End-to-End Point Cloud Processing using Python Wrapper

Ganesan A/L Patmanathan and Low Ming Guan, Electronics System Engineering, *Malaysia-Japan International Institute of Technology, UTMKL*

Abstract- 3D point cloud is beginning to create an accurate 3D model of the real environment. Digital application map of points in space that's turned into 3D models of nearly any item, and it's the beginning point for digital reality.[1] For visualization, planning, and personalization, 3D models based on point cloud data are being used in an increasing variety of industries such as buildings, factories, manufacturing facilities, crime/crash scenes, civil infrastructure, historical landmarks, cityscapes, and much more are all included on a big scale.[8] The data has the potential to alter and is already doing so the way buildings are built and maintained, increasing the precision of data available to everyone from architects to operations managers.

However, there is some differences in doing with the python program which making a graphical representation of information using visual elements known as Data visualization that represent in trends, outliers, and patterns in data.[1] Then, manipulated with the 3D point cloud datasets representing real-world shapes and 3D reconstruction techniques such as Photogrammetry.

Introduction

For more than 20 years, point cloud scanning has been utilised mostly in the mining sector to measure and compute the size of mine shafts or spoil piles. It is increasingly being utilised in the built environment to capture detailed layouts of land, buildings, and structures, as well as the MEP services included inside them. Because of the high degree of precision and breadth of information captured by a point cloud scan, it is frequently used to assist renovations, additions, and even new construction.[12] The features from 3D point clouds are a critical subject in computer vision and graphics because it has several applications, including 3D reconstruction, object segmentation and identification, robot navigation, and object grabbing. A variety of shape analysis issues, such as classification, part segmentation, and point matching, can benefit from geometric characteristics learned from an interaction of local points.

The features from 3D point clouds is a critical subject in computer vision and graphics because it has a wide range of applications, including 3D reconstruction, object segmentation and identification, robot navigation, and object grabbing. Early work on point cloud feature extraction relied mostly on hand-crafted features that ignored point cloud geometric relationships and lacked semantic context information. [13]

Objective

The major goal of this project is to tackle an industrial challenge of perceiving a large structure or construction before it is built in the actual world as a digitalized map of points in space that is then processed into 3D models of nearly any object.

Moreover, here we need to carry out the programming task on the can execute end-to-end process for 3D Point Cloud. By managing the point cloud of the model provided in the CloudComPy and build and automated processing program which could export and analyzed the point cloud.

Literature Review

In this section, various of "3D Point Cloud using Python - CloudComPy" project and article from different sources are referred to gain better understanding on the assigned programming task. After researches had been done, some methods to solve the programming task are identified.

1. 3D Point Cloud

A point cloud is a three-dimensional set of data points or coordinates made up of a huge number of observations. [1] 3D representation of point clouds are made up of hundreds or possibly millions of geo-referenced points. Point clouds are commonly used in industry-standard applications because of the ability of point clouds to provide high-resolution data without the distortion the 3D mesh models that may produce. [2]

The technology for creating and processing point clouds is becoming more affordable and user-friendly. 3D laser scanners and Light Detection and Ranging (LIDAR) technologies are commonly used to take measurements. A laser detects where light strikes surfaces in its line of sight. It can generate an HDR-colored 3D point cloud of its surroundings in less than two minutes. Let's take a closer look at how it generates a point cloud. [1]

2. LIDAR

LIDAR (Light Detection and Ranging) is a remote sensing technique that measures ranges (varying distances) to the Earth using light in the form of a pulsed laser. When these light pulses are integrated with some additional data collected by the airborne system, exact three-dimensional information about the Earth's shape and surface properties is generated. A LIDAR instrument consists of a laser, a scanner, and a specialized GPS receiver. By using LIDAR systems, both natural and man-made surroundings can be analysis with high level of accuracy, precision, and flexibility. [3]

Discrete LIDAR data are created from waveforms, with each point representing a peak energy point along the returned energy. The LIDAR data points will be connected with an X,Y position, and Z (elevation) value. The intensity value of most LIDAR data points represents the quantity of light energy captured by the sensor. A LIDAR system measures the amount of time it takes for emitted light to travel to and from the ground. That time is then utilized to compute the distance traveled. The elevation is then calculated based on the distance traveled. These measurements are made with the help of the major components of a LIDAR system. [4]

