Management Point Cloud Data based on GeoSOT-3D Subdivision Model

Zhongyue Liu¹, Xian Guo¹,*

¹School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture, Beijing, China
*Corresponding author’s e-mail: guoxian@bucea.edu.cn

Abstract: Owing to the rapid development of earth observation technology, the volume of point cloud dataset is growing rapidly. The efficient management and retrieval method is very important for effective use of the large point cloud datasets. Current methods for point cloud data management are generally based on file which makes it difficult to describe internal structure of the modelling objects, while current indexing is generally based on coordinates which is quite time-consuming for large volume cloud point data. To solve these problems, we proposed a method to manage and index point cloud data with a global grid model GeoSOT. The subdivision model of three-dimensional point cloud data was defined to identify the global position of point clouds and internal points with GeoSOT-3D grids. The management and indexing of point clouds were implemented based on the GeoSOT-3D grid codes integrated with the metadata table. A 3D prototype platform for showing the method was developed based on the Cesium. Experimental results show that our method is able to manage, query, visualize massive point clouds as well as other data sources in a unified global position framework.

1. Introduction
The rapid development in laser scanning and LiDAR technique makes it efficient to generate 3D point cloud data from the surface of objects[1]. Due to its simple acquisition, convenient processing and intuitive display, point cloud data have been widely used in surveying, transportation, emergency rescue, cultural heritage, etc. Since point cloud data sets are a collection of discrete points, with the size of data up to GB or TB[2], it remains challenging to efficiently manage and index when applying point cloud data.

As point clouds are designed to represent both the surface and internal structure of geographic entities located in diverse geographic locations and environments, it is necessary to consider the representation of surface and interior structure of geographic entities, as well as global queries and visualizations that integrate the geographic environment. At present, most point cloud data are managed in a file-based way[3]. However, the file-based point clouds management method needs to load the whole file into computer memory when processing, but this method will lead to massive memory consumption or even system crashes when dealing with point cloud data of large scale. Moreover, the storage and management of point cloud data in these systems are mostly based on local coordinates, which makes it difficult to represent the geographic coordinates of entities and determine the relation to environment they located. The storage management method based on discrete points on the surface of a geographic entity cannot support the description and representation of the surface and interior properties of the geographic entities.
From the data indexing aspect, current point cloud data index is mainly based on point such as octree[4], R tree[5], K-D tree[6], regular grid and mixed index[7]. The above index methods are fundamentally flawed when dealing with point clouds. Octree spatial index is subdividing space into eight equal sub-space regularly and iteratively. It is easy to implement and has good operability. However, when dealing with massive and unevenly distributed point cloud data, increasing the depth of octree will not only consume large computation space but also reduce the query efficiency[8]. The basic idea of regular grid index is dividing the space containing point cloud into grids with equal size and recording the point data of each grid. It will calculate the grid of the queried points firstly and then use the grid to retrieve the selected points when user queries. The algorithm is simple and has certain applicability, but if the grid spacing setting is too small, the point cloud distribution inside the grid is not evenly enough, if the grid spacing setting is too large, it will cause a lot of space waste[9]. These methods can only support the index of discrete point location, which are difficult to support the inquiry and management based on geographic entities.

To deal with the difficulties in massive data management, we proposed a point cloud data management method based on GeoSOT-3D global subdivision framework. This method provided an alternative and contributes to the expression of global location and internal attribute. Then we developed a prototype Web platform to organize the spatial information metadata from the point cloud data. Thereby achieving:(1) orderly storage and efficient retrieval of global, massive point cloud data, (2) the ability to provide users with spatial information through a unified global subdivision grid framework, and (3) the ability to establish association between point cloud data and other spatial data through subdivision grid.

2. Methodology

2.1. GeoSOT-3D subdivision grid framework

GeoSOT (Geographic Coordinate Subdividing Grid with One Dimension Integral Coding on 2° Tree) proposed by Cheng[10] has been widely applied in remote-sensing data management. The GeoSOT-3D extended the two-dimensional global subdivision framework into a third spatial dimension: the elevation dimension[11]. By subdividing the Earth through three iterations (initially expanding the Earth $(180^\circ \times 360^\circ)$ into $512^\circ \times 512^\circ \times 512^\circ$ grids, then expanding each $1^\circ$ into $64^\prime$, and finally expanding each $1^\circ$ into $64^\prime$), two-dimensional quadtree subdivisions at the degree, minute, and second levels are obtained[12]. The principle is shown in Figure 1. GeoSOT-3D is congruent and aligned: the largest subdivision grid in the highest level (Level 0) can represent the entire Earth surface, while the smallest subdivision grid in the lowest level (Level 32) can represent the centimeter scale. Table 1 shows the size of the GeoSOT-3D grid at each level and its approximate scale near the equator.

