Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8217220, 6 pages
https://doi.org/10.1155/2022/8217220

Research Article

A Fast Scheduling Method for Massive Oblique Photography 3D Models

Ruibo Chen,[1,2] Ruichao Qu,[1,2] Jiaxing Chen,[1,2] Jiangtao Lei,[1,2] Yitan Luo,[1,2] Rundong Liu ,[1,2] and Zhanyu Ma[3]

[1]*Guangxi Zhuang Autonomous Region Natural Resources Remote Sensing Institute, Nanning, Guangxi 530023, China*
[2]*China-ASEAN Key Laboratory of Satellite Remote Sensing Application, Ministry of Natural Resources, Nanning, Guangxi 5300230, China*
[3]*State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China*

Correspondence should be addressed to Rundong Liu; 18345443845@163.com

Received 30 May 2022; Accepted 29 July 2022; Published 30 August 2022

Academic Editor: Chia-Huei Wu

In view of the large amount of oblique photography 3D model data and the fact that high-performance hardware devices cannot load and render massive oblique photography 3D models at one time, this paper provides a fast scheduling method for massive oblique photography 3D models: get the tile radius, side length, scheduling range, and other parameters from the original data, normalize the original tile to a hundred grid, use a regular grid + quadtree + hexadecimal tree to build a tile spatial index, resample the tile data at each level to generate the elevation model DEM and the real radiographic image TDOM, and use the DEM+TDOM method to reconstruct the layered and block top-level 3D model to achieve rapid scheduling and rendering of massive oblique photography 3D models. Experiments show that the method is efficient and feasible and can meet the visualization and real-time interaction requirements of massive urban 3D models.

1. Introduction

The Ministry of Natural Resources has issued the "Technical Outline for Real 3D China Construction," which clarifies that the construction content of real 3D China includes three levels of real 3D construction: terrain level, city level, and component level [1], with city-level real 3D model data. Generally, oblique photogrammetry technology is used for production, and the amount of 3D model data of the scene can reach TB level or even massive, far exceeding the storage and management capabilities of ordinary computers. At the same time, it puts forward higher requirements for the organization and scheduling of massive 3D data. Due to the complex data structure of the 3D model and the large amount of data, it often causes difficulties in transmission and rendering, which in turn affects the visualization efficiency. Therefore, it is necessary to establish a reasonable 3D model organization structure, management, and scheduling method to optimize the model calling mechanism and accelerate the loading and operation of 3D models, the process of rendering and thus visualization. In the case of massive 3D data, the effect of the above optimization is more obvious. At present, methods such as improving quadtree [2–8], octree, KD tree, and three-dimensional R tree are commonly used or combined (referenced) to organize construction and scheduling.

At present, GIS technology is developing rapidly, but how to meet the application of massive 3D data is an urgent problem to be solved. Oblique photography technology has its obvious advantages, that is, it can meet the needs of rapid acquisition and rapid modeling of large-scale 3D data. It has high precision, low cost, high efficiency, and real effects. It will be very promising in projects such as smart city construction, but how to apply massive 3D data conveniently and efficiently is its pain point.

With the development of computer technology, the use of high-performance hardware equipment to load and render oblique photography 3D model data has played a certain role, but it is still impossible to load and render city-level 3D model data at one time [9, 10]. The model data organization method is optimized, and a new spatial index is constructed by using the regular segmentation method. Move and load the tilted model within a certain range. This method has better effect when loading a small scene range, but in a large scene range, more tile files need to be loaded to realize the display of the entire large scene, thus affecting the efficiency of scheduling and system performance. In response to these problems, this paper provides a fast scheduling method for massive 3D models of oblique photography: take the tile in the 3D data of oblique photography as the basic unit, obtain parameters such as center point, LOD (levels of detail) scheduling range, specify the number of levels, and use the method of combining regular grid + quadtree and hexadecimal tree that hierarchically grade all tiles according to regular grid, and resample the tile data of each level to generate an elevation model DEM and a real projection image TDOM. The top-level 3D model is reconstructed in layers and blocks, and the rapid scheduling and display of inclined 3D model data in a large scene range are realized.

