DOI: 10.1111/tgis.12638

RESEARCH ARTICLE

Transactions in GIS WILEY

Web-based real-time visualization of large-scale weather radar data using 3D tiles

Mingyue Lu[1,2] | Xinhao Wang[1,2] | Xintao Liu[3] | Min Chen[4,5,6,7] | Shuoben Bi[2] | Yadong Zhang[1,2] | Tengfei Lao[1,2]

[1]Collaborative Innovation Center on Forecast and Evaluation of Meteorological Disasters, Nanjing University of Information Science & Technology, Nanjing, China

[2]School of Geographical Sciences, Nanjing University of Information Science & Technology, Nanjing, China

[3]Department of Land Surveying and Geo-Informatics, Hong Kong Polytechnic University, Kowloon, Hong Kong

[4]Key Laboratory of Virtual Geographic Environment, Ministry of Education, Nanjing Normal University, Nanjing, China

[5]State Key Laboratory Cultivation Base of Geographical Environment Evolution (Jiangsu Province), Nanjing, China

[6]Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing, China

[7]State Key Laboratory of Cryospheric Sciences, Northwest Institute of Eco-Environment and Resources, Chinese Academy of Sciences, Lanzhou, China

Correspondence

Xintao Liu, Department of Land Surveying and Geo-Informatics, Hong Kong Polytechnic University, 181 Chatham Road South, Kowloon, Hong Kong.
Email: xintao.liu@polyu.edu.hk

Funding information

National Natural Science Foundation of China, Grant/Award Number: 41871285; Academic Program Development of Jiangsu Higher Education Institutions, Grant/Award Number: 164320H116

Abstract

Weather radar data play an important role in meteorological analysis and forecasting. In particular, web-based real-time 3D visualization will enable and enhance various meteorological applications by avoiding the dissemination of a large amount of data over the internet. Despite that, most existing studies are either limited to 2D or small-scale data analytics due to methodological limitations. This article proposes a new framework to enable web-based real-time 3D visualization of large-scale weather radar data using 3D tiles and WebGIS technology. The 3D tiles technology is an open specification for online streaming massive heterogeneous 3D geospatial datasets, which is designed to improve rendering performance and reduce memory consumption. First, the weather radar data from multiple single-radar sites across a large coverage area are organized into a spliced grid data (i.e., weather radar composing data, WRCD). Next, the WRCD is converted into a widely used 3D tile data structure in four steps: data preprocessing, data indexing, data transformation, and 3D tile generation. Last, to validate the feasibility of the proposed strategy, a prototype, namely Meteo3D at https://202.195.237.252:82, is implemented to accommodate the WRCD collected from all the weather radar sites over the whole of China. The results show that near real-time and accurate visualization for the monitoring and early warning of strong convective weather can be achieved.

1 | INTRODUCTION

It is generally agreed that the damage caused by meteorological disasters can be alleviated by early warning based on the analysis of various weather radar data products. In particular, the 3D visualization of spatiotemporal characteristics of weather radar data can provide rich information to facilitate the monitoring and prediction of strong convective weather with high accuracy of meteorological services. So far, much effort has been dedicated to weather disaster warning and monitoring using weather radar data, which is successful to some extent (Bell & Moore, 1998; Tilford, Fox, & Collier, 2002; Wilson, 1970). With the support of GIS, 3D visual analytics has been one of the most widely used technologies for analyzing and presenting climate simulations and observations, as well as related social and ecological data (Chen & Lin, 2018; Dobesch, Dumolard, & Dyras, 2013; Goodchild, 2018; Nocke, Sterzel, Böttinger, & Wrobel, 2010; Titov, Gordov, Okladnikov, & Shulgina, 2009). However, most early studies on the meteorological radar data products are still constrained to using 2D data structures to represent 3D cloud states (Liu et al., 2017). Obviously, the 2D projected method may neglect considerable yet valuable information that is hidden in the 3D weather radar data, thus causing barriers to data analysis and potential applications (Lu et al., 2018). The 3D modeling and visualization of meteorological data can be dated back to the late 1980s (Liang & Huang, 2007) and 1986 Hibbard (1986). However, the applicable 3D visual analytics of large-scale meteorological data is still in its preliminary stage (Choi, Kang, Kim, Kim, & Choung, 2015; James, Brodzik, Edmon, Houze, & Yuter, 2000; Miller, 1996; Zhang, Liu, & Wang, 2010). First, as mentioned above, despite the nature of large-scale weather radar data, most related work was limited to a small-scale dataset with limited technology support in the early days. Second, since it is hard to disseminate the large amount of weather radar data from the data collection server to remote clients over the web, it is hard to perform web-based visual analytics of various large-scale weather radar data in a real-time manner. Most traditional visualizations of weather radar data involve downloading original data from radar sites to local users, which can cause delays in meteorological analyses and applications. With the recent development of web GIS and other computing technologies (e.g., 3D tiles), the real-time 3D visualization of large-scale weather radar data is now viable.

The 3D tiles technology is a set of open specifications for streaming massive heterogeneous 3D geospatial datasets (Anon, 2019a). The primary purpose of 3D tiles is to improve streaming and rendering performance of massive heterogeneous datasets over the internet (Cozzi, 2015), which makes web-based large-scale 3D model data visualization possible. Although 3D tiles have only been proposed in recent years, they have received wide attention from researchers in various fields due to its advantages in supporting the visualization of large-scale 3D data. 3D tiles represent 3D spatial content including buildings, trees, point clouds, and vector data with built-in models, which can provide efficient 3D visualization (Chen, Shooraj, Rajabifard, & Sabri, 2018; Gan, Li, & Jing, 2017; Haje, Jessel, Gaildrat, & Sanza, 2016; Kulawiak, 2016; Kulawiak & Kulawiak, 2017). With the success of 3D tile applications, recent studies started to modify and extend the built-in data models based on its open specifications in different fields, for example, the visual display of flood data (Herman, Russnák, & Řezník, 2017) and the optimization of the 3D tiles data format (Song & Li, 2018). Although there are many studies on 3D tiles, to the best of our knowledge, no researchers have attempted to use 3D tiles to visualize large-scale weather radar data in a real-time manner.

