An Approach of Web-based Point Cloud Visualization without Plug-in

Mengxuan Ye1,2, Shuangfeng Wei1,2 and Dongmei Zhang1,2

1School of Geomatics and Urban Spatial Information, Beijing University of Civil Engineering and Architecture, Beijing, China
2Key Laboratory for Urban Geomatics of National Administration of Surveying, Mapping and Geoinformation, Beijing, China

E-mail: weishuangfeng@bucea.edu.cn

Abstract. With the advances in three-dimensional laser scanning technology, the demand for visualization of massive point cloud is increasingly urgent, but a few years ago point cloud visualization was limited to desktop-based solutions until the introduction of WebGL, several web renderers are available. This paper addressed the current issues in web-based point cloud visualization, and proposed a method of web-based point cloud visualization without plug-in. The method combines ASP.NET and WebGL technologies, using the spatial database PostgreSQL to store data and the open web technologies HTML5 and CSS3 to implement the user interface, a visualization system online for 3D point cloud is developed by Javascript with the web interactions. Finally, the method is applied to the real case. Experiment proves that the new model is of great practical value which avoids the shortcoming of the existing WebGIS solutions.

1. Introduction

As a new means for surveying and mapping, three-dimensional laser scanning technology make it possible to gather fast a vast volume of spatial point information, which has become an important way to obtain three-dimensional spatial information due to its features such as non-contact, fast and continuously getting information of objects[1]. It is widely used in ancient architecture protection, super-large structural deformation monitoring and city surveying and mapping, etc[2]. The obtained point cloud to be processed is the basis of the subsequent data analysis, 3D mesh modeling and 3D visualization[3]. For this reason, it is necessary to seek a remote, user-friendly, point cloud visualization tool without plug-in for quick assessment of the data quality and the plan of processing[4].

A literature review indicates many web-based 3D mesh viewers are developed, but there are few web-based viewers for 3D point cloud[5], among which are mainly commercial viewers developed by the companies of laser scanner such as Leica Tru View, Faro Scene Web Share, All point Insight, etc. However, these point cloud browser online are mainly customized for the manufacturers themselves. During the development of these browsers, the most important problem lies in that, the standard developed by each company is supported by their own plug-ins. These plug-ins with poor compatibility are required to be downloaded and installed, which brings to the users a lot of inconvenience when the bandwidth is not under ideal conditions. Besides, the formats of point cloud
data acquired from different scanners of the manufacturers are diverse, and these data must be converted before the visualization process, which has also become a big issue.

With the introduction of WebGL, graphic engines such as three.js[6] and Cesium[7] are applied to visualization of point cloud data. Based on these visualization tools, there are also non-commercial systems for scientific research. As an open-source web browser, Potree[8] is required to convert point cloud data locally and upload it to the server, the automation degree is low and the conversion process needs much time. Online LiDAR point cloud viewer[9], a website for visualizing point cloud, supports reading point cloud from server of the users and the data is needed to be placed on their own server, but only ASPRS LAS 1.2 and XYZ formats are allowed. Oscar Martinez-Rubi [10]et al. chose Potree[8] as prototype system to develop an open source website[11], which is based on WebGL and HTML5, but the data is limited to airborne LiDAR data. Zeng X[12] proposed a WebGL-based visualization method for LiDAR point cloud data, which cannot integrate with database for a variety of file formats, still didn’t solve the complex problem of diverse data formats.

This paper presents a Web-based point cloud visualization method without any plug-ins. The proposed model addresses the issue of diverse point cloud data formats, which supports spatial database management and flexible extension. The visualization system is written in JavaScript and has realized online interaction of point cloud by using ASP.NET and WebGL technologies, which also integrates the spatial database based on PostgreSQL. Experiment results indicates that the system can provide users with a good visualization experience for point cloud data in web browser and avoid the problems of plug-in installation and data conversion.

The remainder of the paper has the following structure: Section 2 presents the key techniques. Section 3 discusses the route of technology. Section 4 presents the tests and Section 5 concludes this paper and discusses the future work.

2. The key techniques

2.1. HTML5 and <Canavis> element

HTML is a standard of web development language, which had not been updated more than a decade since HTML4 released. During this decade, the web application development demands more from develop language, such as support for video, asynchronous transportation, vector data rendering, client data storage and other functions[13]. In 2014, the World Wide Web Consortium announced that HTML5 standard specification was finalized and publicly released. Browsers which supports HTML5 include Firefox (Firefox browser), IE9 and later, Chrome (Google Browser), Safari, Opera, etc.

