In Base Processing with a Point Cloud Server

Rémi Cura, etc.

Contents

1. Introduction

1.1. Problem

- Cool, but BIG. Point cloud data is becoming more and more common. Following the same trend, the acquisition speed (points/sec) of the sensor is also increasing. Thus Lidar processing is clocking on the big data door.
- Store/get/process at big scales We need appropriate ways to deal with data at such scale, be it for efficient storing, retrieving, processing of visualizing.
- multi-user/point cloud as a service The use of point cloud is also spreading, going out of the traditionnal geo-spatial LIDAR community. Lidar are commonly used on robots, intelligent cars, architecture, sometimes by non-specialized users. Thus new usages and needs arise which must be tackled. For this, we have to consider point cloud data management as a whole system, and find a solution for this whole system.

1.2. Motivation

- Pointcloud : becoming common (why) Point cloud are becoming common because sensor are smaller, cheaper, easier to use. Point cloud from image (using Stereo Vision) are also easy to get with several mature structure from motion solutions. Point cloud complements very well images, LIDAR pointcloud allowing to avoid the ill-posed problem of stereovision, and providing key data to virtual reality.
- Growing data set + Multi sources At such the size of data set are growing, as well as the number of dataset and their diversity.
- Why is it important (size of the industry) The point cloud data are now well established in a number of industries, like construction, architecture, robotics, archeology, as well as all the traditionnal GIS fields (mapping, survey, cultural heritage)
- PointCLoud users = specialist in processing, not informatics/storing The LIDAR research community is very active. The focus of lidar researchers is much more on lidar processing and lidar data analysis, or the sensing device, than on the lidar data management system as a whole. For instance point cloud compression has received much less attention than surface reconstruction.

1.3. state of the art

RC:1.3.0. when I have access to Zotero

State of the art should include

- storing in filesystem (octree)
- processing (some example)
- paper on storing in database
- what people do with poiint cloud ? (oosterom 2014)

1.4. what's missing in biblio

- limits : file system : list the limits Some of the previously cited articles proposed solution to improve storage (and sometime usage) of point cloud data on a file in a file-system. However such improvements only solve a small part of the problem raised by massive point cloud management. Moreover, the solutions are very specific and may need a large amount of work to be valid end efficient at bigger scales.

- limits of database approach : concept papers, covers only storage (not usages), no in base processing On the other hand, some works have already considered using DBMS to store massive point cloud. These works are more theoretical and focus on studying the tractability of this kind of approaches. This article also only deals with one part of massive point cloud data management (usually the storage), and rarely propose a complete system with substantial data and comprehensive testing.

1.5. objectiv of the paper

- analogy with vector/raster world and server being commons point cloud data is usually a member of the geospatial data. As such, it is interesting to watch the evolution of vector and raster data in the last decade. There has been a massive trend towards server and data base solutions, be it for storage (postgresql), processing (postgis), or visualisation (mapserver). Raster data is quit similar to point cloud data in term of size and semantic level. As such it is particularly interesting to observe that raster data set well above the To scale are common and are easily dealt with server based solutions.

- complete, tested, open source, working system As such, this article proposes a new solution for massiv point cloud data management, ranging from data storage to data retrieval, processing and visualization. This solution has been tested on well sized point clouds, is based on an open source stack, and can be easily reproduced, understood and improved.

- example for each usage We propose real example for common usage of pointcloud.

- can scale The new scale the point cloud data are reaching necessitate to reduce the duplication of data, and do everything in parallel. Our solution is designed to scale, and is based on technology that are currently used at much bigger scale. Thus we have good theoretical indices that the solution will scale well up our tests.

- easily extensible Lastly this solution use a new paradigm for point cloud data management which allows much easier standardisation, re-use and extensibility.

1.6. plan of the article

The rest of this article is organized as follow : In the next section we present our solution. ... In the section XX, we present some results and order of magnitude In the last section we discuss the results, the limitation and improvments of our solution.

2. Method

2.1. introduction

In this section we will present our point cloud server solution.

First we we use a recent analysis of point cloud data usage to detail the system need. Then we present globally our solution. Lastly, we detail how well our solution answer to each need.

2.1.1. work with point cloud data

Before proposing a new point cloud data management system, we need to have a good idea of what are the common needs of people using point cloud data. We resume (OO xxx , 2014), and extract requirements for our system.

Data IO. The most obvious need when working with point cloud data is to **read and write data, and sometime update it. The distinction we make between writing and updating is artificial, and represent the situation where we write in a concurrent context. (for instance, an algorithm classifying a point cloud, executed in a parallel context). From the system perspective, the system should be able to store data (write), and provide data (read), and allow several users (or process) to do both in a concurrency context.**

Data filtering. **Given the size and diversity of new point cloud data set, reading data is not sufficient. One need to be able to work only on a small part (possibly using parallelism). We will refer to this informally has filtering, with a broad acceptance : ranging from taking a sub-set of the point cloud to more active methods like getting only points having certain properties (subsampling, attributes ...)**

Processing. **Processing point cloud data is often the ultimate goal of researcher using those data, and the subject of the majority of article on point cloud. As such, the range of processing methods is very large. That being said, every method needs to work efficiently with it's input data. With huge data, the system should provide easy partitionning and parallelising options.**

Visualization. **The human visual faculty are very impressive, and visualization of 3D data is not only a global trend for our society, but also a powerful way to understand complex data. Yet visualizing billions of points on screen is a challenge and need not only a dedicated graphical solution, but also appropriate subsampling of data (obviously we don't want the rendering engine to render all the points, not even ask for all the points then choose which on to render).**

2.2. Global presentation

2.3. Choosing between a filesystem solution and a DBMS solution

; talk about nosql trend explain that chosing depends on usage (ex: pure storing, no access: filesystem way better)

2.4. Storing pg Pointcloud

how to store points in base -¿ depends on what we want to do with it

2.4.1. introduction to pg pointcloud
2.4.2. principle of storing
2.4.3. compression
2.5. system data IO
2.5.1. writting data to the base
2.5.2. reading data from the base

file stream : point as a service

2.5.3. update data (concurrently)
2.6. Filtering data
2.6.1. what kind o filtering
2.6.2. on meta data : several data sources
2.6.3. spatial
2.6.4. attribute
2.6.5. on function result
2.7. Process data
2.8. Visualize data

3. Result

3.1. Storing pg Pointcloud

6 billions points, several data set

3.1.1. compression

2 factor all inclusive

3.2. system data IO

3.2.1. writting data to the base

write as fast as we acquire

3.2.2. reading data from the base

about 100k pts/sec

3.2.3. update data (concurrently)

3.3. Filtering data

few ms

3.3.1. what kind o ffiltering

3.3.2. on meta data : several data sources

3.3.3. spatial

3.3.4. attribute

3.3.5. on function result

3.4. Process data

for image : few sec per image for point cloud, few secs per k of points

3.5. Visualize data

using LOD, constantly streaming

4. Discussion

5. Conclusion

RC:5.0.0. A ecrire !

6. bibliography

7. TODOLIST

Liste des points à traiter

Liste des points à traiter