3D Point Cloud using Python - CloudComPy

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 main objective of this project is to solve the industrial-based problem on visualizing the massive building or construction before implemented in the real environment as digitalized map of points in space which are processed to become 3D models of almost any object.

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 is created to analysed the 3d point cloud models by using the python to develop it. In the project the algorithm is build with an GUI interface, the python scripts, plugin and also the command line are used to configure the 3d model with CloudComPy. GUI Graphical Unit Interface which used for this project is Cloudcompare in the Anaconda environment. The GUI is use to be 3 functions in this project The REPL (Read Print Eval Loop) which allows for quick testing of some actions, one of command tools to view the results. Next is the editor which allows to create, edit, save python script and most importantly run them.[8] The third function is act as a file runner, which shows a small dialog where you can select a file and run it without having to have the editor opened. Followingly is the script we used in the project with the python language that need to import Python modules that provide access to functions and data-structures of cloudcompare.[11] Thenceforth, plugin use to to create scripts or module that register actions that will then be launch-able from the Ui, that possible to indicate a path where custom python plugin is stored. The path will be scanned to import all the python module which allows to register the action needed by the functon. Finally the command line which similarly as commandline script which able to make plugin take all arguments that follows the script name and populate python's sys.argv with them.[11]

Procedure

To run the Cloud-Compare Python in linux operating system we need a data science platform which based in the python. Here we have the Anaconda3 which a Python-based data science platform that's able to run the binary CloudCompy program in its environment. Hence that, the CloudComPy is newly launch program that's which under development program hasn't yet fully stabilized. So the binary file used are only bale to run under the operating system of Linux 64, on recent distribution and with Anaconda3.

Install Anaconda on Ubuntu

First step to install the Anaconda is to get all the dependencies set up on the system, so need to run the command below on the terminal to prepare the dependencies.

sudo apt install libgl1-mesa-glx libegl1-mesa libxrandr2 libxrandr2 libxss1 libxcursor1 libxcomposite1 libasound2 libxi6 libxtst6

After preparing the dependencies of Anaconda's, proceed to download the installation script of it that available on the Anaconda's repo site, By using the Wget downloader command able to get the installation script.

wget https://repo.anaconda.com/archive/Anac *onda3-2020.11-Linux-x86_64.sh -O ~/Downloads/Anaconda3-2020.11-Linux-x86_64.sh*

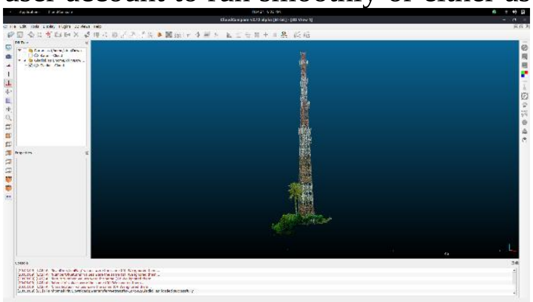
After completing the Downloading the installation script, proceed to the downloaded directory to access it by entering the command below.

cd ~/Downloads

After entering the command, there's an new directory will appear in the terminal by following with the computer host name, so the use the chmod command to update the installation script's permissions for script must have permissions changed so that it can execute, by using the full command below.

sudo chmod +x Anaconda3-2020.11-Linux-x86_64.sh

After permission access run the installation of Anaconda on Ubuntu without the sudo command and as your user account to run smoothly or either as



root program into linux operating system can be done by adding the sudo command before it.

./Anaconda3-2020.11-Linux-x86_64.sh
sudo ./Anaconda3-2020.11-Linux-x86_64.sh

Followingly, the program needs to accept the license agreement for proceed to installation and yes command for the acceptance of reading the script. The select the default directory to install the program, finally its completed and will ask to open the graphical unit interface of anaconda or can access through the terminal by anaconda-Navigator.

Second part is to test the CloudComPy binary on the Linux operating system with the Anaconda3. The binary file are can download through the simulation.openfields platform in the Zip file format. Extract the file and copy 4 folders to the "envs folder" of the "Anaconda3 file" in the "Home folder" of the OS, then merge it. Thenceforth, need to create an environment on the Anaconda for the CloudComPy to run from the terminal, below shows the commands to do.

. ~/anaconda3/etc/profile.d/conda.sh
conda activate
conda create --name CloudComPy39 python=3.9
 # --- erase previous env if existing
conda activate CloudComPy39
conda config --add channels conda-forge
conda config --set channel_priority strict
conda install qt numpy psutil boost xerces-c pcl gdal cgal cmake pdal opencv ffmpeg mysql "qhull=2019.1" matplotlib "eigen=3.3.9" tbb openmp

Concequently, install the binary in the preferred directory, then load the environment to run the CloudCompare or CloudComPy. To do so, snap install cloudcompare.