3. CloudCompare

Basically, CloudComPy is the Python module interfacing CloudCompare library. CloudCompare is a program for processing 3D point clouds, and was initially created to compare two dense 3D point clouds (such as those obtained with a laser scanner) or a point cloud and a triangular mesh. It is based on an octree structure that is specialized to this task. [5]

CloudCompare is a program for registering and interrogating point clouds, as well as doing other helpful actions such as subsampling to minimize size and performing rapid data visualizations and animations. It able to quickly change the look of the point cloud with the EDL or SSAO filter, and also able to decreases the file's complexity and size by lowering the number of points, while retaining the form. [6]

4. Anaconda

Anaconda is a Python package built by Continuum Analytics that comes preinstalled with many essential Python libraries for data analysis. It includes many tools needed in data science and machine learning in a single installation, making it ideal for quick and straightforward deployment. Anaconda able to create virtual environments to segregate various libraries and versions,and it also includes its own package manager which is called conda, from which libraries can be installed. [7]

Methodology

In this project, an algorithm for analysing 3D point cloud models is developed using Python. The algorithm in the project is built with a graphical user interface, and python scripts, plugins, and the command line are used to configure the 3D model with CloudComPy. Cloudcompare in the Anaconda environment is the GUI Graphical Unit Interface that was used for this project. In this project, the GUI serves three purposes. The REPL (Read Print Eval Loop), one of the command tools for viewing the results, allows for quick testing of some actions. Following that is the editor, which allows you to create, edit, save, and, most importantly, run Python scripts. [8] The third function is to act as a file runner, displaying a small dialogue where you can select a file and run it without opening the editor. Following is the script we used in the project with the Python language to import Python modules that provide access to cloudcompare's functions and data-structures.[11] Plugins are then used to create scripts or modules that register actions that can then be launched from the UI, with the option to specify a path where the custom Python plugin is stored. The path will be scanned to import all Python modules that allow the functon to register the action. Finally, the command line, which works similarly to a command line script in that it can make the plugin take all arguments that follow the script name and populate Python's sys.argv with them. [11] So, from the command line we need to generate the animated process of the module point cloud and analyze it.

Procedure

There will be some tools be having been used to manage the point cloud processing in this project. The Compare Point Clouds tool compares a series of point clouds to locate spots that have changed more than a predetermined minimum difference. This can be used to determine whether or not a file has been edited, as well as to detect changes over time when point clouds from two separate time periods are compared. The clouds may have slight variations in collection discrepancies that may be overlooked with the lowest distance setting, and then the major changes in the terrain will be noticed between the two clouds as time goes on. Then the clouds changes are varied to simulated for data processing which are recorded in the video link given together with the images below.

The 3D surface deviation analysis on the verticality of the module is carried out in this study utilising the CloudCompare software and the C2C distance computation method. The determination and selection of the reference and comparison datasets are the first steps. The deviation analysis result is the final output of the 3D surface deviation analysis. In comparison to the reference dataset, the result may be utilised to detect the poin cloud lies on the verticality built of the modules in the compared dataset. CloudCompare software provides a colour scale that depicts the value of the C2C distance computation for a better comprehension of the result.