The <canavis> tag in HTML5 can be achieved for drawing vector data[14]. This tag interface can draw points, lines, polygons, text, raster images without any third-party plug-in by using JavaScript. At the same time, attributes and many events of <canavis> tag allow JavaScript to respond to some events to realize graphic and analysis operations.

2.2. WebGL and Three.js

As part of the family of HTML5, WebGL[15] is a web 3D graphics standard developed by Khronos Association, it is originated from the Mozilla Foundation and currently developed by developers of WebGL Working Group of the Khronos and other corporations such as Google, Opera, Mozilla and Apple. WebGL is an OpenGL-based JavaScript programming interface which provides rendering function in the browser. WebGL runs on the Canvas element in HTML5.

Three.js[6] is an open source 3D graphic engine based on WebGL, which allows JavaScript to operate GPU in the browser to realize true 3D scenes. It provides basic framework for WebGL-based visualization of point cloud data, which greatly simplifies the developing process. Therefore, in this paper, Three.js is applied as graphic engine for visualization.

3. The route of technology
The route of technology are as follows: Firstly, building database for the raw point cloud data; Secondly, exchanging data between database and web browser; Finally, creating and visualizing three-dimensional scenes. The process is described in details below.

3.1. Building the database
PostgreSQL is an open-source object-relational database server[16]. It is fast, lightweight and support many file types with rich client interfaces. Accordingly, PostgreSQL is chosen as the spatial database in our work. The steps of building the database are summarized as follows:
- Establishing the project in PostgreSQL and creating tables to store data in the database.
- In order to visualize different formats of point cloud data files, coordinates XYZ and RGB color information extracted from raw data will be saved.
- C++ Libpq interface is chosen as the connection between application and database, the maximum transmission speed of which in campus network can reach to 40MB/s.

The building process is shown in **Figure 1**.

3.2. Data exchange
Since the database has been built, the point cloud data information stored there need to transferred into the browser for rendering. Web Service is used in this period to complete the mission. Web Service can be converted to a web application that can be used by other applications. Basic platform of Web Services is XML + HTTP. By using the ASP.NET AJAX, Web Service can be called with above-mentioned Libpq embedded in the script in order to obtain data.

The process can be reformulated with more details in **Figure 2**.

3.3. 3D scenes
To show complete 3D scenes, four essential components are required including Scene, Renderer, Objects, Camera.

In order to build WebGL scene by using three.js, a HTML page is needed to be created firstly. Three.js library is required to be introduced in the <head> tag. Also, a WebGL renderer is created for loading the scene in the <body> tag. The scene is a container, which is used to store and keep tracking of the objects to be rendered. The Renderer completes conversion process from two-dimensional plane to three-dimensional space of the computer window and then the scene is displayed to the users. After the rendering of three-dimension elements, it is ready to write the script of the relevant code in Javascript. At the beginning, the coordinates and color information of point cloud is obtained from database provided by the callback function.
This research is based on Particle System for point cloud rendering. Considering point cloud as a whole, the system allows manipulating point cloud efficiently such as translation, rotation and scale changing. These interactions need to be added to the camera of the scene with its parameters adjusted, so that the observer can browse the object in the scene from different angles and level of details.

The 3D objects is presented in the Scene by using ‘scene.add( )’ function provided by Three.js to load point cloud data, then various components of the above object are initialized and the scene began rendering.

4. Experiment and Analysis

In this paper, the point cloud data of Beijing University of Civil Engineering and Architecture (hereafter referred as BUCEA) is adopted as the experimental data. For the development platform, we choose Visual Studio 2013 to build the visualization system with PostgreSQL database integrated. ASP.NET and WebGL technologies are also applied to realize the system’s function.

The system consists of three modules: database, web server and the client. PostgreSQL is applied as spatial database. At the same time, the solution uses C# as the programming language to implement server-side response to a variety of operations of the client. The client, using FireFox as a graphic platform, avoids developing another visual interface. The system is running as Figure 3.

![Figure 3. Point cloud management system](image)

Database: It provides interface as the connection between the systems and spatial database. It’s responsible for unified organization and management of point cloud data. To a certain extent, it plays the role of a bridge and link in the relationship between database and browser. The database, PostgreSQL, is just a container for spatial data and Libpq is the conversion channel for data processing.

Server: It is supported by the Web server. Web server provides a runtime environment for the development, deployment, operation, and management of web application services. Web server can also support managing spatial data. In this process, web server receives a request from client, which is processed in WebService at the same time, then returns it to the client after obtaining the results.

Client: Browser acquires data by initiating Ajax Web Service interface. Web Service is responsible for obtaining background data, then the processed data is sent to the browser. As the interface of the system, the Client is the forefront of this system through which users can get provided WebGIS. The system has transferred part of the analysis function to the client, which can increase WebGIS interactive experience and reduce the pressure on the Web server.

