

## 3D Visualization of Point Clouds Using HTML5 and WebGL

---

**Junjie Huang<sup>1</sup>**

*Beijing University of Civil Engineering and Architecture  
Beijing, 100044, China  
E-mail: junjie@stu.bucea.edu.cn*

**Jia Wang**

*Beijing University of Civil Engineering and Architecture  
Beijing, 100044, China  
E-mail: wangjia@stu.bucea.edu.cn*

The wide application of the 3D laser scanner leads to an urgent demand in visualization of point clouds, which, however, has been only limited to desktop-based solutions for years. As an emergent online 3D Engine, Web GL makes several web renderers available. To address the current issue of online point cloud visualization, this paper proposed a method of web-based point cloud visualization without plug-in. This method combines ASP.NET and Web GL, by using the spatial database PostgreSQL to store data of different formats, and HTML5 to implement the user interface. A visualization system for 3D point clouds is thus developed by JavaScript with the web interactions. Finally, this method is applied to the real cases. Results of experiments have verified that the method is very useful in avoiding the shortcomings of current solutions for point clouds online.

*CENet2017  
22-23 July 2017  
Shanghai, China*

---

<sup>1</sup>Speaker

## 1. Introduction

As a new type of measurement method, three-dimensional laser scanning technology makes it possible to obtain a large amount of spatial point information quickly, thus becoming an important method to obtain three-dimensional spatial information with its non-contact[1], fast and continuous acquisition of target information. Widely used in ancient building protection, special deformation proved be useful in large structures and urban measurement[2] and so on. Scanning the acquired point cloud data is a follow-up data analysis, generating a grid model and a three-dimensional visualization of the basis of the display[3]. Therefore, it is of great significance to seek a remote, interactive good, no plug-in network point cloud visualization tool for fast acquisition and evaluation point cloud Data quality and point cloud processing schedule [4].

At present, there are many 3D model browsing tools based on Web in foreign countries, but few of them are aimed at point cloud[5]. Online point cloud browsing is mainly developed by 3D laser scanner manufacturers, such as Faro Scene, Web Share, Leica Tru View, and Allpoint Insight, Point, Cab, web, Export and so on[5]. As an open source network point cloud browser, Potree needs to convert data and upload it to the server locally, with low automation and long conversion time[6]. Point cloud network browser and Online LiDAR point cloud viewer[7] support server point cloud reading, and users need to put the point cloud on their own servers, but it only supports ASPRS LAS 1.2 and XYZ format. Oscar, Martinez-Rubi[8] and others have developed an open-source Web site based on WebGL and HTML5 using Potree[6] as a prototype system, but the data are limited to airborne LiDAR data. In China, research on visualization of Web based on point cloud less, and Zeng Xun[9] put forward a kind of Airborne LIDAR point cloud data visualization method based on WebGL, but not for a variety of data formats of point cloud database integration, which still did not solve the problem of complex and diverse data formats.

In this paper, a Web-based plug-in 3D point cloud visualization method is proposed. For the multi-point cloud data format, spatial database management is adopted. Compared with the traditional model, the new method is more extensible and solves the problems of complicated data formats. In this paper, the real point cloud data, for example, are combined with ASP. NET and WebGL technology, based on PostgreSQL spatial database and JavaScript to write network 3D scene visualization system to achieve the three-dimensional point cloud network interaction. Experiments show that the system can provide users with a good three-dimensional visualization experience of the lidar cloud data in the web browser, and avoid the problem of installing plug-ins and data conversion.

## 2. Technical Route

The technical route of this paper is arranged as follows: firstly, the data of the original point cloud data is built, then the network data exchange is carried out, and finally the 3D scene is created and visualized. The detailed description is shown as follows.

### 2.1 Database Construction

PostgreSQL is a free object-relational database server (database management system), and its installation is simple, fast and lightweight, but the data type and database client interface is

very rich. Therefore, PostgreSQL is selected as the spatial database. The process of building a database is outlined below:

(1) First, build the project database in PostgreSQL and build tables in the database to store the data.