![Figure.1 Geosot-3D grid multi-level subdivision diagram](image)

| Level | Size   | Scale near the equator | Level | Size   | Scale near the equator | Level | Size   | Scale near the equator |
|-------|--------|------------------------|-------|--------|------------------------|-------|--------|------------------------|
| 0     | $256^\circ$ | Earth                  | 11    | $16^\prime$ | 32km                  | 22    | $1/2^\prime$ | 16m                   |
| 1     | $128^\circ$ | $1/4$ Earth            | 12    | $8^\prime$  | 16km                  | 23    | $1/4^\prime$ | 8m                    |
| 2     | $64^\circ$  | $16384$km              | 13    | $4^\prime$  | 8km                   | 24    | $1/8^\prime$ | 4m                    |
| 3     | $32^\circ$  | $8192$km               | 14    | $2^\prime$  | 4km                   | 25    | $1/16^\prime$| 2m                    |
| 4     | $32^\circ$  | $4096$km               | 15    | $1^\prime$  | 2km                   | 26    | $1/32^\prime$| 1m                    |
| 5     | $16^\circ$  | $2048$km               | 16    | $32^\prime$ | 1km                   | 27    | $1/64^\prime$| 0.5m                  |
In the GeoSOT-3D subdivision framework, the code order is Z-sequence coding and the code direction of different regions is shown in Figure 2. Set the grid code of k-level as \( a_0, a_1, a_2...a_k \) \((0 \leq k \leq 31, a_0 = 0, \text{value range of } a_0-ak \text{ is 0-7}). In the region of the northeast hemisphere, 0-7 are arranged in reverse Z order. In the regions of the northwest hemisphere, southeast hemisphere and southwest hemisphere, the code of grid units is the same as the northeast hemisphere except \( a_1 \), which are symmetric about the prime meridian, the equator and the origin respectively.

![Figure 2 The sequence of GeoSOT-3D grid code](image)

2.2. Subdivision model of three-dimensional point cloud data

Based on GeoSOT-3D framework, the method to manage and visualize point cloud data in this paper is that the space from the core to the outer layer of the earth has been completely covered by the subdivision grid, and a multi-scale voxel structure has been formed. The spatial location of each point cloud data can be transformed into 3D subdivision grid codes. Due to the uniqueness of these codes, the target will be positioned in a specific spatial region, so that the spatial region location oriented point clouds organized mechanism can be established by using GeoSOT-3D region identification code of the point cloud data without changing the point clouds format.

Organization and management of point cloud data based on GeoSOT-3D framework includes three steps: global position of point cloud data, multi-scale identification of internal points and visualization of Web system.

2.2.1. The global position of point cloud data based on GeoSOT-3D

The global position of point cloud data is to establish consistent correlation between data and geospatial regions based on subdivision code according to GeoSOT-3D subdivision code framework. The basic idea to calculate subdivision code is to calculate the minimum bounding box of point cloud data, then select a subdivision level similar with spatial scale according to the spatial region of the minimum bounding box, and finally obtain the global position subdivision code through grid encoding function.

For the whole point cloud data file, the level of subdivision code should be first determined. Suppose the length of the minimum bounding box is \( D_L \), the width is \( D_W \), the height is \( D_H \), the subdivision level of GeoSOT-3D grid is \( L_k \) \((k \in [0,32])\), and the side length of relevant GeoSOT-3D grid is \( D_k \). Compare the maximum value of the bounding box with grid length, if \( D_L \geq D_W \geq D_H \), the level of subdivision code is \( L_A = L_k + 1 \) when \( D_k + 1 \leq D_L < D_k \).

The number of codes is determined on the basis of the determination of subdivision code level. Assuming that the eight corner coordinates of minimum bounding box of point cloud model are transformed to the corresponding subdivision code in the \( L_A \) level in order \( C1, C2...C8 \) (Z-ordered from the lower left corner point, as shown in Figure 2):

1) If \( C1=C2=...=C8 \), the number of subdivision code for point cloud model is 1.
2) If \( C_1 = C_2 = C_3 = C_4 \) and \( C_5 = C_6 = C_7 = C_8 \), the four codes are equal and different from the others, the number of subdivision code for point cloud model are 2.