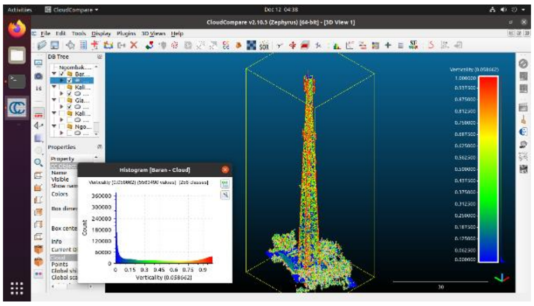
Figure 1: Result of Baran.las file

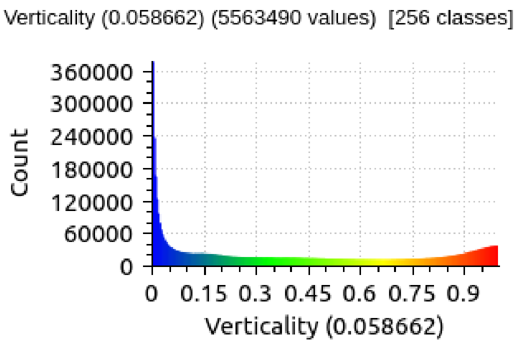

Figure 2: Histogram of Baran.las file

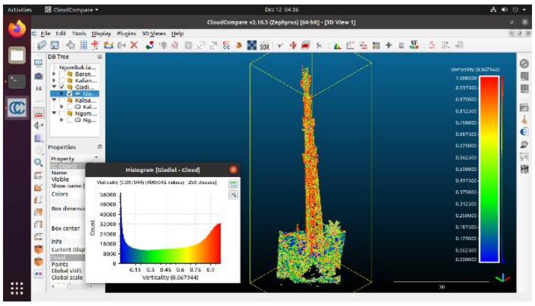
Figure 3: Result of Gladior.las file

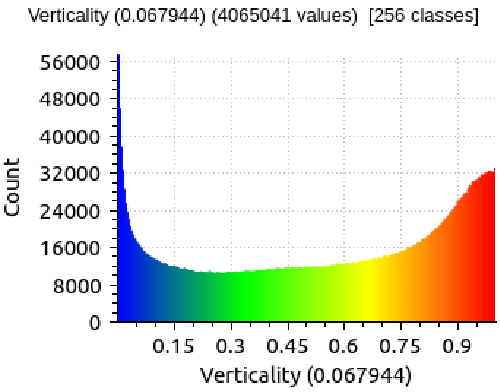

Figure 4: Histogram of Gladior.las file

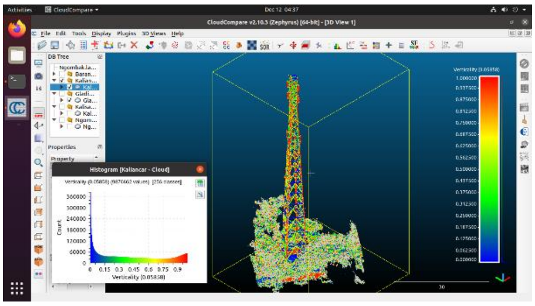
Figure 5: Result of Kaliancar.las file

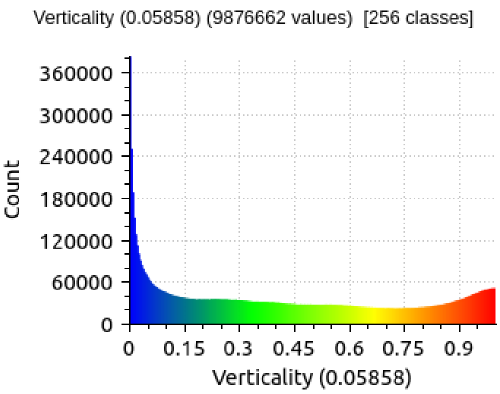

Figure 6: Histogram of Kaliancar.las file

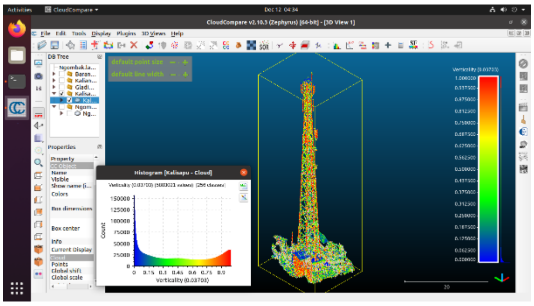

Figure 7: Result of Kalisapu.las file

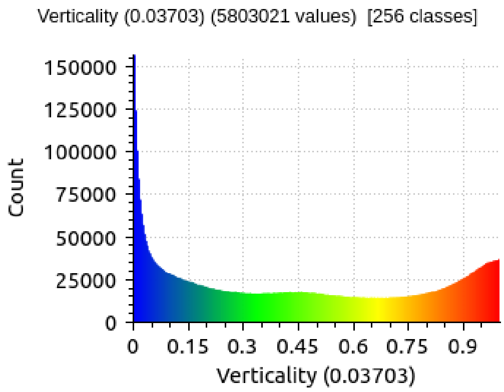

Figure 8: Histogram of Kalisapu.las file

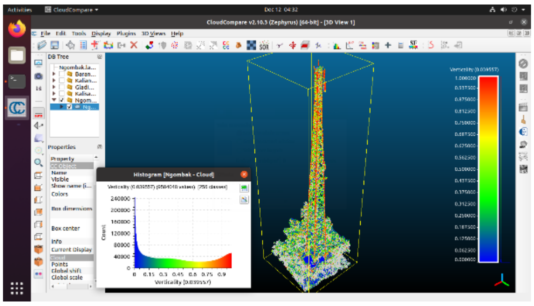

Figure 9: Result of Ngombak.las file

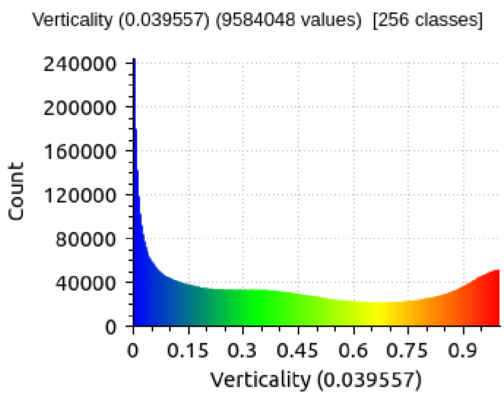

Figure 10: Histogram of Ngombak.las file

The Analysis video of those module are provided in the link given below:

https://drive.google.com/drive/folders/1mJI14nanSl406lcsSseJOUjqSezSlvln?usp=sharing

Conclusion

Several studies were conducted in this project to learn how to create a "3D Point Cloud using Python - CloudComPy." The overview and capability of 3D Point Cloud were studied and analysed, and it was found to be capable of providing high-resolution data without distorting 3D mesh models. Anaconda is a Python package that comes preinstalled with many essential Python libraries, whereas Python module will interface with CloudCompare library in CloudComPy. Throughout the project, we are able to import, sub-sample, export, and visualize a point cloud with millions of points. Finally, we able to execute end-to-end process for 3D Point Cloud by managing the point cloud of the model provided in the CloudComPy and build and automated processing program which could export and analyzed the point cloud.

Reference