Meanwhile, the advances in weather detection and data storage technologies have greatly increased the amount of meteorological data such as weather radar composing data (WRCD), which is collected from multiple single radar sites and organized into spliced grid data. The WRCD enables the analysis of large amounts of meteorological data and brings challenges in many aspects, such as its complex data structure. Thanks to the rapid development of web-based GIS, it is now feasible to perform 3D visualization of large-scale meteorological data in a real-time manner (Abimael, Scheer, & Sato, 2012; Lu, Chen, Wang, Min, & Liu, 2017). In order to make full use of the 3D information of the WRCD, this article aims to propose a new framework to perform real-time 3D visualization of the WRCD taking advantage of 3D tiles and WebGIS. A case study is implemented using all the weather radar data across the whole of China, and the results demonstrate its efficiency and effectiveness in the real-time 3D visual analytics of large-scale WRCD.

The remainder of this article is structured as follows. The basic concepts and characteristics of weather radar and weather radar data are introduced in Section 2. The weather radar data visualization scheme using 3D tiles is introduced in detail in Section 3. In Section 4, the specific WRCD from the whole of China is used to process the visualization experiment using the above mentioned method, and then the results of the experiment are analyzed and discussed. Section 5 draws conclusions and suggests future studies.

2 | WEATHER RADAR AND COMPOSING DATA

Weather radar is one of the main devices for the monitoring and early warning of strong convective weather. Weather radar has a short period, a wide coverage, and the ability to observe the 3D information of strong convective weather, providing high-precision data for the monitoring and early warning of severe weather disasters. As shown on the left of Figure 1, weather radar emits a series of pulse electromagnetic waves, and then receives echo electromagnetic waves at the spatial interval of 1 km. The received echo waves can reflect how their waves are scattered and absorbed by precipitation particles such as clouds, rain, and snow, based on which the spatial distribution and vertical structure of precipitation can be detected (Yu et al., 2006). Conventional weather radar can reach weather conditions within a radius of 400 km, and it is obvious that such a range is not enough for the observation and analysis of weather changes across large areas. To solve this problem, weather radar data from multiple radar sites are combined and organized into a grid data structure (i.e., WRCD).

To create the WRCD, the weather radar echo data is interpolated into grid data on the same horizontal plane (right of Figure 1). The grid data records the first grid point coordinate value, cell size, and the attribute value of each grid to express the weather radar detection results of the grid coverage area. In order to record the change in weather conditions over the elevation, the WRCD puts together grids from different horizontal planes to express their structure in the vertical direction. In general, a complete WRCD contains multiple layers of similarly structured grid data layers. Each layer of grid data consists of a large number of columns and rows, and the cell size represents the size of each mesh. The large-scale multilayer WRCD produces a large number of grids with many weather radars, which brings opportunities as well as challenges in real-time 3D visualization. For instance, to accurately express the geometry of the grid data, the grids needs to be drawn one by one. When processing visualization, the mesh is processed into a quadrilateral patch with color and then is drawn patch by patch. Although the entire drawing process seems simple, as the number of patches increases, the pressure on the 3D graphics drawing of the web browser gets heavier, and the drawing speed gets slower and slower. When the number of patches reaches a certain level, the browser may crash. Therefore, a viable 3D visualization scheme for large-scale WRCD needs to be proposed.

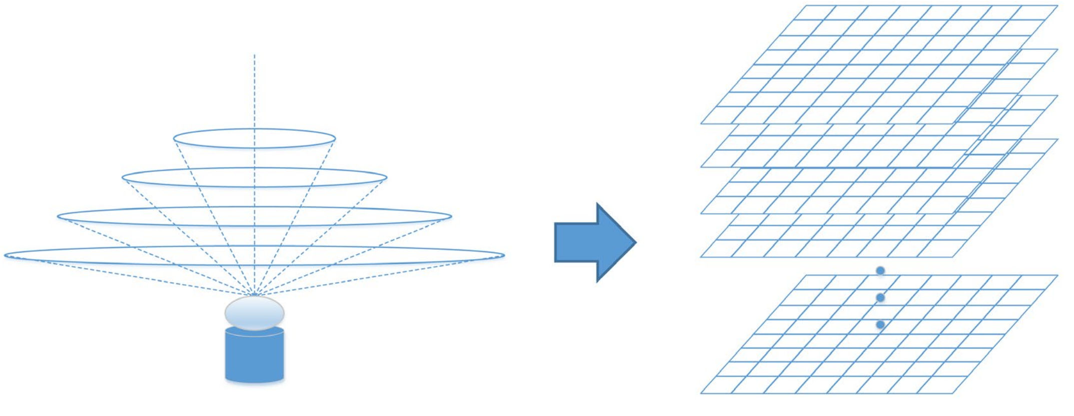

FIGURE 1 Weather radar and composing data structure

3 | THE PROCESS AND VISUALIZATION OF WRCD

This section describes details of the proposed 3D visualization strategy, which is divided into four parts (Figure 2). The first step is the preprocessing of levels of detail through data layering and data segmentation. The second step is to construct an index structure for the parsing data. The third step is to transform the data for 3D tiles data. Finally, the last step is to generate 3D tiles data which is used to process the 3D visualization of WRCD.

3.1 | WRCD preprocessing of levels of detail

3.1.1 | Data parsing and layering

The first step in conventional weather radar visualization is data parsing. However, unlike the conventional scheme, in this work, the data parsing in the proposed scheme is data layering processing of WRCD. Data layering is divided into two parts: layering in height and layering in precision. The layering in height is to express the multilayer structure of the WRCD, which shows the weather conditions in the vertical direction. The data layering in precision is actually to express the same grid data with different precision to show the different levels of detail of the WRCD. The advantage of this method is that data within the range can be expressed with different precisions at different viewing angles to reduce visualization pressure.

