

Extracting Data from Shape Files

Kazi Hasan Zubaer

1105024

April 12, 2016

Installing QGIS

QGIS is a software to handle geographic information. The *.shp* file format is a popular file format to represent geographic information and we got road data from Dhaka City Corporation in shape format. To handle the files I used QGIS. It is available on all platforms but works well on Ubuntu. This documentation uses Ubuntu platform. We can install the software directly from the software center for Ubuntu.

Configuring QGIS

We will need *MMQGIS* plugin which doesn't come by default with QGIS. To add *MMQGIS* plugin, simply go to plugins → Manage and Install Plugins → Get More. Then search and install *MMQGIS*. To make it work you will need python *xlw* and *xlrd* support. To install these, the easiest way is to install *pip* on your Ubuntu system and then by using *pip* you will be able to install *xlw* and *xlrd*.

Loading and Extracting Node from Shape File

- Open QGIS and click on *Add Vector* button on the upper left corner. Browse to the shape file.

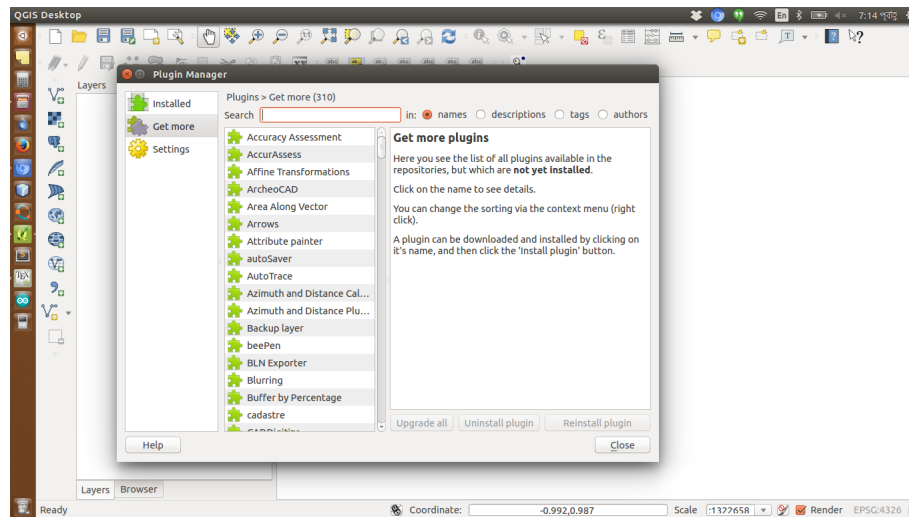


Figure 1: Installing new plugins

- After loading the shape file, choose Vector → Geometry Tools → Extract Nodes. Save the new information in a new shape file (that will be the default option when you will try to extract nodes).
- Close the old shape file.
- Right click on the newly created shape present on the left side of QGIS interface and choose *Open Attribute Table*. You will be able to see some information about the nodes.
- Choose Vector → Geometry Tools → Export/Add geometry columns. Open the attributes table again. You will be able to see new columns with coordinates of the nodes.
- Choose MMQGIS → Import/Export → Attributes Export to CSV File. Select ID, ROAD_TYPE, LENGTH, XCOORD, YCOORD and save the CSV file.

Using ShapeParser

Now we have got a CSV file. The CSV file contains the description of each node. A row represents a node. The row contains the road id the node