2. Scheduling Method of Massive Oblique Photography 3D Model

2.1. Obtaining the Original Parameters. The tile pyramid model is a common multiresolution hierarchy structure, which is often used in 2D maps and 3D terrain services. The change of the visible field range dynamically selects the tiles suitable for the resolution to load.

During the data production process, the oblique photography 3D model will generate metadata files according to the actual location of the survey area and organize the oblique photography 3D model data in a folder formed by the tile + grid row and column number that stores the 3D model data of different levels in the grid unit. Based on the metadata file and the 3D model data at each level in the tile folder, the original parameters of the oblique photography 3D model are calculated and obtained: radius dR, side length dS, and top-level LOD scheduling range dRange. On the basis of reading the original parameter data, calculate the distance coefficient n of LOD (levels of detail) according to the following formula:

$$2^{\mathrm{n}} = \frac{dRange}{dR}. \quad (1)$$

Among them, n is the LOD distance coefficient of the tile, dRange is the scheduling range of the top LOD, and dR is the tile radius.

2.2. Organization of Oblique Photography Model Based on Regular Grid + Quadtree + Hexadecimal Tree. Spatial index, also known as spatial access method, is a description of data location information stored on the medium and refers to a data structure arranged in a certain order according to the location and shape of spatial objects or a certain spatial relationship between spatial objects, which contains general information about the spatial object, such as the object's identifying bounding rectangle and a pointer to the spatial object entity. As an auxiliary spatial data structure, the spatial index is between the spatial operation algorithm and the spatial object. It improves the speed and efficiency of the spatial operation by excluding the spatial objects that are not related to the specific spatial operation. In recent years, one of the main functions of the rapid development of geographic information system is to retrieve spatial data and query the required information. As the key technology of spatial database and geographic information system, the performance of spatial index technology will directly affect the storage efficiency of spatial data and the performance of spatial retrieval, which is related to the overall performance of geographic information system. There are two commonly used object segmentation methods in GIS at present, the regular segmentation method and the object-based segmentation method. The spatial indexes corresponding to these two partitioning methods are spatial indexes based on regular grids (grid indexes, BSP tree indexes, and quadtree indexes) and spatial indexes based on object partitioning (R-tree family indexes). Among these index structures, grid spatial index, quadtree spatial index, and R-tree series spatial index are the most common. Efficient and rapid planning and establishment of a spatial index of real 3D model data is an important way to improve the rapid retrieval efficiency of oblique photography 3D model data.

The method of establishing spatial index based on grid or R-tree, quadtree, etc. is widely used in the management of spatial data such as terrain and images [11], but in the 3D model of oblique photography, the index is only established inside the tile and the massive. In the process of scheduling and loading of oblique photography 3D model data, all tiles are only loaded in series, which cannot meet the needs of fast loading in practical applications. Therefore, this paper combines the convenience of regular grid division and the advantages of quadtree and hexadecimal in constructing spatial index in oblique photography model [12, 13], using regular grid + quadtree + hexadecimal tree way to construct a spatial index of oblique photographic models.

2.3. Building a Tile Hierarchical Block Grid Dictionary. The 3D model data of oblique photography is usually stored in the form of tile folders, and each tile stores the model data of various levels. During data production, the smaller the grid size used, the more tiles will be. Considering the actual situation of production operations, the grid size is usually between 100 m and 200 m. In the data scheduling of the oblique photography 3D model, the grid of the original data needs to be further hierarchically gridded to meet the needs of dynamic scheduling. The specific process of hierarchical gridization of the original tile in this paper is shown in Figure 1.

(1) The specified level grid contains the original number of tiles, calculates the size of the new grid, and normalizes it

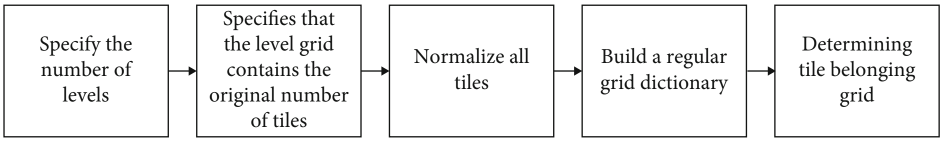
Specify the number of levels
Specifies that the level grid contains the original number of tiles
Normalize all tiles
Build a regular grid dictionary
Determining tile belonging grid

Figure 1: Process for building tile grid fields.