. ~/anaconda3/etc/profile.d/conda.sh
conda activate CloudComPy39
export LD_LIBRARY_PATH=~/anaconda3/envs/CloudComPy39/lib:${LD_LIBRARY_PATH}
export

```
LD_LIBRARY_PATH=~/anaconda3/envs/CloudComPy39/lib/cloudcompare:${LD_LIBRARY_PATH}
export
LD_LIBRARY_PATH=~/anaconda3/envs/CloudComPy39/lib/cloudcompare/plugins:${LD_LIBRARY_PATH}
anaconda3/envs/CloudComPy39/bin/CloudCompare
cd
<anaconda3/envs/CloudComPy39/doc/PythonAPI_test
ctest
. envPyCC.sh
python test001.py
```

At the final step, open the file Dr. shared and start testing the point view changes in the CloudComPy. The result are shown below.

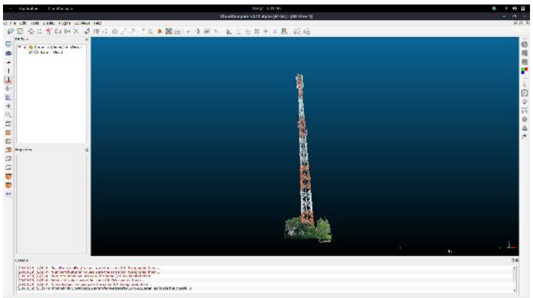


Figure 1: Result of Baran.las file

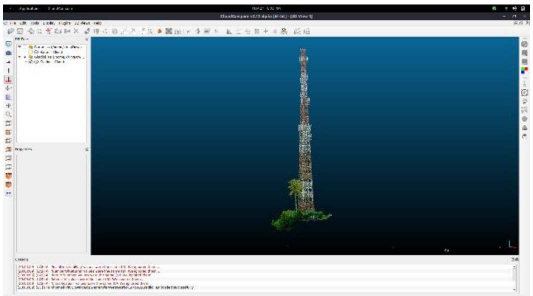


Figure 2: Result of Gladiol.las file

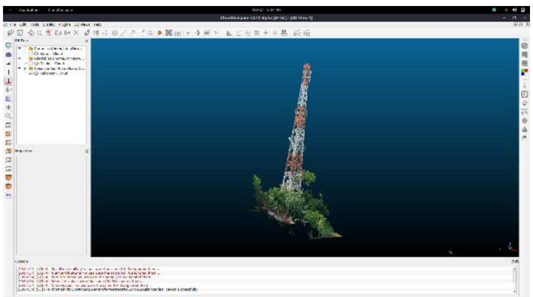


Figure 3: Result of Kaliancar.las file

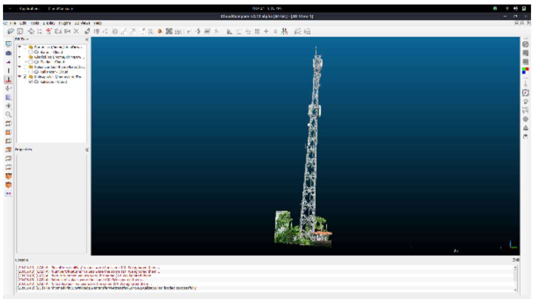


Figure 4: Result of Kalisapu.las file

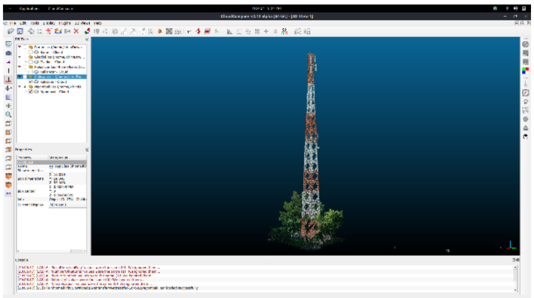


Figure 5: Result of Ngombak.las file

Conclusion

In this project, various researches were undergone to understand how to achieve an "3D Point Cloud using Python - CloudComPy". The overview and the ability of 3D Point Cloud was studied and analyzed, where it able to provide high-resolution data without the distortion the 3D mesh models. The functionally and the relationship of Python with both CloudCompare and Anaconda was also analyzed, where Anaconda is a Python package that comes preinstalled with many essential Python libraries, while Python module will interface with CloudCompare library in CloudComPy. Throughout the project, we manage to import, sub-sample, export and visualize a point cloud that composed of millions of points.

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) C*loudComPy: 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