ISSN: 2688-8645

Visualization of Vehicle LIDAR Point Cloud Data based on Unity 3D

Tanggui Huang[1, a]

[1]School of Logistics Engineering, Shanghai Maritime University, Shanghai 201306, China.

[a]1195056764@qq.com

Abstract

A method of local visualization and real-time visualization system design of large-scale vehicle-mounted laser radar point cloud data based on Unity 3D engine technology is proposed. The visualization method of cloud data in this location is to build a multi-level detail model (LOD) based on a multi-resolution octree, then select the LOD model through human visual feature technology, and finally render the Mesh. The visualization method of real-time point cloud data is to assign values to the attributes of a specified number of clones generated by the system in advance, so as to achieve real-time requirements and the effect of point cloud drawing. On the basis of the above methods, the first version of the software system for visualization of LIDAR data was developed. Experiments show that the system can provide users with a good interactive 3D visualization experience of vehicle LIDAR point cloud data.

Keywords

Interactive visualization, Vehicle LIDAR, Unity 3D, Point cloud.

1. Introduction

In recent years, LIDAR has been rapidly developed and widely used due to its small use environment restrictions, high degree of automation, fast real-time, and high precision. At the same time, the functions of various types of LIDAR software have been continuously improved. At present, there are three mainstream commercial softwares, Terra Solid, Li-DAR-Suite, and Li DAR360. However, the functions of these software are used for later data processing and cannot achieve real-time visualization of point cloud data. So, the system designed in this paper integrates the visualization of local cloud data files and the real-time visualization of point cloud data (every time the radar scans, it visualizes a circle), which greatly facilitates the visual processing of point cloud data by users.

Xianfeng Huang et al. [1] proposed a method based on sequential Quadtree data preprocessing to quickly draw large-scale point cloud data. Jianwen Mo et al. [2] applied multi-resolution quad-tree adaptive level of detail technology for terrain simulation on 3D radar terminals. How to visualize large-scale radar point cloud data more realistically and smoothly is a research hotspot in the neighborhood of radar point cloud data processing. The preprocessing of point cloud data is one of the key steps. After comparing the performance of a variety of vehicle-mounted LIDAR point cloud data preprocessing methods, the system proposed in this paper uses a multi-resolution octree-based point cloud data preprocessing.

At present, the real-time visualization of radar is generally visualized through OpenCV and PCL. Although OpenCV can achieve real-time display, its visual interface is usually relatively simple, it is troublesome to develop interactive functions, and the user's interactive experience is poor. PCL mainly has better advantages in the later processing algorithms of point cloud data, and the visualization effect of point cloud data is general. Therefore, this article uses Unity 3D

ISSN: 2688-8645

technology combined with C# language for the first time to visualize point cloud data in real time.

2. System Function Framework

The module planning of this system can basically meet the requirements of local visualization and real-time point cloud data. At the same time, by designing some commonly used interactive functions, it greatly facilitates the use of users.

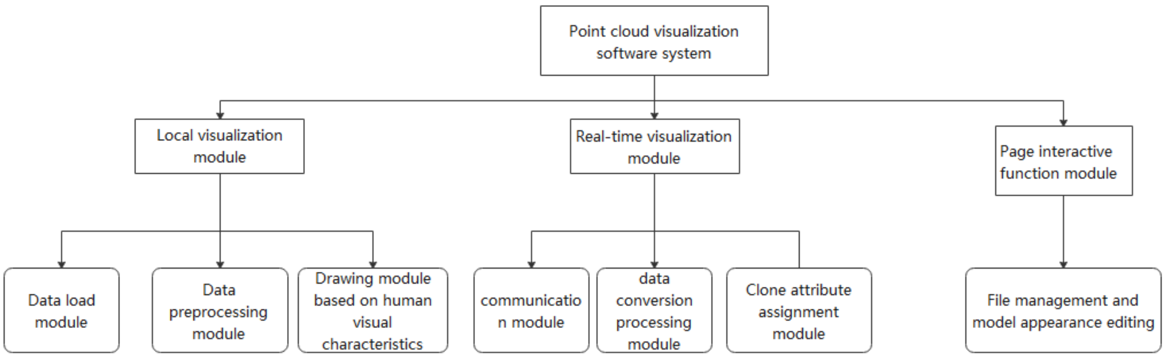

Fig 1. System structure diagram

3. Implementation Principle and Introduction of Each Module of the System

3.1. Local Visualization Data Load Module

The system integrates the analysis library for common pts, LAS, .xyz, PCD, txt format point cloud data formats. The analysis library is written in C# language, and the analysis functions of various formats of point cloud data are encapsulated in different static classes. When designing the analytic library, set the formal parameters of the analytic function inside to “point cloud data path + file name”. When calling, only need to instantiate the class to be called, and pass in the point cloud data path and file name as parameters to obtain the point cloud data content required by the data preprocessing module. At present, the point cloud data obtained by the module after analysis includes three-dimensional vector coordinates and reflectivity.