1. (2021, February 23) *All You Need to Know About 3D Point Clouds.* Retrived from : https://blog.hexagongeosystems.com/all-you-need-to-know-about-3d-point-clouds/

2. Lindsey H. (n.d) *3D Point Clouds*. Retrived from : https://help.dronedeploy.com/hc/en-

us/articles/1500004860641-3D-Point-Clouds

3. (2021, February 26) *What is lidar?* Retrived from : **https://oceanservice.noaa.gov/facts/lidar.html**

4. Leah A. Wasser. (2020, October 7) *The Basics of LiDAR - Light Detection and Ranging - Remote Sensing.* Retrived from : https://www.neonscience.org/resources/learning-hub/tutorials/lidar-basics

5. (n.d) *Cloud Compare.* Retrived from : https://www.ch.cam.ac.uk/computing/software/cloudcompare-0

6. (n.d) *Point Cloud: CloudCompare.* Retrived from : https://blogs.gre.ac.uk/designsupport/3d-realisation/process-present/cloudcompare/

7. Yufeng G. (2017, November 10) *Which Python package manager should you use?* Retrived from: https://towardsdatascience.com/which-python-package-manager-should-you-use-d0fd0789a250

8. Derrik Diener. (2021, March 24) *How to install Anaconda on Ubuntu.* Retrived from: https://www.addictivetips.com/ubuntu-linux-tips/how-to-install-anaconda-on-ubuntu/

9. Dgirardeau. (2021, October 19) *CloudComPy: Introduction.* Retrived from: https://github.com/CloudCompare/CloudComPy#testing-cloudcompare-1

10. Paul Rascle. (n.d) *CloudComPy: a first example.* Retrived from: https://www.simulation.openfields.fr/documentation/CloudComPy/html/introduction.html

11. Thomas Montaigu. (n.d) *CloudCompare-PythonPlugin.* Retrived from: https://tmontaigu.github.io/CloudCompare-PythonPlugin/getting_started.html

12. (n.d) *What is 3D Point Cloud Laser Scanning?* Retrived from: https://sircleuk.com/services/building-surveys/3d-point-cloud-laser-scanning/

13. Zehao Zhao. (2021, February 16) *Local Feature Extraction Network for Point Cloud Analysis.* Retrived from: https://www.mdpi.com/2073-8994/13/2/321