3) If \( C_1 = C_2, C_3 = C_4, C_5 = C_6, C_7 = C_8 \), the two codes are equal and different from other codes, the number of subdivision code for point cloud model are 4.

4) If \( C_1 \neq C_2 \neq \ldots \neq C_8 \), the number of subdivision code for point cloud model are 8.

After obtaining global position subdivision code of point cloud model, the database index table is established with subdivision code is the primary key and the metadata information is other column items, as shown in Table 2. The data is stored in the table to establish the association between subdivision code and the data information, enabling point cloud data to be managed and queried via the subdivision code primary key.

| ID | Column name         | Data type          |
|----|---------------------|--------------------|
| 1  | Id                  | Int                |
| 2  | GridCode            | Varchar(255)       |
| 3  | FileName            | Varchar(20)        |
| 4  | GridLevel           | Char(2)            |
| 5  | TopLeftLatitude     | Double(7,4)        |
| 6  | TopLeftLongitude    | Double(7,4)        |
| 7  | Height              | Double(7,4)        |
| 8  | Box_Length          | Double(7,4)        |
| 9  | Box_Width           | Double(7,4)        |
| 10 | Box_Height          | Double(7,4)        |
| 11 | Filepath            | Varchar(100)       |
| 12 | Urlpath             | Varchar(100)       |

2.2.2. Multi-scale identification of internal points based on Geosot-3D

For the internal data structure of 3D data, a specific or minimal scale grid cell is defined or calculated in GeoSOT-3D subdivision framework to aggregate the internal structure of 3D data. These grid cells are called GeoSOT-3D internal structure code to identify the spatial structure of the 3D data. Geosot-3D internal structure code of 3D data is a more detailed description of the internal structure under the global position code. The purpose is to establish the location association of the internal structure to locate the single internal data block of 3D data files.

For the internal structure of point cloud data which is massive of discrete point data, so select the minimal scale grid cell of GeoSOT-3D subdivision framework, which means selecting 32th grid level and the side length of grid is about 1.5cm. Then internal points data of the point cloud model can be encoded by grid encoding function.

After obtaining multi-scale identification code of internal points, the database index table is established with multi-scale identification code is the primary key and the metadata information is the other column items, as shown in Table 3. The data is stored in the table to establish the association between subdivision code and the data information, enabling internal points data to be managed and queried via the subdivision code primary key.

| ID | Column name | Data type          |
|----|-------------|--------------------|
| 1  | Id          | Int                |
| 2  | GridCode    | Varchar(50)        |
| 3  | GridLevel   | Int                |
| 4  | Gauss_X     | Double(12,6)       |
| 5  | Gauss_Y     | Double(12,6)       |

2.2.2. Multi-scale identification of internal points based on Geosot-3D

For the internal data structure of 3D data, a specific or minimal scale grid cell is defined or calculated in GeoSOT-3D subdivision framework to aggregate the internal structure of 3D data. These grid cells are called GeoSOT-3D internal structure code to identify the spatial structure of the 3D data. Geosot-3D internal structure code of 3D data is a more detailed description of the internal structure under the global position code. The purpose is to establish the location association of the internal structure to locate the single internal data block of 3D data files.

For the internal structure of point cloud data which is massive of discrete point data, so select the minimal scale grid cell of GeoSOT-3D subdivision framework, which means selecting 32th grid level and the side length of grid is about 1.5cm. Then internal points data of the point cloud model can be encoded by grid encoding function.

After obtaining multi-scale identification code of internal points, the database index table is established with multi-scale identification code is the primary key and the metadata information is the other column items, as shown in Table 3. The data is stored in the table to establish the association between subdivision code and the data information, enabling internal points data to be managed and queried via the subdivision code primary key.

| ID | Column name | Data type          |
|----|-------------|--------------------|
| 1  | Id          | Int                |
| 2  | GridCode    | Varchar(50)        |
| 3  | GridLevel   | Int                |
| 4  | Gauss_X     | Double(12,6)       |
| 5  | Gauss_Y     | Double(12,6)       |
Thus, by combining GeoSOT-3D global position for point cloud data and multi-scale identification for internal points data, point cloud data and internal points data can be correlated consistently on the basis of GeoSOT-3D subdivision framework.