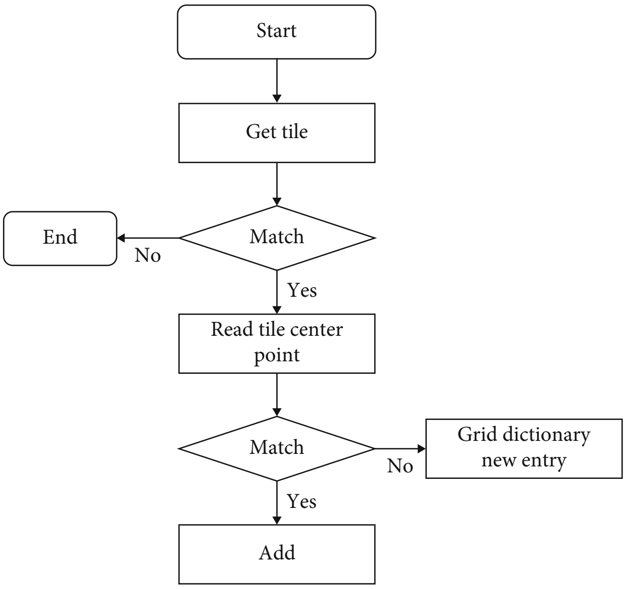

Figure 2: Regular grid establishment process.

According to the tile side length dS of the original data and the number of original tile contained in the grid n^2, calculate the grid size according to formula (2):

$$dGrideSide = dS \times n. \tag{2}$$

Among them, dGridSide is the grid size, dS is the original tile side length, and n is the base number of the original grids contained in the grid.

The grid size dGridSide is shown in formula (3), so that the grid is an integer multiple:

$$dGridSide = INT\left(\frac{\left(\mathrm{n} \times \sqrt{2 \times \mathrm{d}Range^2}\right) + 50}{100}\right) \times 100. \tag{3}$$

Among them, DGridSide is the grid size, dRange is the original Tile scheduling range, and n is the base number of the original grids included in the grid.

(2) Establish a hierarchical block grid dictionary to determine the original Tile attribution

Loop through all the original tile data, read the center point of each tile data, and determine the tile grid to which the tile center point belongs. If there is no corresponding tile grid in the tile grid dictionary, add an entry in the dictionary, such as follows: If there is, add it. The specific process is shown in Figure 2:

2.4. Quadtree + Hextree Algorithm to Build Spatial Index. The quadtree index is a tree-structured data index proposed by Tayeb in 1998. Its principle is to divide a rectangular area into four subareas, and each subarea is divided into quarters again, recursively from top to bottom and operates until the number of elements contained in each subregion is less than or equal to the specified capacity. The quadtree has obvious spatial characteristics. According to the spatial characteristics of the quadtree, it can be used to make a spatial index.

In the process of tile gridization and building a hierarchical block grid dictionary, the grid to which each tile belongs to has been determined. The hierarchies use different methods to build spatial indexes. This paper divides the hierarchy into three stages: the top layer, the middle layer, and the bottom layer. The top layer uses a regular grid to build a spatial index, and the number of grids on the 8 × 8 top layer is the way to build a hexadecimal tree. The specific determination method of each stage is calculated by formula (4):

$$\begin{cases} \mathrm{TopLy} = 2^x, x = 0 \\ MiddleLy = 2^x, 1 \le x < 2^{n/2} & , n \text{ is } a\ multiple \text{ of } 2 \\ BottomLy = 2^x, 2^{n/2} \le x \le n. \end{cases} \tag{4}$$

Among them, TopLy is the top-level range, MiddleLy is the middle level range, BottomLy is the bottom level range, x is the number of layers of the current layer, and n is the total number of layers of the layer.