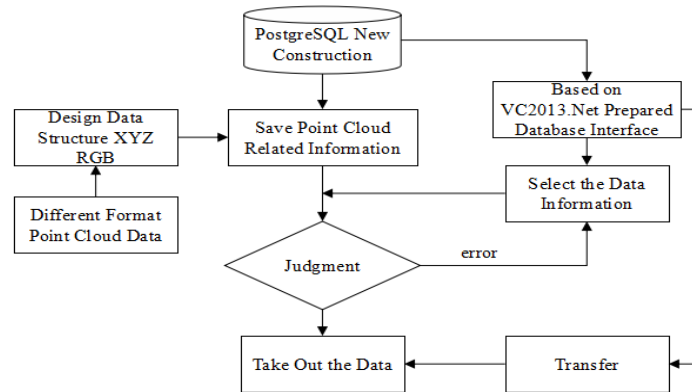
(2) In view of the point cloud data format, for the convenience of visualization, all data are extracted, and the coordinates of X, Y, Z and color RGB information are saved. Later, attributes can be increased according to different cloud format requirements.

(3) The application to access the database based on database interface to complete data access operation, the client interface of different characteristics are not the same and each has its own advantages and disadvantages. The previously mentioned client interface type PostgreSQL support is very rich, and this paper made a comparative experiment in the access to the ODBC, ADO.NET, C++ bottom of the three kinds of common mode the database access mode (Table 1), the final choice of C++ access mode, which in the campus the network transmission speed can reach 40M/s.

The amount of data	ODBC	ADO.NET	Libpq
300000	6s	7s	2s
700000	7s	7s	4s
1000000	13s	13s	7s

**Table 1:** Comparison of Different Connection Ways

Using the Libpq library (PostgreSQL provided in the C application program interface), in the VC++ .NET 2013 platform to write dynamic connection PostgreSQL server applications, the use of prepared storage and call point cloud data interface function, the point cloud data into the database, extract the data. The overall construction process is shown in Fig. 1.



**Figure 1:** Database Construction

## 2.2 Data Exchange

After the database is built, you need to transfer the stored point cloud data to the browser for drawing. This is done by using the Web Service. Web Services can convert applications to web applications that can be exploited by other applications. The basic Web Services platform is XML + HTTP.

Read the above-mentioned Libpq interface into the Web Service, then use ASP.NET AJAX from the script to call Web Service.

In the front page of the JavaScript call Web Service, the method is as follows:

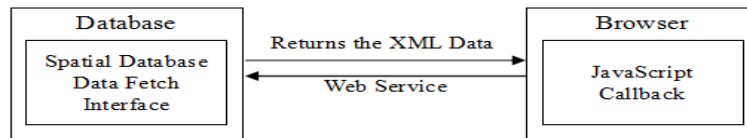
```
< asp: ScriptManager ID = " ScriptManager1 "
runat = " server" >
```

```

< Services >
< asp: Service ReferencePath = " WebService. asmx"
InlineScript = " True" / >
< /Services >
< /asp: ScriptManager >

```

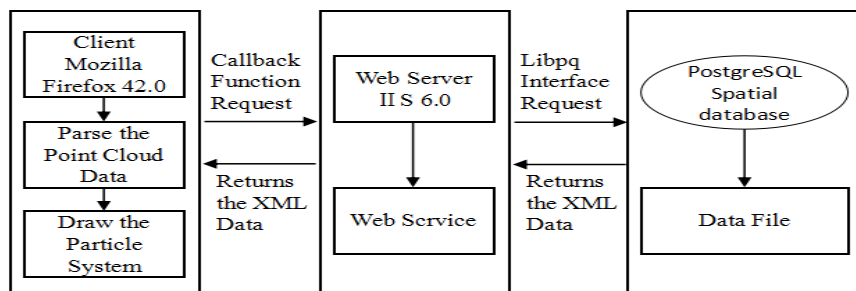
And then through the page in the JavaScript callback function to call to obtain data. The flow is shown in Fig.2.