The actual operation effect of the built system in FireFox browser window is shown in Figure 4. In order to easily observe and verify the visual effects, panoramic image is also given in Figure 5. The number of points rendered in the scene is 4,102,504 and interactive operation runs smoothly.
Figure 4. 3D point cloud of BUCEA.

Figure 5. Panoramic photo of BUCEA.

In the interface, the upper left is the monitor of FPS, while the right is the control menu, in which the size, color and other attributes of the point cloud can be adjusted, effects are compared in Figure 6 and Figure 7.

Figure 6. Color: green, particle size: 0.01.

Figure 7. Color: red, particle size: 2.81.

5. Conclusions
In this paper, a solution of point cloud data visualization based on WebGL has been introduced and the WebGIS system has been built and tested. Combined with the actual requirement, the paper presents the system development platform, choice of development tools, system structure and functions, and key techniques. Finally, point cloud data visualization based on web is achieved. Test demonstrates that, for online users, it is capable of providing fluent experience when browsing point cloud data. Also, Aiming at the data formats and plug-in problems often occurs in development of publishing system, the spatial database is integrated, which can be used in the actual operating period.
However, due to time and technique reasons, there are many problems to be continue studied in the future. Follow-up studies will focus on the following aspects:

- Large amount of data to read and display. When the point cloud data is massive, the page loading become difficult and improving the performance of massive data reading and visualizing will be an emphasis of future research;
- Comparison with similar research. Due to time limit and other factors, no comparison was conducted to other point cloud publishing system, efficient comparisons is also the object of the next step.

References

[1] ZHANG, R J., WANG, Y M., LI, D R., 2006. Technologies of rapid coping with large amount of 3D laser scanning data. Science of Surveying and Mapping, China, 31 (5), pp. 93-94
[2] LIU, J N., ZHANG, X H., 2003. Progress of Airborne Laser Scanning Altimetry. Geomatics and Information Science of Wuhan University, China, 28(2), pp. 132-137
[3] E. Maravelakis, A. Konstantaras, A. Kritsotaki, D. Angelakis, M. Xinogalos, 2013. Analysing User Needs for a Unified 3D Metadata Recording and Exploitation of Cultural Heritage Monuments System, Advances in Visual Computing, Lecture Notes in Computer Science, Volume 8034, pp. 138-147
[4] J. Fernandez, A. Singhania, J. Caceres, K. Slatton, and R. K. M Starek, 2007. An overview of LiDAR processing software Technical report, Geosensing Engineering and Mapping, Civil and Coastal Engineering Department, University of Florida
[5] M. De La Calle, D. Gómez-Deck, O. Koehler and F. Pulido, 2011. Pointcloud visualization in an open source 3d glob3. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXVIII-5/W16, 2011 ISPRS Trento 2011 Workshop, Trento, Italy
[6] http://threejs.org/
[7] http://cesiumjs.org/
[8] http://potree.org/wp/
[9] http://lidarview.com/
[10] Oscar Martinez-Rubi, Stefan Verhoeven, Maarten van Meersberge et al, 2015. Taming the beast: Free and open-source massive point cloud web visualization
[11] http://ahn2.pointclouds.nl/
[12] Zeng, X., 2012. WebGL based LiDAR Point clouds visualization Journal of Hunan University of Science & Technology, China, 27(4), pp. 60-64
[13] Maravelakis E, Konstantaras A, Kabassi K, et al, 2014. 3DSYSTEK Web-based Point Cloud Viewer[C]// International Conference on Information, Intelligence, Systems & Applications, lisa. IEEE, pp. 262-266
[14] Xu Z., Zhu J., 2011. Research of WebGL based on HTML5 and JSON[C]// Computer Science and Network Technology (ICCSNT), 2011 International Conference on. IEEE, pp. 1714-1717
[15] WebGL Specification. https://www.khronos.org/registry/webgl/specs/1.0/
[16] Li ,X F., 2014. Research on Storage and Query for the Fine 3D Spatial Data. Master’s Thesis, Beijing University of Civil Engineering and Architecture, China

Acknowledgments

This work is supported by The Importation and Development of High-Caliber Talents Project of Beijing Municipal Institutions (CIT&TCD201404070), Beijing Natural Science Foundation(No.8142014), State Key Development Program of Basic Research of China (No. 2012CB725301), the National Natural Science Foundation of China (No. 40801159) and Key Laboratory for Urban Geomatics of National Administration of Surveying, Mapping and Geoinformation (No. 20111215N).