Since each layer of the WRCD has the same grid structure at the same horizontal plane, no further layering processing is required in the vertical direction, and the layering of the original data is used directly for layering in height, which will generate multiple grid layers. The purpose of data layering in precision is to display different scenes with different precision, which requires different precision forms of the same data. Therefore, in order to complete this data layering, it is necessary to extract different precision information for the data. WRCD is large-scale grid data, which means that different levels of precision are represented by different grid sizes. Therefore, there are two ways to perform data layering: one is to perform data analysis first, and then the data is layered; the other is to directly divide the data into multiple data of different precisions by multiple different resolution

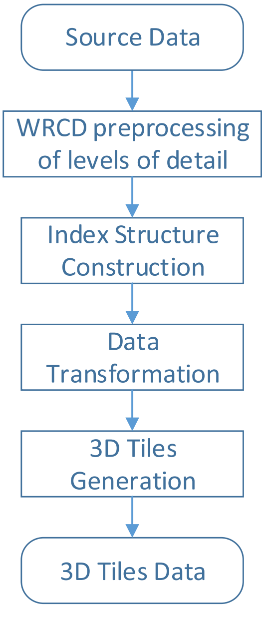

FIGURE 2 WRCD visualization process

schemes when data parsing. In the two data layering methods, multiple meshes are combined into one mesh to obtain grid data with lower precision. Therefore, when layering data, first we determine the number of grids of the layer data, and then we use the ratio of the number of high-precision grids to the number of low-precision grids to express the scaling size of the grid. Finally, the scaling size is distributed by the number of rows and columns, which indicates how many rows and columns are merged into a grid.

The solution proposed in this article is a visualization scheme starting from data parsing, so the data layering will be processed after data analysis. In order to show the overall structure of WRCD when the viewing angle is high, the first layer of the entire data will be presented as a whole in the visualization, so the number of grids in the first layer must match the number of browsers that can be loaded smoothly. The scaling size of the first layer can be determined based on the number of grids of the grid data. Then, according to the scaling size of the first layer, the scaling size of the second layer is adjusted and used to generate the second layer of data, which has more grids. In this way, the layering can be completed until the last layer is loaded according to the accuracy of the data itself. When parsing data, according to the scaling size, every few rows and every few columns are merged into one mesh. After loading all the meshes, a layer of data is generated, and then read according to the next scaling size until the last layer of data is generated.

In the process of data layering in precision, since multiple meshes are merged into one mesh, it is necessary to determine the attribute value size of the new mesh. In general, the commonly used attribute value substitution is to replace the attribute value of the new grid with the average of all grids or the attribute value of the center grid. However, the WRCD is often focused on the area of high echo intensity. In order to visually express which area has high echo data, the strategy adopted here is to use the maximum value of all mesh attribute values as the attribute value of the new mesh. Through the above steps, the data parsing and data layering steps can be completed and a series of data layers with different precision (Figure 3) is produced.

3.1.2 | Data segmentation

After the data parsing and layering step, each layer of original grid data is processed into data layers of different precision. According to the grid scaling of these data layers, only the first layer of data can be loaded in the browser at the same time, and the data from the second layer to the last layer can still not be fully loaded into the browser. In this case, the data needs to be displayed in blocks, which means that only a part of the data is displayed at the same time, and the remainder is not loaded.

Data segmentation is actually a scheme of cutting each layer of data into different data blocks as needed, ensuring that each block of data can be loaded separately in the browser, which makes the original data completely expressed by multiple blocks of data. The main purpose of data cutting here is to reduce the pressure on the browser in the visualization, so a simple data cutting method is enough. Since the WRCD is a type of grid data, each mesh is a square, so the complete grid structure can be preserved while cutting without destroying a single mesh. In the process of mesh cutting, the meshes of data can be divided into several parts by the number of rows and columns, and all the meshes in each part are a block of data.

After the data layering is complete, the data of the first layer can be directly loaded without cutting, so the cutting is required from the second data layer until the last data layer. Therefore, the method of cutting can still be judged by the number of grids. To complete the cutting of one layer of data, it is necessary to ensure that the

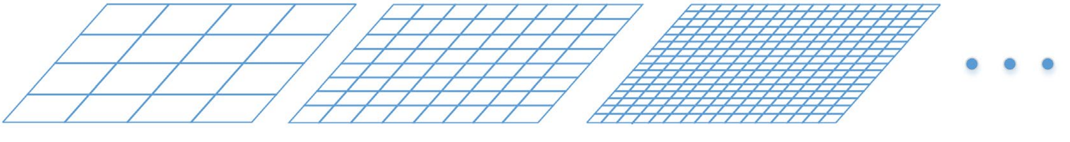

FIGURE 3 Data layering with different resolutions

number of grids of each data block after cutting can be smoothly loaded in the browser, and that all the data blocks of this layer can form the original data range. When cutting, the coverage of the data block in the previous layer is used as the basis of the cutting scheme of the current layer, and the cutting of the corresponding region in the current layer is completed by equally dividing the data block coverage according to the number of rows and columns (Figure 4). Then, after cutting each corresponding region in the current layer which matches the range of the data block in the previous layer in the same way, the current layer cutting is completed. Then, the next layer is cut according to the data blocks in the current layer. Continue the steps above until the last layer segmentation is complete. The data segmentation of one grid layer is then finished. This segmentation operation is performed on all grid layers to complete the data segmentation processing of the WRCD.

3.2 | Index structure construction of WRCD

After preprocessing of levels of detail, different data layers and data blocks have already given every grid layer of the WRCD levels of detail. The purpose of assigning the levels of detail to the WRCD is to reduce the pressure on the browser by only loading the required parts during visualization, which requires an index structure for all data layers and data blocks. Therefore, on the basis of the data layer and the data block, in order to convert the WRCD into 3D tiles data and visualize, the index structure of the data layers and the data blocks must be configured for each grid layer.

The data index structure construction is to connect different data layers and different data blocks, which means that each data block of any data layer can be quickly found. In the data segmentation, the current data layer is deliberately divided according to the previous data layer, in order to form the index structure. Starting from the first layer, the coverage of each data block of the upper data layer is consistent with the range of the corresponding data blocks of the lower data layer, which makes the WRCD with data layers and data blocks have a tree structure. As shown in Figure 5, since the data block of the upper layer often corresponds to multiple data blocks of the lower layer instead of the conventional two or four data blocks, the tree structure here is not a commonly used binary tree, quad tree, or the like, but only a common tree. When indexing, it is traversed according to this tree structure, and then finds the appropriate data block for visualization.