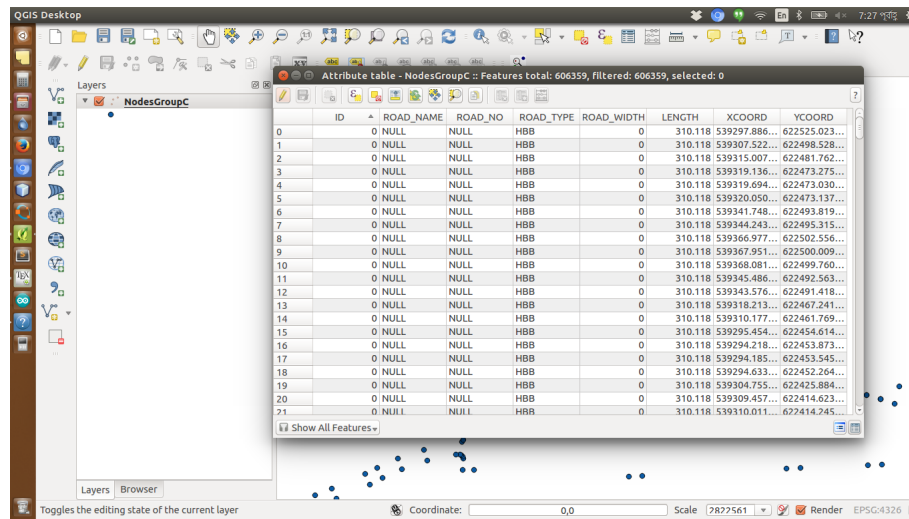


Figure 2: Attributes table

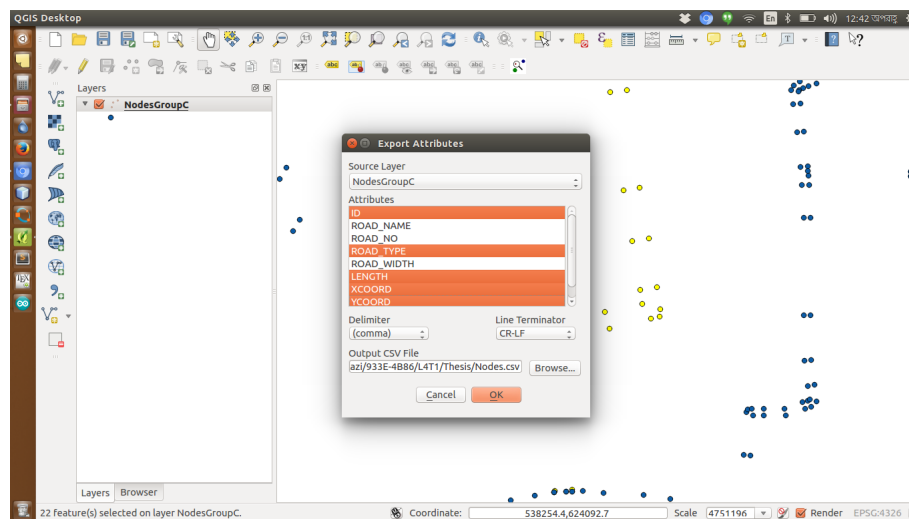


Figure 3: Exporting to CSV

belongs to, the length of that road, type of the road and coordinates of the node. It has nodes on a road on both sides in a cyclic order. In our simulator, we need nodes on one side of the road in order and the width of the road. Here I have followed the following algorithms to convert the information:

- To find the width, I have calculated the area of a node using *Shoelace theorem*. Then I have divided the area with length.
- To find points on one side, I have added points one by one. When the length consisted by the added points becomes close to the given length, I have stopped.

If we keep the CSV file in the project folder of *ShapeParser* and run the program, it will generate a text file with columns ROAD_ID,WIDTH,XCOORD,YCOORD.

The description of different classes of *ShapeParser* is given below:

Point Class

This class mainly represents a geometric point. We have member methods to find its magnitude and distance from another point.

Road Class

The road class has following member variables:

- *id*, the id of the road.
- *n*, number of points on original description (Points on both side of the road).
- *width* and *length*, width and length of the road.
- *ptsRaw*, point array to hold points on both sides of the road.
- *ptsFinal*, points on one side of the road.

The road class has the following member methods:

- *Road*, the constructor.

- *Calculate*, to calculate the width and ptsFinal.
- *getID*, *getWidth*, *getpoints* - , getter methods.

ShapeParser Class

The driver class which handles the input and output. Modify the name of the input and output files here according to your input and output file name. I have ignored roads with ROAD_TYPE *Katcha*. Include the roads if necessary by changing the corresponding part of this class.