**Figure 2:** Data Exchange

### 3.Applications

In this paper, the real point cloud data of some areas of Daxing campus in Beijing University of Architecture are used as experimental data, and Visual Studio 2013 is used to develop the visualization system with ASP Web and WebGL technology integrated PostgreSQL database. The system is mainly composed of three modules: database, Web server, and client. PostgreSQL is as a spatial database to C # for the programming language to achieve the server-side client response to a variety of operations; client, using Firefox browser as a graphical browser to achieve the platform, without the need to develop other visual interface, the system running Fig. 3 shows.



**Figure 3:** Point Cloud Management System

Database side: the database side through the spatial database interface is provided by the background system and database link to achieve point cloud spatial data and attribute data unified organization and management. In this process, the database PostgreSQL is just a container to access spatial data while Libpq is the data processing conversion channel.

Server side: the server is supported by the Web server. The Web server provides an operating environment for developing, deploying, running, and managing network application services. The Web server provides management and support for spatial data layers, providing data acquisition and organizing functions. The Web server receives the request from the client, and processes it with the Web Service, and then gets the processing result and returns it to the client.

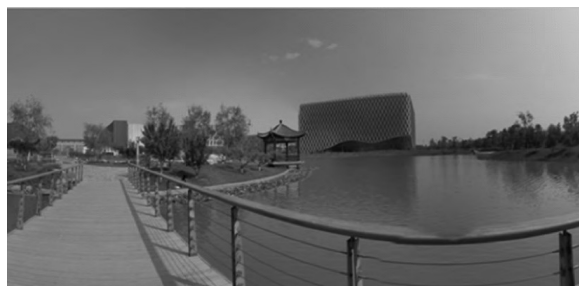
Client: The browser initiates an AJAX access to the data acquisition interface provided by the Web Service, which collects the background data and sends the processed data to the browser. This article adopts HTML5 technology to achieve the point cloud data display, and make use of AJAX technology and Web Service technology to achieve spatial data transmission,

and part of the analysis function transferred to the client, which can increase the Web GIS interactive experience and reduce the pressure of the Web server.

Fig. 4 and Fig. 5 show the point cloud renders scenario (Fig. 4) and the real panorama image (Fig. 5) to facilitate the observation and verification of visualization. As is shown in Fig. 4 and Fig. 5, the number of renderis points 4102504, rotation, zoom, translation and other interactive operation smooth.

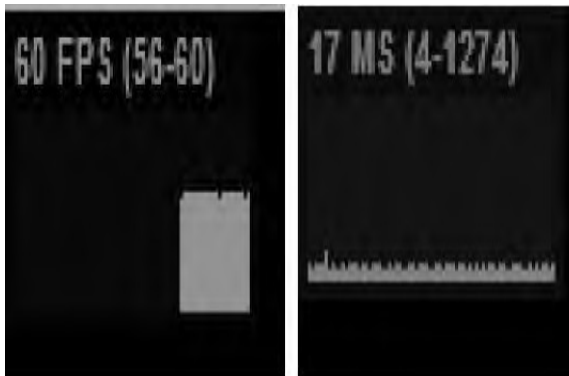
In the system of front-end interface, the top left is the real-time rendering frame number monitor (Fig. 6). In the top right of the page for the operation menu (Fig. 7), you can adjust the point cloud particle size, color, transparency, rotation speed and other attributes.

In order to verify the usability and operating effect of the menu, the effect of changing the particle size and position of the point cloud is shown in Fig. 8 and Fig. 9.