When data segmentation is processed, the first layer of data can be visualized without being cut, so the first data layer has only one data block, which is the root node of the entire tree. Starting from the second data layer, since the data block of the lower layer can be spliced out of the coverage of the upper data block, the data block

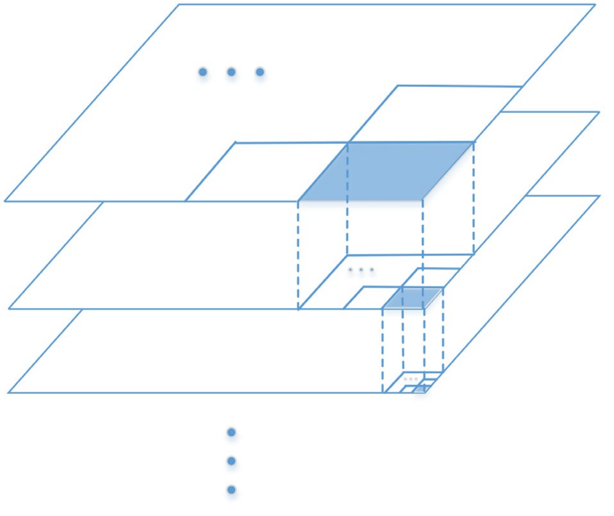

FIGURE 4 Data segmentation strategy

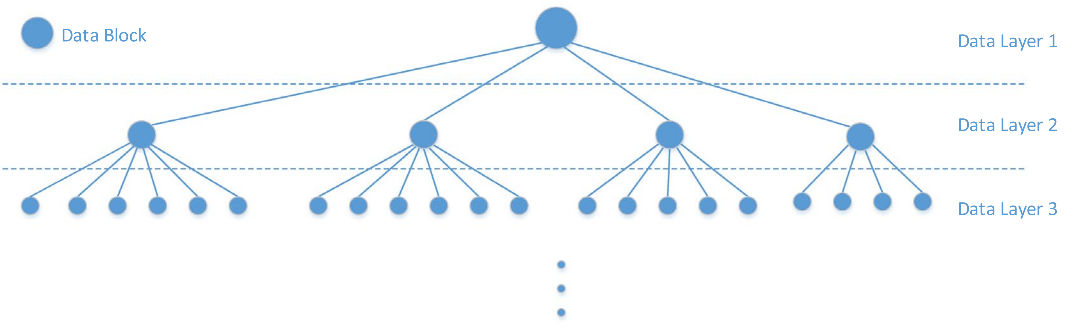

FIGURE 5 Tree structure of WRCD

of the lower layer within the range of the upper data block is the sub-block of the upper data block, which means that the data blocks in the lower data layer are the nodes of the tree. This structure continues until the last layer, meaning that the entire tree structure contains all data blocks of all data layers, so that each data block can be found when indexing. The depth of the tree is the number of data layers of the WRCD.

After the above processing, all data layers and data blocks of one grid layer are stored in a tree structure, and every grid layer of the WRCD will be stored in such a tree structure to complete the index structure construction of the WRCD.

3.3 | Data transformation for 3D tiles

Through the above two steps, the WRCD already has the level of detail and the tree structure, which makes web-based large-scale loading theoretically possible. However, in order to load the data into the browser, a suitable data format needs to be applied as a vector to the visualization process. 3D tiles data has obvious advantages in the field of web-based large-scale 3D model loading, which is why 3D tiles are selected for visualization of the WRCD.

Since 3D tiles data is a data format designed for 3D model web visualization, the data organization of 3D tiles is different from that of radar data, which makes it necessary to do some processing on the grid data to cater for 3D tiles. First of all, in terms of the coordinate system, the 3D model itself has a coordinate system based on the Cartesian coordinate system, which is different from the radar composing data in which the data is recorded as latitude and longitude. In terms of color rendering, a color table that returns texture properties based on the changes in radar composing data property values also needs to be defined. Finally, the index structure of the composing data needs to be stored in the index file of the 3D tiles to ensure that the levels of detail of the data can be displayed when visualizing. Therefore, this section introduces the three aspects of coordinate transformation, texture configuration, and index structure configuration.

3.3.1 | Coordinate transformation

Since the vertices of the model patches are expressed based on the Cartesian coordinate system in the field of visualization of the 3D model, it is necessary to convert the latitude and longitude coordinates of the WRCD into Cartesian coordinates. When the latitude and longitude coordinates are converted into Cartesian coordinates, the coordinate origin of the Cartesian coordinate system is at the center of the Earth sphere, while in 3D tiles data rendering, the model center needs to be set at a certain point on the surface. Therefore, after coordinate transformation, it is also necessary to move the coordinate origin of the Cartesian coordinate system to the center point of the surface data.

The transformation of the latitude and longitude coordinate system into a Cartesian coordinate system is the first step of the entire coordinate transformation. As shown in Figure 6, it converts the latitude, longitude, and elevation coordinates of a point O_n on the surface of the Earth into a Cartesian coordinate system with O_e as the origin. Its simplest conversion method can be obtained from Equation (1), in which R is the distance from the center of the sphere to the ground, and h is the elevation. However, in general, we usually use conversion methods from some GIS application library (such as CesiumJS).

$$\begin{cases} x = (R+h)\cos(\varphi)\cos(\lambda) \\ y = (R+h)\cos(\varphi)\sin(\lambda) \\ z = (R+h)\sin(\varphi) \end{cases} \tag{1}$$

After converting the latitude and longitude coordinates to Cartesian coordinates, the coordinate conversion work does not end. Since the center point of the model cannot be set to the center of the Earth when the 3D tiles model is visualized, and a point on the ground needs to be used as the coordinate origin, the Cartesian coordinate system is also required to be transformed.