2.5. DEM+TDOM Reconstruction Hierarchical Block Top Model. After the original oblique photographic data tile is layered and gridded, in order to speed up the fast loading of a certain block of data scheduled to the corresponding level, this paper rewrites the layered and divided tile data on the basis of the layered and divided grid. Build the top-level model data, so that after scheduling to a grid at the corresponding level, the top-level model data of the grid can be quickly scheduled and loaded, increasing the buffer time for scheduling more detailed model data and improving the smooth rendering of massive 3D model data. Based on the spatial index, the top-level model of the hierarchical block grid is reconstructed. The specific process is shown in Figure 3.

(1) Read the value in the block grid dictionary, and fit the radius dBlockR of the block grid by the least squares fitting method

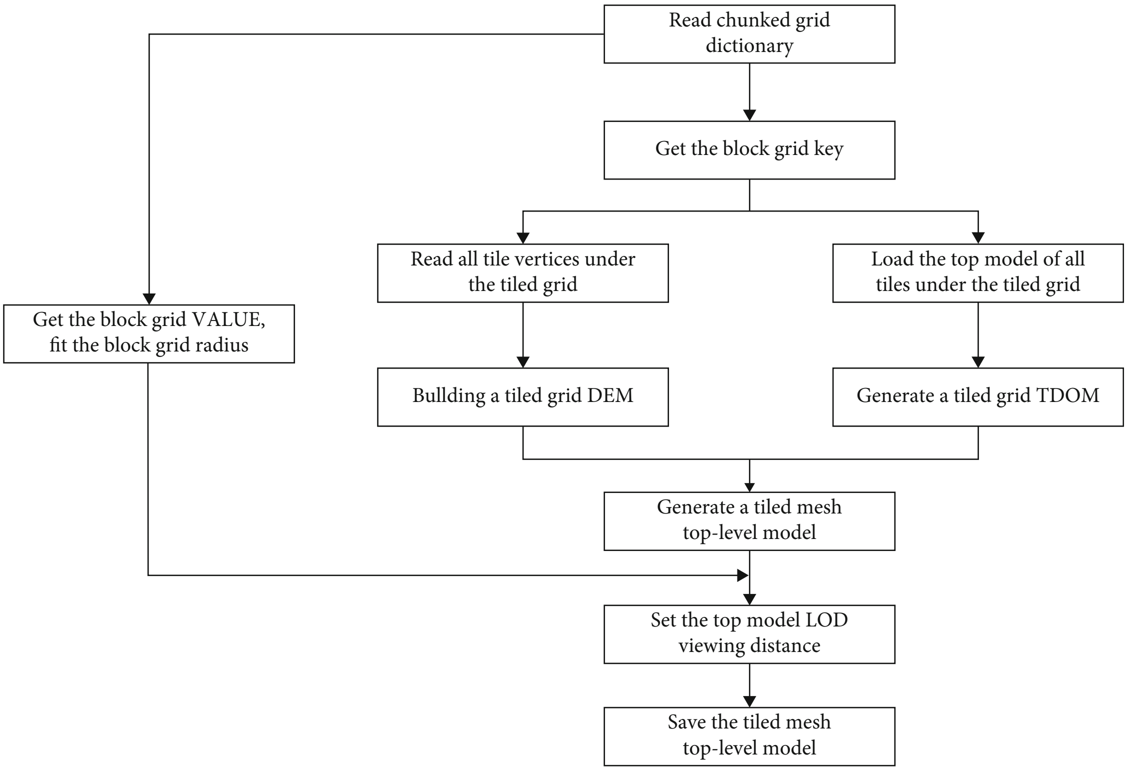
Read chunked grid dictionary
Get the block grid key
Read all tile vertices under the tiled grid
Load the top model of all tiles under the tiled grid
Get the block grid VALUE, fit the block grid radius
Bullding a tiled grid DEM
Generate a tiled grid TDOM
Generate a tiled mesh top-level model
Set the top model LOD viewing distance
Save the tiled mesh top-level model

Figure 3: Reconstruction of layered and block top-level model.

