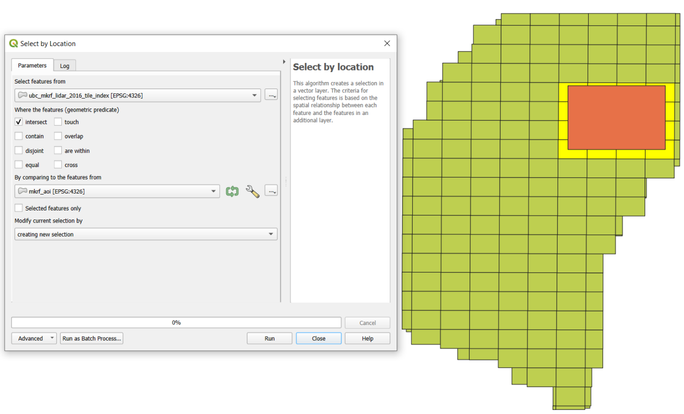
Upper right (UR): -122.52506, 49.34135

Step 4: Open a notepad text editor and convert this bounding box into a geojson polygon feature with the following syntax:

```
{
  "type": "Polygon",
  "coordinates": [
    [
      [LL_longitude, LL_latitude],
      [UR_longitude, LL_latitude],
      [UR_longitude, UR_latitude],
      [LL_longitude, UR_latitude],
      [LL_longitude, LL_latitude]
    ]
  ]
}
```

Replace with the correct longitude/latitude values. Note that this creates a square polygon and the fifth coordinate is the same as the first, which topologically encloses the polygon. Save the geojson in your QGIS project folder as "mkrf_aoi.geojson" then open the file in QGIS. You should see that our AOI spans 16 total tiles.

Step 4: Click the "Select by Location" tool and "Select features from" the **ubc_mkrf_lidar_2016_tile_index** layer by intersecting with the **mkrf_aoi** you just made. Now the intersecting tiles are selected. Turn off **mkrf_aoi** layer so that you can see the selection.



Step 5: Click the "Identify Features" tool and click on one of the selected tiles, which will highlight it in red and open the attributes for the polygon. Expand "ubc_mkrf_lidar_2016_tile_index", "url", "(Actions)", and "url". Click the hyperlinked URL to download the tile to your QGIS project folder. Repeat this step for all 16 tiles.

Step 6: Add all of the downloaded tiles to your QGIS map canvas. The default symbology is the classification attribute, but only ground returns have been classified for the research forest.

However, the tiles are all still stored in separate files and some portions are outside our AOI. So next, we are going to filter, merge, and crop the point cloud, but we are going to do this outside of QGIS because it will be faster and more reliable. We are going to use the Point Data Abstraction Library (PDAL) command line utility to perform this processing. You can read more about the extensive PDAL functionality here: https://pdal.io/ Note that many of the functions we are going to use with PDAL are available as tools through QGIS, but you have less control over the options and parameters in QGIS.

Step 7: Open the OSGeo4W shell and navigate to your QGIS project folder where your LiDAR data are located. For example, `cd C:\users\paul\Documents\QGIS\mkrf_lidar`. Type the command `pdal --version` to make sure that PDAL was installed with your current version of QGIS. If a version number is returned in the console window, then continue to the next step, otherwise ask your instructor to help you install PDAL using the OSGeo4W installer.

PDAL can run functions in two ways. First, as subcommands, much like you have used with GDAL in prior labs. For example, `pdal merge [filename1.laz] [filename2.las] output.copc.laz` will use the `merge` subcommand and the file names that follow to merge many different las/laz/copc files together into the named output. Subcommands are generally good for small or incidental tasks like converting a file format or re-projecting data, but if you want to apply a more complex workflow then you should use a pipeline.

Pipelines are JSON-formatted files that give PDAL a set of instructions for reading, processing and writing point cloud data. With pipelines, you can define every step of the processing that you want and you can specify the finest level of detail at every stage. Pipelines are executed linearly, so we read the instructions from top-down. Below is an example of a pipeline that we are going to use, which is described in more detail below:

```
{
    "pipeline":
        [
            "AQ11.copc.laz",
            "AQ12.copc.laz",
            "AQ13.copc.laz",
```

```
            "AQ14.copc.laz",
            "AR11.copc.laz",
            "AR12.copc.laz",
            "AR13.copc.laz",
            "AR14.copc.laz",
            "AS11.copc.laz",
            "AS12.copc.laz",
            "AS13.copc.laz",
            "AS14.copc.laz",
            "AT11.copc.laz",
            "AT12.copc.laz",
            "AT13.copc.laz",
            "AT14.copc.laz",
            {
                "type":"filters.range",
                "limits":"Classification[2:2]"
            },
            {
                "type":"filters.crop",
                "bounds":"([-122.55275,-122.52506],[49.32325,49.34135])",
                "a_srs":"EPSG:4326"
            },
            {
                "type": "filters.merge"
            },
            {
                "type":"writers.las",
                "filename":"mkrf_aoi_lidar.las"
            }
        ]
}
```

This pipeline will take all of our input tiles and filter, crop, merge, and write them out to a new file called "mkrf_aoi_lidar.las".

```
            {
                "type":"filters.range",
                "limits":"Classification[2:2]"
            },
```

This first stage applies a filter stage with a function called range. Basically, this filter is telling PDAL that we only want the points that are classified as ground returns `Classification[2:2]` where `2:2` indicates the range of values from 2 to 2, which is the ground return classification code. So only ground returns are passed to the next stage:

```
{
    "type":"filters.crop",
    "bounds":"([-122.55275,-122.52506],[49.32325,49.34135])",
    "a_srs":"EPSG:4326"
},
```

This next filter stage crops the ground returns from the previous stage using the bounds of our AOI. We need to specify the spatial reference system `a_srs` of these coordinates since the LiDAR data are in a projected coordinate system (EPSG:26910). So only ground returns that fall within our AOI are passed to the next stage:

```
{
    "type": "filters.merge"
},
```

This next filter stage merges all the ground returns in our AOI into a single stream, which is then passed to the last stage that writes it to an output file in LAS (uncompressed) format:

```
{
    "type":"writers.las",
    "filename":"mkrf_aoi_lidar.las"
}
```

Step 8: Copy the contents of the pipeline to a text editor and save the file as "process-lidar.json" in your QGIS project folder.

Step 9: Return to the OSGeo4W shell and run the pipeline using the following command: `pdal pipeline process-lidar.json`. It may take several minutes for this step to complete, but in the end you should have a file called "mkrf_aoi_lidar.las" in your QGIS project folder. Drag it into QGIS and inspect it.

Task 2: Visualize LiDAR data in QGIS

The default symbology of LiDAR data in QGIS will be the classification, but we only have ground returns in our file, so everything will appear brown. Try symbolizing some of the other attributes like Intensity, ReturnNumber, and ScanAngleRank.