As can be seen from Figure 7, when performing 3D tiles rendering, the ground point is taken as the origin, and the x-axis is the tangent of the longitude line passing through the point, and the true north direction is the positive direction. The y-axis is the tangent to the latitude line passing through the point, and the east direction is the positive direction. The z-axis points to the center of the Earth along the normal to the point. In order to display the model correctly, it is necessary to convert the Cartesian coordinate system from a coordinate system with point O_n as the origin to a coordinate system with point O_e as the origin. This step is actually to calculate the coordinate values of the points in the coordinate system with O_n as the origin in the coordinate system with O_e as the origin. The entire conversion process will be disassembled into multiple steps, as described in detail below.

The rotation of the coordinate axis is actually the rotation of the coordinate point. First, the angles θ and φ (Figure 8) at which the entire image needs to be rotated are calculated according to the positional relationship between the center point and the origin. Rotate all the points around the z-axis by θ°, and the direction of rotation is opposite to the direction of the right-hand spiral, so the rotation around the z-axis can be completed using Equation (2):

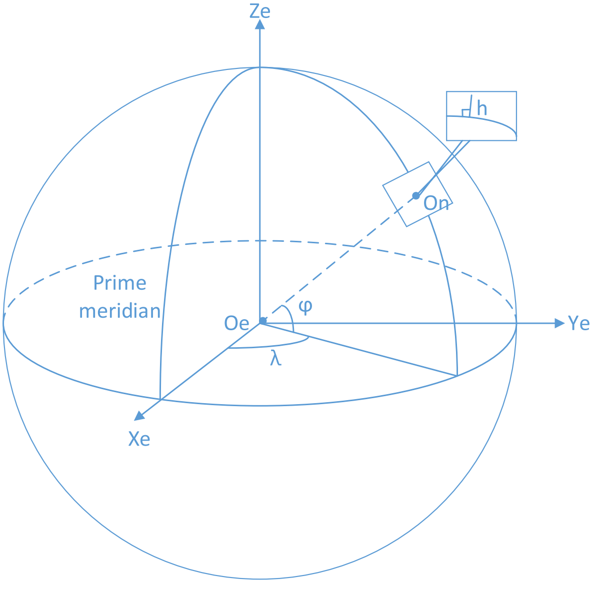

FIGURE 6 Schematic diagram of the coordinate system

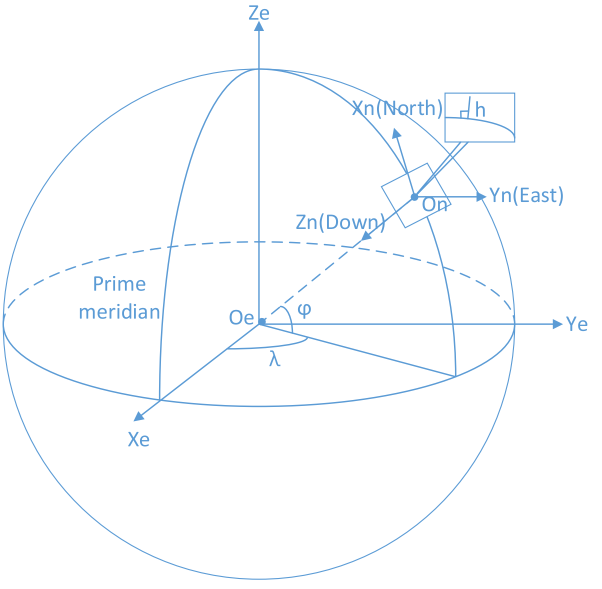

FIGURE 7 Cartesian coordinate transformation

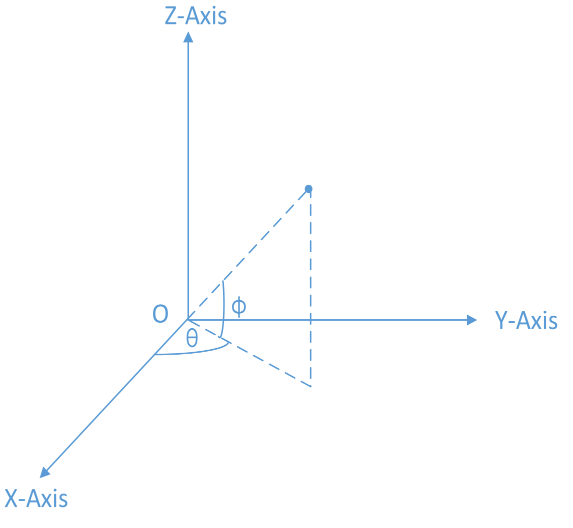

FIGURE 8 Coordinate rotation angle diagram

$$\begin{cases} x_1 = -x\cos(\vartheta) - y\sin(\vartheta) \\ y_1 = x\sin(\vartheta) - \cos(\vartheta) \\ z_1 = z \end{cases} \tag{2}$$

Then rotate the point around the y-axis ($90° - \varphi$). Similarly, the direction of rotation is opposite to the direction of the right-hand spiral, so you can use Equation (3) to complete the rotation around the y-axis:

$$\begin{cases} x_2 = -x_1 \cos\left(\frac{\pi}{2} - \varphi\right) + z_1 \sin\left(\frac{\pi}{2} - \varphi\right) \\ y_2 = y_1 \\ z_2 = -x_1 \sin\left(\frac{\pi}{2} - \varphi\right) - z_1 \cos\left(\frac{\pi}{2} - \varphi\right) \end{cases} \quad (3)$$

After completing the two-step rotation, the two coordinate systems are as shown in Figure 9, so the second coordinate transformation is performed according to Equation (4) to obtain the final result of the 3D image in the Cartesian coordinate system:

$$\begin{cases} X = -x_2 \\ Y = y_2 \\ Z = R - z_2 \end{cases} \quad (4)$$

Set the same elevation value for the grid points of the same grid layer. Different grid layers have different elevation values. Then, apply the above coordinate transformation to the vertices of all mesh points, and convert the geometry information of every mesh object into Cartesian coordinates with the origin of the 3D model as the origin. The entire WRCD completes the coordinate transformation, so that the geometric information of the data can be used by 3D tiles.