3.2. Local Visualization Data Preprocessing Module

Point cloud data preprocessing is the core of local visualization. The point cloud data organization form of the Quadtree is generally suitable for the data mainly presents a planar (X-axis, Y-axis) distribution, while the vehicle-mounted LIDAR obtains higher-precision three-dimensional point cloud data (X-axis, Y-axis, Z-axis). Therefore, the data organization form of the Quadtree is not effective in the rapid visualization of the vehicle LIDAR point cloud. The octree is an extension of the Quadtree in space, and is mainly used for processing data in complex scenes such as 3D image processing. After the point cloud multi-resolution storage structure based on the octree structure is constructed, the point cloud data can be quickly rendered [3]. Therefore, the visual data organization of the point cloud data in the local point cloud data visualization module of the system in this article uses an octree organization form. The process is as follows:

(1) Determine the number of octree layers and root node cloud density according to the Bounding Box and data volume of the point cloud data;

(2) Determine the node and the number of layers each point belongs to;

(3) For the root node, if the distance between the point and other points in the root node is greater than the threshold, it is included in the root node until no new point can be added, and then the same method is used for the child nodes to fill in the child node data.

(4) The point cloud data that is not attributable is directly discarded;

The index established by using the multi-resolution octree directly constitutes the multi-level model of detail (LOD) of the point cloud data. The number of point clouds of each node of the model is not much different, and the data can be loaded quickly to achieve rapid drawing of large-scale point clouds.

3.3. Local Visualization Drawing Module ased on Human Visual Characteristics Module

The point cloud visualization method in the literature [4-7] is to load the entire point cloud statically at one time, so that a lot of time will be spent in the loading process, and if you want to obtain a better visualization effect, it requires hardware very high. One of the main technologies for accelerating the display speed of 3D data models is visual correlation technology. The basic principle is to reconstruct the neighborhood around the viewpoint according to the stereo vision characteristics of the human eye [8]. After the point cloud data is preprocessed, only the data of the root node of the octree (the LOD model with the lowest accuracy) is loaded when the drawing module starts, and the subsequent input of point cloud data is processed by a newly opened background thread to avoid affecting the user’s interactive operation. This module calculates the Euclidean distance between the camera's point of view and the target center point in Unity as the basis for choosing the accuracy of loading LOD models with different accuracy. First, when the distance is within the visibility threshold of a certain accuracy LOD model, the system will dynamically extract the data of the corresponding model from the octree in the preprocessing module. Then, dynamically generate Mesh equivalent to the amount of point cloud data in Unity 3D. Finally, the point cloud data is assigned to the generated Mesh, thereby completing the progressive drawing of the point cloud. This not only speeds up the drawing speed, but also the visualization effect is to simulate the visual characteristics of the human eye, which can bring a better user experience to the user.

3.4. Real-time Visualization Communication Module

This module integrates singleton classes of commonly used communication protocols TCP/UDP.

This type of communication protocol is written based on Socket (Socket is the middleware abstraction layer for communication between the application layer and the TCP/IP protocol family), which is suitable for most radars on the market. Since TCP is transmitted in the form of a stream, compared to the UDP module, the TCP module involves operations such as sub-packaging and sticking. In order to solve this problem, the module provides an interface for users to define the form of a package. When the user calls the interface, write the parameters of "message header + message length + end specified symbol", and then the program will automatically subpackage and paste the package. In addition, although the UDP received must be a complete packet, there are also packet loss problems caused by network delays and long receiving end processing time. In order to effectively solve this problem, the module provides setting parameters for the size of the receiving buffer, and the user can adjust the size of the most suitable receiving buffer according to the specific situation of the radar.