**Figure 4:** 3D Point Cloud of BUCEA

**Figure 5:** Panoramic Photo of BUCEA



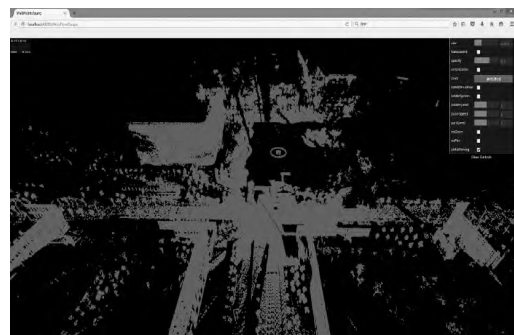
**Figure 6:** Rendering Frames



**Figure 7:** Control Menu



**Figure 8:** Control Menu Operating Results



**Figure 9:** Control Menu Operating Results

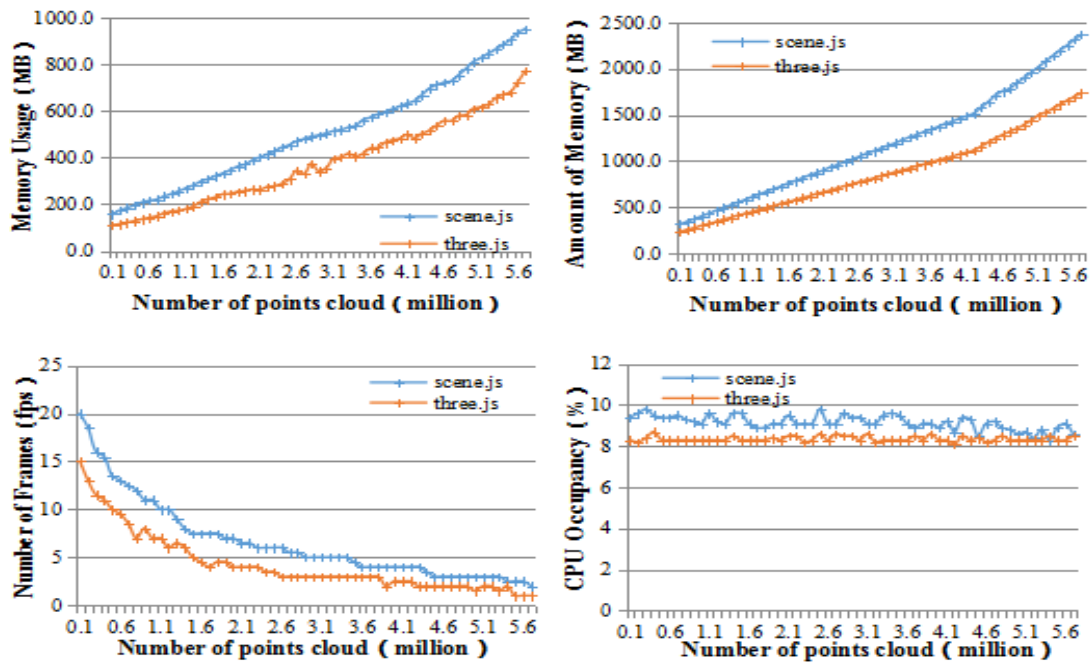
( point cloud colors: red, particle size: 2.818) ( point cloud color: green, particle size: 0.01)

The performance of scenejs and threejs is tested and analyzed in this paper. The test experiment, the variables for point cloud data volume (MB), the dependent variable is the amount of memory (MB), Xian Cunzhan (MB), the amount of frames (FPS), CPU occupancy rate (%), GPU occupancy rate (%), numerical retained after a decimal point .

Experiments on scene.js and three.js on a large number of geometric renders performance of a comparative test. The amount of memory (MB), memory consumption (MB), frames (FPS), CPU occupancy rate (%), GPU occupancy rate (%) and 5 indicators of the statistical data (Fig. 10). The conclusions are as follows:

- Memory usage limit of about 3G;
- Scene.js is better at frame rate performance;
- Scene.js has a CPU occupancy rate of 1%;
- Scene.js has a low GPU footprint of 1%.

For this reason, the building fire blueprint review system uses three.js on the 3D engine.



**Figure 10:** Load Display of Point Cloud Data

To improve the future work, this paper uses scene.js and three.js engine to test each index of the system data transmission, and Maximum Transmission speed, Transfer Time, Transfer Time, Transfer Time and the result is shown in Table 2.