3.3.2 | Texture configuration

Texture information is an important part of 3D model data, which determines the basic information of the color, transparency, and other aspects of the 3D model surface. Unlike the fine model visualization field that 3D tiles are often used in, the models in the meteorological field do not need texture mapping, but express information

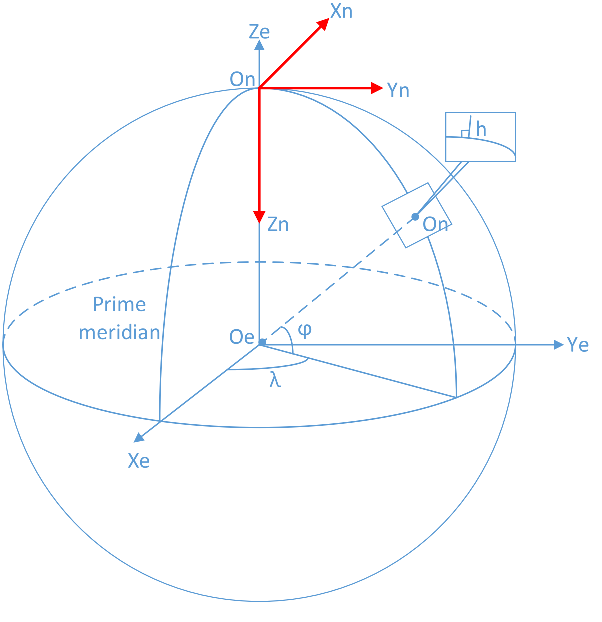

FIGURE 9 Result of rotation around the *z*-axis and the *y*-axis

through fixed colors. This article uses 3D WRCD, so the texture processing is easier. It only needs to set the color and transparency corresponding to the attribute value of every mesh, instead of generating a separate texture photo file. When texture mapping is performed, the attribute values of the radar composing data are classified, and each color texture is in one-to-one correspondence with a range of attribute values, so that the attribute values of the entire data can be expressed in different colors.

3.3.3 | Index structure configuration

In order to perform large-scale visualization of data, 3D tiles also use LOD (levels of detail) technology, which coincides with the previous settings of the levels of detail of WRCD. Therefore, this step is to record all the data layer and data block index structure of each grid layer into the index structure of 3D tiles, and add index conditions for them, which makes it possible that only some of the required data is loaded during visualization.

Since the index structure of 3D tiles is also a tree structure and supports one root with multiple nodes, which is similar to the WRCD index structure, it only needs to put the constructed weather radar index tree structure into 3D tiles to complete the index structure construction of 3D tiles. In terms of index conditions, it is necessary to consider the configuration of both the data layer and the data block, which determines how to visualize the data layers and data blocks.

The index condition of data layers is determined by the geometric error value, which is used to calculate the threshold for loading or not the data block when the view angle is changed. According to the LOD structure of the WRCD, when the line of sight is high, the data layer that needs to be visualized has a small resolution and a wide coverage, which means that the node of the tree it represents is high. Therefore, when setting the index condition of the data layer, the data block from the same data layer should have the same geometric error value. When the node represented by the data layer has a higher level in the tree, its geometric error value is larger.

The index condition of data blocks is determined by the bounding volume, which specifies the coverage of the data block. When performing 3D tiles visualization, it is determined whether the data block is loaded by calculating whether the data block coverage is within the view coverage. The data block coverage of the WRCD is recorded by the bounding volume of the region type (Figure 10), which defines a boundary geographic region parallel to the ground according to six values of west, north, east, south, minimum height, and maximum height. These properties can be calculated from the number of rows and columns and the cell size of the data block.

First, let 3D tiles inherit the tree structure of the WRCD, and then determine the index conditions by setting the geometric error and bounding volume properties of 3D tiles. After the above two steps, the index structure configuration of 3D tiles of one grid layer is complete. Then, the index structure is generated for each grid layer to prepare for the generation of 3D tiles index files.

3.4 | 3D tiles generation of WRCD

The final step in the large-scale WRCD visualization strategy is to generate 3D tiles data that can be visualized based on previously processed data and configuration parameters. The 3D tiles file consists of the index file tileset.json and the Batched 3D Model (b3dm) file (Anon, 2019b; James et al., 2000). The index file contains all the node indexes of the entire 3D model and the binary file describes the details of the model.

The grid data is first converted into a b3dm file, which is the concrete model. The general conversion method is to convert some existing models into 3D tiles data. Therefore, the grid data is first organized into obj data format. The mesh data of the coordinate-converted data block is converted into geometric information. The attribute value of each mesh is converted into texture information according to the color table defined earlier. Then, the

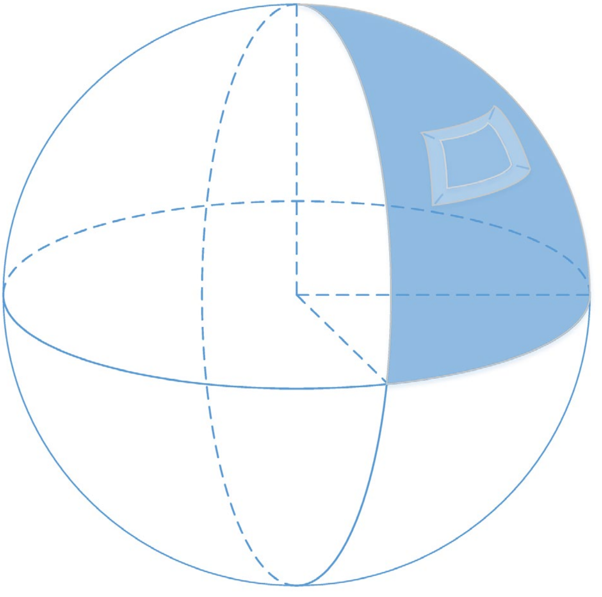

FIGURE 10 Region bounding volume

geometric information is used as the vertex information of obj data, and the texture information is defined as a fixed mtl file. These are combined to form an array in obj format. Next, according to the idea of converting obj data to 3D tiles, first convert it to gltf data, and then embed the gltf data into b3dm. Each data block of each data layer is converted into a b3dm file in this way, and then the details of the 3D tiles model can be generated.