3.5. Real-time Visualization Data Conversion Processing Module

The data source received by the data conversion processing module is the data of each packet received and processed by the communication module. The function of the data conversion processing module is to convert this packet of data into container type data that can be assigned to the clones (some small circle model generated by unity). The reason why there is no more general real-time LIDAR point cloud visualization software on the market is to transmit data

ISSN: 2688-8645

content. Here, each radar has its own content that needs to be transmitted (coordinates, reflectivity, scanning angle, etc.). The visualized point cloud data content mainly includes XYZ coordinate information, RGB, scanning angle and other data that can be directly seen in the sense. Therefore, in order to solve the problem of different transmission data content of different LIDAR's, this module provides a script reference interface, that the user can write the data analysis code segment about using the LIDAR on the specified script. This script specifies that the input data type is a byte array, and the output data type is different according to the data type. And the specific provisions are as follows:

Table 1. Output data type

Information category	type of data
XYZ coordinate information	Vector3(float(coordinates))
Reflectivity	Float[]
RGB	Int[]
Scan angle	Float[]

3.6. Real-time Visualization Clone Attribute Assignment Module

If a clone is dynamically generated for each frame of point cloud data of the LIDAR, then large-scale clones will be continuously generated and destroyed during the drawing process, which will consume a lot of time and memory, that is obviously unrealistic. Therefore, the module adopts the following processing methods:

1) The user enters the number of clones to be generated before starting the real-time visualization module. In the Unity3D script life cycle, the cloned program will be called in the Start function of the script, and the Start function is called before the Updata function and is executed only once.

2) In the Updata function, each frame of LIDAR's point cloud data is assigned to the clones, and the real-time visualization of point cloud data is achieved by continuously changing the coordinates, color and other attributes of the clones.

3.7. Page Interactive Function Module

The design of the interactive function of the joining page is mainly to improve the user's interactive experience. At present, this module has designed file management, point cloud model appearance editing and other modules. Frist, file management module includes common reading file, saving file, exporting file, etc. Point cloud model appearance editing including model size, model color, model reflectivity, etc. And the design and implementation of this module is based on the UI framework of Unity 3D, reaching the level of game-level human-computer interaction effects.

4. Systematic Experimental Results

Due to space reasons, the results of the two core modules (local visualization module and real-time visualization module) of the laser radar data visualization software system are displayed here.

ISSN: 2688-8645

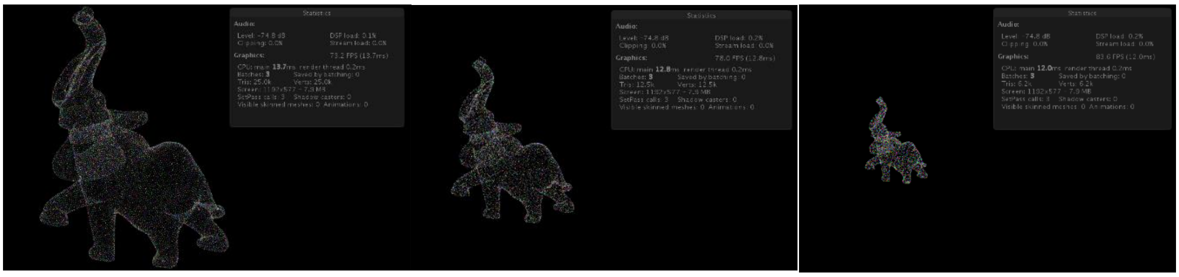

Fig 2. Local visualization module experiment results

The test local real-time module uses the famous elephant point cloud model. The above experimental results from left to right represent the LOD model with decreasing accuracy. It can be seen from the above that FPS (frames per seconds) gradually increases, indicating that the time spent rendering the model is gradually decreasing. In addition, Verts and Tris, the number of triangles and vertices drawn, gradually decrease, indicating that the farther the viewpoint is from the target, the lower the accuracy, and the less point cloud data extracted by the system for rendering. Therefore, a multi-resolution octree LOD model for large-scale point cloud data visualization is feasible.