The Number of Clouds	Maximum Transmission speed		Transfer Time		Transfer Time		Transfer Time	
	Scene.js	Three.js	Scene.js	Three.js	Scene.js	Three.js	Scene.js	Three.js
0.5million	8.5MB/s	10.8MB/s	3.3s	2.3s	8.0s	5.0s	Excellent	Excellent
1million	10.8MB/s	15.1MB/s	6.2s	4.1s	11.0s	5.0s	Good	Excellent
2million	26.7MB/s	35.6MB/s	16.3s	10.1s	27.0s	16.0s	Good	Excellent
3million	33.5MB/s	40.7MB/s	21.6s	15.0s	41.0s	25.0s	Average	Good

**Table 2:** Test of Each System Indexes

As can be seen from Table 2, as the amount of data increases, the maximum speed of the three.js 3D engine increases, and the transmission and rendering are more time-consuming, but the operating system maintains good fluency. Three.js is an open source 3D engine based on WebGL, which allows JavaScript to operate GPU and implement true 3D on the browser side. It provides the basic framework needed for the visualization of point cloud data network based on WebGL, which greatly simplifies the development process of visual applications.

## 4. Conclusion

This paper introduces a WebGL-based point cloud data browsing method, and builds a Web GIS system to test it. The key technologies of database construction, front and back data exchange and visualization process are combined with practical needs, and the implementation mechanism and processing flow are expounded. Finally, Web point cloud data display is realized. Experiments show that Web-based point cloud display system for online users in the browser to provide a smooth interactive three-dimensional visual experience. The trial system has a database integration for a variety of point cloud formats, and the network visualization does not need to download any plugins.

In this paper, the method of visualization of network point cloud without plug-in is discussed and experimented. However, due to time and technical level, there are still some shortcomings in this paper, which need to be further studied. Subsequent studies will focus on the following:

(1) Large data volume read display. When the point cloud data is large, the page load becomes very difficult, and then how to improve the large amount of data point cloud data read display will be the focus of future research;

(2) Further enhance the client's ability to analyze space. This article only focuses on the client to achieve the point cloud scene visualization, and did not integratr point cloud data pickup analysis and editing functions, so next I will further study how to achieve point cloud data analysis function;

(3) Due to the constraints of time and conditions, there is no other point cloud release system for system efficiency comparison, and the system interface is not beautiful, which will be further studied in the next research work.

## References

- [1] Zhang Ruiju, Wang Yanmin, Li Deren et al. *Technologies of rapid coping with large amount of 3D laser scanning data* [J]. *Science of Surveying and Mapping*, 2006, 31(5): 93 ~ 94, 68.
- [2] Liu Jingnan, Zhang Xiaohong. *Progress of airborne laser scanning altimetry* [J]. *Geomatics and Information Science of Wuhan University*, 2003, 28(2): 132 ~ 137.
- [3] E. Maravelakis, A. Konstantaras, A. Kritsotaki, D. Angelakis, M. Xinogalos, "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, 2013, pp: 138 ~ 147.
- [4] J. Fernandez, A. Singhanian, J. Caceres, K. Slatton, and R. K. M. Starek, "An overview of lidar processing software Technical report", *Geosensing Engineering and Mapping*, Civil and Coastal Engineering Department, University of Florida, 2007.
- [5] M. De La Calle, D. Gómez-Deck, O. Koehler and F. Pulido, "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, 2-4 March 2011, Trento, Italy.
- [6] [Http://potree.org/wp/](http://potree.org/wp/) [OL].
- [7] [Http://lidarview.com/](http://lidarview.com/) [OL].
- [8] Oscar Martinez-Rubi, Stefan Verhoeven, Maarten van Meersberge et al. 2015, Taming the beast: *Free and open-source massive point cloud web visualization*.

- [9] Zeng Xun. *WebGL based LiDAR point clouds visualization*[J]. Journal of Hunan University of Science & Technology, 2012, 27 (4) : 60 ~ 64.