Then, according to the constructed 3D tiles index structure, tileset.json of each grid layer is generated. The tree structure is completely imported into the index file, and each data layer and the index condition of each data block are recorded as attribute values in the index file. Finally, in the index file, the index condition and the specific geometric data are linked by the file path of b3dm, which allows the index file to guide the specific model visualization. In addition, in order to ensure that the model can be loaded into the correct position during visualization, the transform property of tileset.json needs to be set. Using the geographic coordinates of the WRCD center point as the origin, calculate a 4 × 4 transformation matrix, which is the transform attribute. Finally, the tileset.json file of all grid layers is aggregated into a single tileset.json to serve as an index file for the entire WRCD.

After the two steps above, the generation of the 3D tiles file is complete. By uploading 3D tiles data in the browser, web-based large-scale WRCD visualization can be realized, which also makes real-time visualization of the WRCD possible.

4 | EXPERIMENT AND RESULTS

This section implements the solution illustrated in Section 3 to validate the feasibility using the WRCD from weather radar sites across the whole of China. The experimental WRCD in China has a wide range of applications in real life, and contains important forms of weather radar data expression. To investigate the advantages of 3D tiles in large-scale WRCD visualization, an open source WebGIS JavaScript library, namely CesiumJS (Cesium, 2016), which supports 3D tiles, is used to handle the visualization to ensure that the benefits of these data can be fully exploited.

4.1 | Data processing

The data used in the experiment is the WRCD with 24 grid layers, which covers the whole of China. The boundary of the WRCD is from east longitude 73°, north latitude 12.2° to east longitude 135°, north latitude 54.2°, and the center point of the first mesh of each grid layer in the composing data is at east longitude 73.0°, north latitude 12.2°. This radar, composing data with 6,200 columns and 4,200 rows, has a cell size of 0.01°. It can be seen that each grid layer of this experimental data has 26,040,000 grids, which requires the strategy described above to complete such large-scale weather radar data visualization.

According to the steps of the visualization strategy, the first step is preprocessing of levels of detail. The data layering in height will generate 24 grid layers, which means that every grid is a single layer. Then, the data layering in precision will be processed. Every grid layer of experimental data will be divided into three layers in total. In the first data layer, the meshes of 20 rows and 20 columns can be merged into one mesh whose cell size is 0.2°. In the second data layer, the meshes of 10 rows and 10 columns can be merged into one mesh whose cell size is 0.1°, while the third data layer maintains the resolution of the original data. When the data segmentation is processed, the first data layer—which has only one data block—does not need to be cut, and the second data layer is cut from the middle of the grid data, which means it has four data blocks. The third data layer is cut according to the range of the data block in the second data layer, and the third layer area in each data block range is equally divided into 4 columns and 3 rows, which generate a total of 12 data blocks.

The next step is to construct an index structure for each grid layer, in this case to generate a tree structure with a depth of 3 using three data layers (Figure 11). The first data layer is the root of the tree, and the four data blocks of the second data layer are the four child nodes of the root node. Since there are 12 data blocks of the third data layer in the coverage of each data block in the second data layer, the nodes of the second layer of the tree have 12 child nodes, respectively. The third step is the data transformation for generating 3D tiles, which requires coordinate transformation and some settings. First, all the grids of the data blocks in the three data layers are coordinate converted so that they can be correctly identified and used by 3D tiles. Then, the attribute value of the WRCD ranges from 0 to 75. So, starting at 0, every five attribute values are assigned a new color and transparency value, which produces a color table. Finally, the index structure is set. The tree structure just constructed is used as the index structure of the 3D tiles. Set a geometry error value for each of the three data layers, and set the bounding volume properties according to the geometry of the data block.

In the last step, generate 3D tiles data for visualization based on the above processing results. First, the grid points in all the data blocks that have been converted into model coordinates are used as geometric information, and the texture information is generated according to the grid attribute values and the color table. Combine geometric information and texture information to generate a b3dm file. Then, convert the configured index structure into the index file (tileset.json) of each grid, and generate an index file (tileset.json) for the whole WRCD. Finally, the coordinates of the center point of the composing data are added to the index file as position information of the entire model on the 3D Earth, and 3D tiles data supporting the loading of the 3D model on the browser are generated.

4.2 | Design and development of meteorological data 3D visualization platform

In order to verify the 3D visualization results of the WRCD in 3D tiles format, a meteorological data 3D visualization platform (MDVP) is designed and developed based on Cesium (Figure 12), which is a JavaScript library based on WebGL with interfaces for various data sources for elevation, imagery, and vector layers (Schilling, Bolling, & Nagel, 2016). It has an operation interface specially designed for 3D tiles, which can complete the visualization task of 3D tiles.

The basic geographic data loading module can load the provincial boundary map, the 3D terrain, and some other basic background data, so that the user can obtain more geographic information when using it. The radar