2.2.3. The visual representation based on Web system
Cesium is an open source JavaScript library for displaying 3D earth and spatial data, which can be used to display massive 3D model data, image data, terrain elevation data, vector data, etc. In this experiment, JavaWeb and Cesium map engine platform are used to implement point cloud data query and visual expression. Figure 3 shows the flowchart of the developed Web system platform.

![Flowchart](image)

Figure 3. The flowchart of the proposed system

3. Experiment and Results

3.1. Experimental environment and data
The hardware environment for the experiment was Precision 7920 Rack Dell Work Station with 64GB of running memory and MySQL 8.0.18 as the database version. The experimental data were four point cloud data sets with Gauss coordinates, acquired and located in Beijing and Zhejiang Province, China. In order to test the efficiency of the platform based on GeoSOT-3D grid, we collected two simulated point cloud data sets with relative coordinates. According to data description in Table 4, the data size of each point cloud model ranges from 10MB to 1GB, and the number of points inside the point cloud model ranges from 100,000 to 10 million.

| Id | Object                              | Location(Lat, Lng) | Data Size | Number of Points |
|----|-------------------------------------|--------------------|-----------|------------------|
| 1  | Teaching Building of Peking University(Beijing, China) | (116.312,39.996) | 162Mb     | 3 million        |
| 2  | Library of BUCEA(Beijing, China)     | (116.281,39.7451) | 653 Mb    | 5 million        |
| 3  | The rural house 1(Zhejiang, China)   | (119.9026,30.8731) | 1.03GB    | 10 million       |
| 4  | The rural house 2(Zhejiang, China)   | (119.9321,30.8521) | 64.8 Mb   | 2 million        |
| 5  | Ship Model(Simulated)                | (1,1)              | 8.74 Mb   | 269,974          |
| 6  | A building(Simulated)                | (0,0)              | 562Mb     | 8 million        |

The prototype platform was shown in Figure 4. The system can query 3D point cloud data and 2D remote sensing image data released by local server according to grid code, draw GeoSOT and Geosot-3D grid of different scales all over the world or in a local area. The functions of this system include switching image map, transforming 2D and 3D perspectives, measuring the length and area.
3.2. Point cloud data coding and visual experiment based on GeoSOT-3D

1) In the experiment, firstly, Gauss coordinates of each minimum bounding box should be converted to WGS84 coordinates, then we chose an appropriate subdivision scale for point cloud data by comparing the scale relationship between the minimum bounding box length and the grid size. The data in this experiment were buildings of most point cloud data sets, and the side length of the bounding box ranges from 10m to 100m. Therefore, according to the scale of different point cloud data, we selected GeoSOT-3D grid of different scales (20th-22nd level) as the subdivision level, then obtained the global position code by the grid encoding function, and finally we set global position code as the index column of the data table. Then we stored attribute information of the point cloud model in the table as designed in Table 2. The system can query and visualize the corresponding point cloud data according to the global position code, view the information of bounding box and coordinates. Data with real or assumed coordinates can be queried and visualized through grid code. As a result, we have implemented the global position expression of point cloud data, as shown in Figure 5 and Figure 6.

2) We directly selected the smallest subdivision grid (32 levels) of GeoSOT-3D framework as the subdivision level because the internal structure is massive amount of point data, and then obtained the codes of each point data through the grid encoding function. Finally, the codes are set as the primary key and stored in the table designed in table 3 with other attribute information of point data. The system
can query and visualize the internal points data of point cloud model according to the internal structure code, view the information such as grid level, grid code, geographic coordinates and others. We also applied the proposed method with other real-world and simulated data sets to test the multi-scale identification of the internal structure units of the point clouds, as shown in Figure 7.

![Figure 7: Query and visualize internal points based on internal structure code](image)

3) The prototype platform designed based on GeoSOT-3D subdivision framework can not only manage, query and visualize 3D point cloud data, but also implement the above operation for other organizational geographic environment information data (such as 2D remote sensing image data) based on GeoSOT subdivision framework, as shown in Figure 8. Therefore, this system can express the association of different spatial data based on the same system platform.

![Figure 8: Query and visualize remotely sensed imageries based on GeoSOT codes](image)

3.3. Analysis of experimental results

The experimental results of the point cloud data management method based on the GeoSOT-3D grid code include the following aspects.