Table 1: Introduction of experimental data.

Data area	Amount of data (unit: GB)	Number of tiles
City 1	65.25	2880
City 2	71.27	5254
City 3	94.47	6392
City 4	136.64	11760

Table 2: Load time/memory usage comparison.

Data area	Load time (unit: milliseconds)		Occupied memory (unit: MB)	
	Normal method	The method of this paper	Normal method	The method of this paper
City 1	10782	1052	885.6	356.4
City 2	18480	2895	1308.4	458.9
City 3	23590	3570	1393.5	479.7
City 4	41285	6955	1807.5	692.1

(2) Resample to generate a tiled grid DEM (digital elevation model). Read the vertices of all tile data under the grid, merge to generate a discrete point array, $100m \times 100m$, resample the grid, calculate the elevation value of the corresponding grid point, and generate a block grid DEM file

(3) Generate a tiled grid TDOM (true projection image). Load the merged tile top-level data into the scene, set the pitch angle of the camera in the 3D scene to -90°, set the projection relationship to orthographic projection, and save the effect content in the scene to obtain a real projection image

(4) The top-level model of mesh construction and graphics is composed of block meshes. The DEM generated in (2) is used to construct an irregular triangular network as a skeleton, and the real projective image generated in (3) is textured to generate top-level model data

(5) Calculate the LOD viewing distance dEye of the top model data according to formula (5), and set it to the top model

$$dEye = dBlockR \times 2^{n/2}. \tag{5}$$

Among them, dEye is the LOD viewing distance, DBlockR is the block grid radius, and n is the level

(6) Save the hierarchical block top model data

3. Experimental Results and Analysis

This paper selects the oblique photography 3D data of four main urban areas of a city in Guangxi Zhuang Autonomous Region to verify and test the fast scheduling method provided in this paper. This paper selects the oblique photography 3D data of four main urban areas of a city in Guangxi Zhuang Autonomous Region to verify and test the fast scheduling method provided in this paper. The specific data

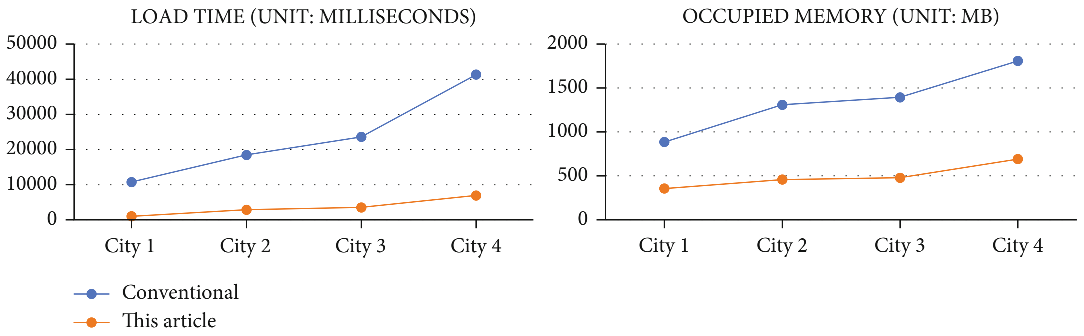

Figure 4: Load time/occupied memory comparison chart.

Table 3: Comparison of domestic and foreign mainstream three-dimensional platforms and the scheduling method in this paper.

Platform	Data is displayed for the first time Time (ms)	All data Display time (ms)	Data scheduling Efficiency	Data Clarity	Memory usage (MB) when data is clear
The method of this paper	100	7000	Faster	Keep it original	700
A domestic platform 1	100	35000	Generally	Keep it original	1950
A domestic platform 2	100	12730	Faster, scheduling stops when switching locations	Keep it original	1630
a foreign platform	100	50800	Slower	Clear when close	1450

is shown in Table 1. To validate our method, all validation tests are performed on the same desktop computer (CPU: Intel i7 8-core 16-thread; graphics card: NVIDIA Quadro P5000; memory 32 GB; solid-state disk: 4 TB).