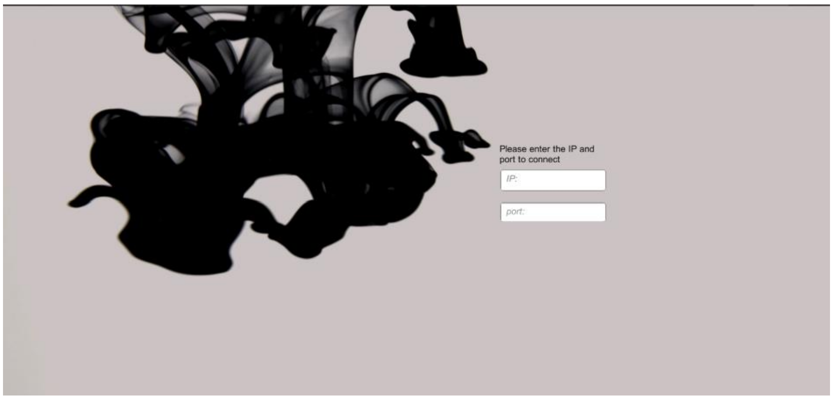

Fig 3. Real-time module startup page

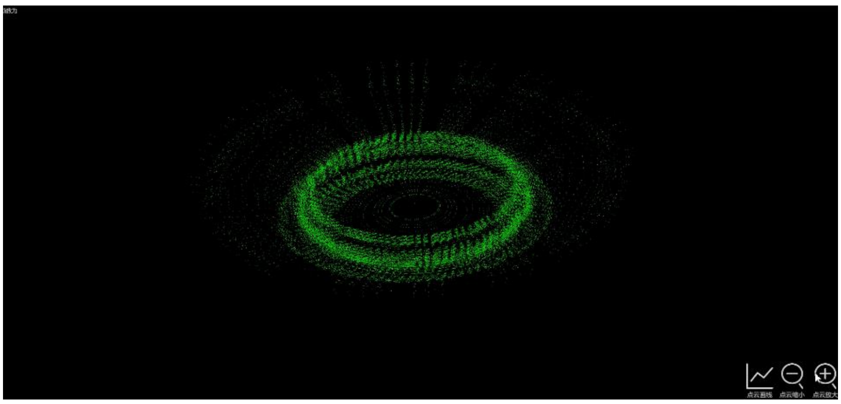

Fig 4. Radar real-time visualization of city squares

The LIDAR used to test the real-time visualization module is 16L-LiDAR (VANJEE TECHNOLOGY), Figure 3 shows the page connected to the LIDAR by TCP protocol. You need to enter the IP and Port of the LIDAR. For the LIDAR with the UDP protocol, the software system is equivalent to the server used to receive information (strictly speaking UDP does not distinguish between client and server) It also needs to bind its own IP and Port, so the pages of the two are similar. The test in Figure 4 is that the LIDAR is placed in an empty stadium, and the outermost stairs and the middle guardrail can be seen. Explain that the design of the real-time module is reasonable and can achieve the expected effect.

The test results of the entire software system function show that the system can run stably for a long time, and all functions can run in a timely and correct manner, which greatly improves the effect of point cloud data visualization and meets the expected requirements of the design.

5. Summary

This article uses Unity3D technology for the visualization of point cloud data of vehicle LIDAR for the first time, which enhances the realism and user experience of large-scale point cloud data visualization, and can enrich the human-computer interaction experience between users and the point cloud visualization system. This article describes the implementation principle of the system in detail. Therefore, users can design the system according to their own needs, so that they can efficiently complete the local or real-time point cloud data visualization of LIDAR without paying expensive professional software fees. At the same time, the vehicle-mounted LIDAR visualization technology based on Unity 3D lays a certain foundation for the future development and improvement of general LIDAR software systems.

References

[1] Xianfeng Huang, Tao Cuang, Wangshou Jiang, et al. Real-time rendering of airborne LIDAR point cloud data. Journal of Wuhan University (Information Science Edition). Vol. 30 (2005) No. 11, p. 975-978.

[2] Jianwen Mo, et al. Application of Adaptive LOD in 3D Radar Terminal Simulation. Computer Simulation. Vol. 6 (2006) No. 6, p. 201-204.

[3] Yan Li, Hu Xiaobin, Xie Hong, et al. Research on Management and Visualization of Massive Point Cloud Data of Vehicle-mounted LiDAR. Journal of Wuhan University (Information Science Edition). Vol. 42(2017) NO. 08, P.1131-1136.

[4] Zhi Xiaodong, Lin Zongjian, Su Guozhong et al. Reseatch on organization of airborne LIDAR point cloud based on improved quadtree algorithm. Computer Engineering and Applications. Vol. 46(2010) No. 9, p. 71-74.

[5] Yv Fei, Cujiang Chen, Liyuan Wang. Multi-scale visualization and efficient management of massive lidar point cloud data. Engineering Investigation. Vol. 044 (2016) No. 009, p. 69-73.

[6] Luode An, Liao Liqiong. Ground lidar data simplification based on quadtree partition. Journal of Computer Applications. Vol. 25 (2005) No. 2, p. 420-421

[7] Yang Jiansi: Research on the visualization method of airborne/ground mass point cloud data organization and integration (Doctoral dissertation, Wuhan University, China, 2011)

[8] Soetebier I, Birthelmer H, et al. Managing large progressive meshes. Computers & Graphics, Vol. 28(2004) NO. 05, P.691-701.