First, the data are coded with the GeoSOT-3D codes, including the global position code of point cloud data and the multi-scale identification code of internal points.

Second, the metadata information was inserted with the GeoSOT-3D grid codes as the primary key into the designed table. The experiment avoided the problem about inconsistent management of point cloud data.

Third, a 3D prototype platform was developed based on Cesium. We can not only express the relationship between point cloud data and the grid, but also establish association with other spatial data through the grid. The global subdivision grid has geographic location attributes, so geographic location information of the point cloud data can be queried through the grid attributes.

Finally, in addition to point cloud data, the grid codes also have the ability to organize and manage other spatial data. With the increase of data size and type, the platform can efficiently manage various types of data, establish effective associations, promote the sharing and application of various types of data, and improve the efficiency of integrated data management, query and presentation.
4. Conclusion
We proposed a method to organize, manage and visualize point cloud data based on GeoSOT-3D subdivision framework. GeoSOT-3D subdivision framework is seamless, multi-scale and has spatial geographic information. The method in this paper can avoid the problem of inconsistent management and queries by organizing the point cloud model and internal points data according to subdivision grid. A 3D prototype platform was developed based on Cesium to implement the global position of point cloud data and multi-scale identification of internal points. Taking point cloud models in Beijing and Zhejiang, China as an example, the efficient management, query, visualization of point cloud data is implemented, and the correlation with other spatial data in global geographic scene further verifies the effectiveness of the proposed method.

We still need to consider how to integrate more kinds of spatial data and improve the retrieval efficiency. In addition, the implementation of associated management about the point cloud model in the system and the corresponding building in the geographical entity is an important research direction in the future. These future studies will refine our approach to make it more comprehensive and applicable.

Acknowledgement
This research was funded by the National Key R&D Program of China (2018YFB2100700 or 701-705), the Research Program of Beijing Advanced Innovation Center for Future Urban Design (UDC2019031321), The Pyramid Talent Training Project of Beijing University of Civil Engineering and Architecture (JDYC20200322), and the National Natural Science Foundation of China (41601389).

References
[1] Wand, M., Berner, A., Bokeloh, M., et al. (2008) Processing and interactive editing of huge point clouds from 3D scanners. Computers & Graphics, 32: 204-220
[2] Gobbetti, E., Marton, F. (2004) Layered point clouds: a simple and efficient multiresolution structure for distributing and rendering gigantic point-sampled models. Computers & Graphics, 28: 815-826
[3] Li, J., Fang, H.Y., Ma, Y.R., et al. (2015) Research on Point Cloud Data Management Based on Spatial Index and Database. Advanced Materials Research, 850-851: 685-688
[4] Baek, N., Yoo, K.H. (2019) An accelerated rendering scheme for massively large point cloud data. Journal of Supercomputing, 5: 1-11
[5] Gong, J., Zhu, Q., Zhong, R., et al. (2012) An Efficient Point Cloud Management Method Based on a 3D R-Tree. Photogrammetric Engineering Remote Sensing, 74: 373-381
[6] Zhang, H., Jing, M. (2011) The Organization and Visualization of Point Cloud Data Based on the Linear Quadtree. Journal of Taiyuan Normal University, 144: 425-432
[7] Van Oosterom, P., Martinez-Rubi, O., Ivanova, M., et al. (2015) Massive point cloud data management: Design, implementation and execution of a point cloud benchmark. Computers & Graphics, 49: 92-125
[8] Poux, F., Hallot, P., Neuville, R., et al. (2016) Smart Point Cloud: Definition and Remaining Challenges. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2/W1: 119-127
[9] Axelsson, P. (1999) Processing of laser scanner data—algorithms and applications. ISPRS Journal of Photogrammetry and Remote Sensing, 54: 138-147
[10] Li, S., Cheng, C., Chen, B., et al. (2016) Integration and management of massive remote-sensing data based on GeoSOT subdivision model. Journal of Applied Remote Sensing, 10: 034003-1-15
[11] Zhai, W., Tong, X., Miao, S., et al. (2019) Collision Detection for UAVs Based on GeoSOT-3D Grids. ISPRS International Journal of Geo-Information, 8: 299-313
[12] Zhang, H., Cheng, C., Miao, S. (2019) A Precise Urban Component Management Method Based on the GeoSOT Grid Code and BIM. ISPRS International Journal of Geo-Information, 8: 159-176