In this paper, the comparison between the conventional method and the method in this paper in terms of loading time and memory usage and the comparison between the method in this paper and the mainstream 3D platforms at home and abroad in terms of scheduling efficiency, loading time, and display clarity are shown, and the specific experimental analysis is as follows:

(1) The conventional method and the method in this paper are compared in terms of loading time and memory usage

From Table 2 and Figure 4, the loading time and the specific values of memory occupied, as well as the line graph, it can be seen that the scheduling loading time of the method in this paper is only about 1/5 of that of the conventional method, and the memory usage is only 2/5 of that of the conventional method. Left and right, the efficiency of scheduling and loading of oblique photography 3D models has been greatly improved

(2) In order to compare and analyze the scheduling method in this paper and the scheduling display of mainstream 3D platforms at home and abroad

From Table 3, we can compare and analyze the same data of mainstream 3D platform scheduling at home and abroad. It can be seen that the method in this paper has obvious advantages in terms of scheduling time, efficiency, data display clarity, and memory usage and can realize the optimization of massive inclined 3D models, fast scheduling and display of data.

4. Conclusion

The 3D model of the city is the basis for constructing a digital city. Due to its different modeling methods and different usage environments, there is currently no unified 3D model standard for cities [14–18]. Regardless of the format, the 3D model has the characteristics of complex data structure and huge amount of data, which poses a huge challenge to data storage, transmission, and especially the visualization process. Therefore, it is necessary to design a reasonable and efficient 3D model data organization and management method to realize the rapid visualization of model data. This paper conducts research on 3D model data acquisition, preprocessing, organization, storage, management, scheduling methods, rendering, etc., and according to experiments [19–21], it is verified that the method in this paper can improve the scheduling and rendering efficiency of massive oblique photography 3D models.

Based on the research and analysis of the existing oblique photography 3D model data organization and scheduling methods, based on tile, a regular grid + quadtree + hexadecimal tree is used to construct a spatial index to organize and manage tile data. On this basis, the layered and block top-level DEM is reconstructed by regular grid sampling, the top-level TDOM is obtained by orthographic projection, and the top-level 3D model of the layered block is reconstructed by using DEM+TDOM to realize fast scheduling of massive oblique photography 3D models. The advantages of this method in the rapid scheduling of massive oblique photography 3D models are verified by experimental comparison and analysis, and the application in the self-developed 3D platform has achieved good results, but there are still some problems [22–27], such as the first loading. It takes a long time to build the spatial index and the top-level 3D model when using the data, and further research and improvement are required.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

We declare that there is no conflict of interests in this paper.

References

[1] Ministry of Natural Resources, *Real 3D China Construction Technology Outline (2021 Edition): Natural Resources Ban Fa [2021] No. 56 [N]*, 2021.

[2] Q. Y. Chen, G. Liu, X. G. Ma et al., "Local curvature entropy-based 3D terrain representation using a comprehensive quadtree," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 139, pp. 30–45, 2018.

[3] C. Y. Yao, G. H. Peng, Y. L. Song, and M. Duan, "A quadtree organization construction and scheduling method for urban 3D model based on weight," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, p. 42, 2017.

[4] L. Daoyuan, L. Yingcheng, and X. Jincheng, "Research on management of the massive 3D city models," *Science of Surveying and Mapping*, vol. 36, no. 5, pp. 70–72, 2011.

[5] L. Daoyuan, L. Yingcheng, and X. Jincheng, "Large-scale city research on 3D model management technology," *Surveying and Mapping Science*, vol. 36, no. 5, pp. 70–72, 2011.

[6] W. Yang, *Data Schedule and Scene Management of Huge 3D City Model*, Wuhan University, Wuhan, 2005.

[7] F. Wang, D. Pan, and J. Wang, "Dynamic dispatching and organization of massive data of urban 3D model," *Journal of University of Chinese Academy of Sciences*, vol. 32, no. 3, pp. 409–415, 2015.

[8] W. Feng, P. Deji, and W. Jun, "Dynamic organization and scheduling method for massive data of urban 3D models," *Journal of University of Chinese Academy of Sciences*, vol. 32, no. 3, pp. 409–415, 2015.

[9] S. Deng, Y. Wang, and H. Dou, "Dynamic scheduling of three-dimensional massive model data based on databasepager," *Science of Surveying and Mapping*, vol. 38, no. 4, pp. 97–100, 2013.

[10] S. Deng, Y. Wang, and H. Dou, "Dynamic scheduling of massive 3D data models with data paging technology," *Science of Surveying and Mapping*, vol. 38, no. 4, pp. 97–100, 2013.

[11] Y. Chaode and Z. Xuesheng, "Geography and geo-information science," *The Review of Spatial Indexes in GIS*, vol. 20, no. 4, pp. 23–27, 2004.

[12] C. Yongkang, *Research on Visualization of Large Data Scene in 3D GIS*, Institute of Remote Sensing and Digital Earth Chinese Academy of Sciences, Beijing, 2004.

[13] Y. Chunyu, P. Guihui, and D. Mengqi, "Research and implementation of key techniques for 3D rendering of large scene," *Geospatial Information*, vol. 17, no. 10, pp. 96–98, 2019.

[14] L. Rundong and C. Ruibo, "Data scheduling method and system for massive oblique photography 3D model," China, ZL02010380307.4.2020-05-28.

[15] Z. Xiaowei, "Research on data management and application of massive oblique photography 3D model based on root node aggregation technology," *Surveying and Mapping and Spatial Geographic Information*, vol. 44, no. S1, pp. 209–212, 2021.

[16] T. Shanshan, *Research on Rapid Generation and Display Optimization of Large-Scale 3D Terrain with Multi-Resolution*, Southeast University, 2018.

[17] S. Yang, L. Changhui, and S. Shuang, "Research and application of block storage of massive laser point cloud based on entity model index," *Surveying and Mapping Engineering*, vol. 25, no. 4, pp. 7–10, 2016.

[18] W. Guoniu, "Urban 3D pipe network construction and efficient visualization," *Surveying and Mapping Science*, vol. 40, no. 4, pp. 67–70, 2015.

[19] F. Yan, R. Guo, M. Wang, and G. Xingye, "Research on data organization and management method of 3D urban model," *Surveying and Mapping Science*, vol. 36, no. 1, pp. 215–217, 2011.

[20] W. Bao, X. Zhu, and L. Xu, "Data organization and management method of multi-source heterogeneous city 3D model," *Modern Surveying and Mapping*, vol. 44, no. 6, pp. 34–36, 2021.

[21] L. Tianyi, *Research on Data Management and Organization of Urban 3D Model for Rapid Visualization*, National University of Defense Technology, 2017.

[22] H. Cheng, H. Liang, Z. Xiaolong, Z. Wang, and D. Yongge, "Research on entity-oriented 3D city model data organization method," *Surveying and Mapping and Spatial Geographical Information*, vol. 40, no. 3, pp. 74–77, 2017.

[23] Y. W. Chaokui, Y. Wu, and C. Guo, "Data division and distributed storage method of 3D city model," *Journal of Earth Information Science*, vol. 17, no. 12, pp. 1442–1449, 2015.

[24] W. Mingguang, "A spatial index driven by spatial distribution pattern," *Journal of Surveying and Mapping*, vol. 1, pp. 108–115, 2015.

[25] P. Babahajiani, L. Fan, J.-K. Kämäräinen, and M. Gabbouj, "Urban 3D segmentation and modelling from street view images and LiDAR point clouds," *Machine Vision and Applications*, vol. 28, no. 7, pp. 679–694, 2017.

[26] M. Bremer, A. Mayr, V. Wichmann, K. Schmidtner, and M. Rutzinger, "A new multi-scale 3D-GIS-approach for the assessment and dissemination of solar income of digital city models," *Computers, Environment and Urban Systems*, vol. 57, pp. 144–154, 2016.

[27] J. Chen, J. Li, and M. Li, "Progressive visualization of complex 3D models over the internet," *Transactions in GIS*, vol. 20, no. 6, pp. 887–902, 2016.