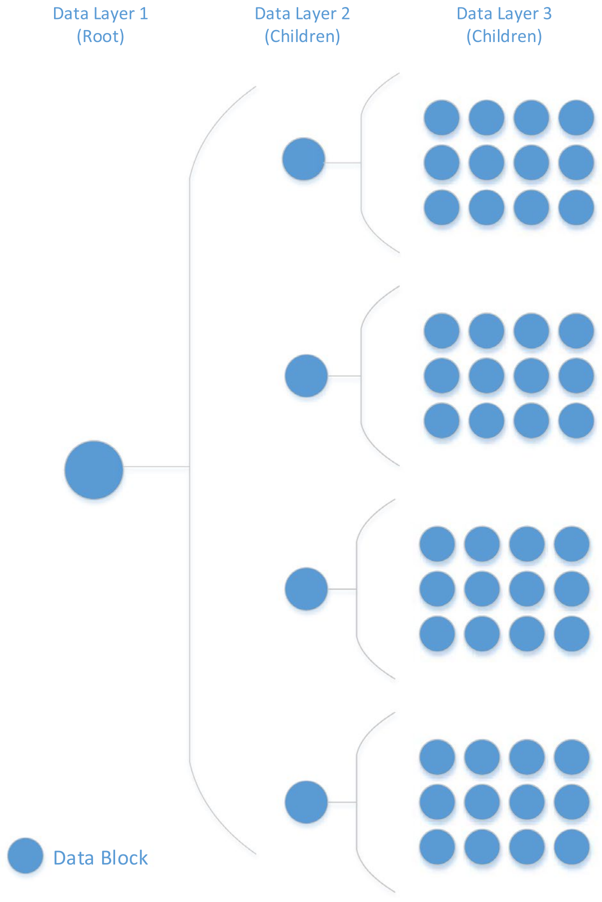

FIGURE 11 Index structure of data layers

data drawing module is mainly for the visualization of small-scale weather radar product data. The main function of the 3D model drawing module is to load some common model data, self-built models, and 3D tiles data. The experimental data in this article load the WRCD through this module to study the effect of 3D tiles drawing large-scale grid data.

After the MDVP is built, this platform can be used to complete the 3D tiles data visualization experiment. Since the 3D tiles model of the WRCD has a large amount of data, the entire data set needs to be released as a data service. After completing the 3D tiles data release, 3D tiles is loaded on the platform page according to the visual interface provided by Cesium to further verify its visualization effect.

4.3 | Discussion

4.3.1 | Visualization results

The 3D tiles model of the WRCD is visualized through the 3D model drawing module of the MDVP, which proves that 3D tiles have the ability to complete 3D visualization of large-scale WRCD. The result of the visualization of the 3D tiles file is shown in Figures 13 and 14.

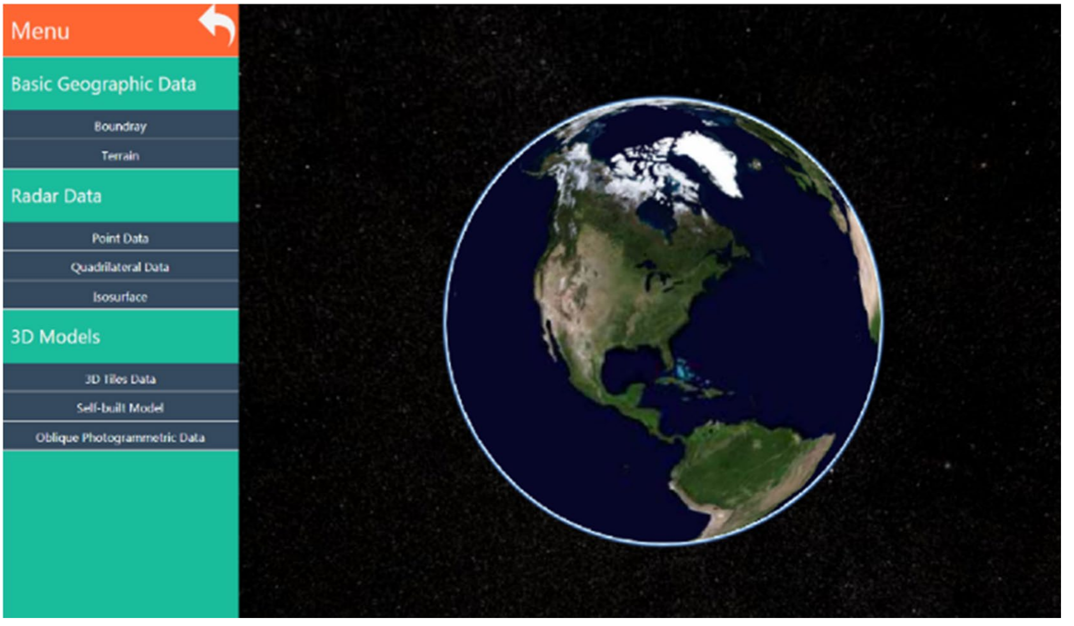

FIGURE 12 Radar data visualization platform

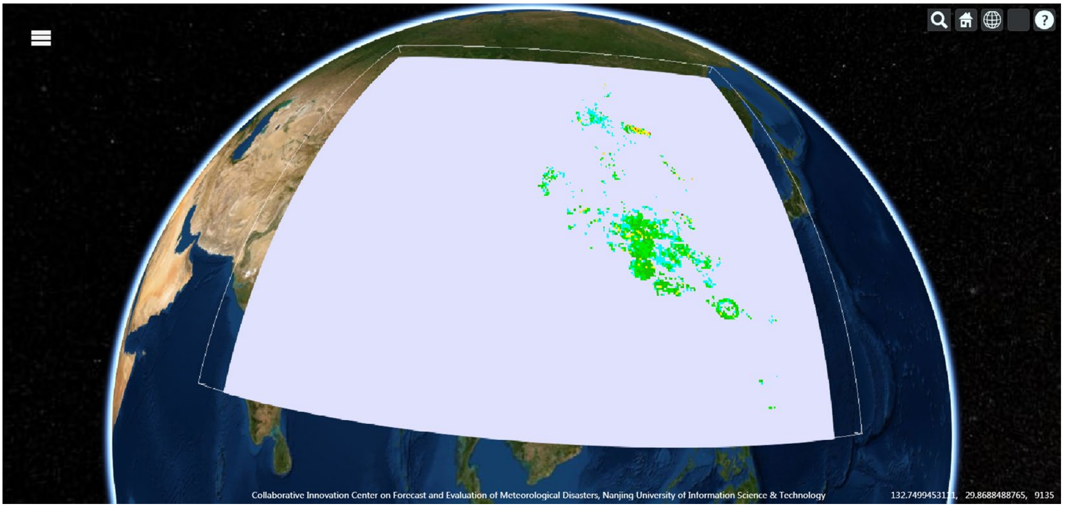

FIGURE 13 Render full data with bounding volume

According to the effect in the figure, all the grid layers are successfully loaded at the same time. Figure 13 shows the visualization of the complete WRCD. The white parts in the picture represent the no value grids. From the visualization result, it can be seen that the grid layers of the WRCD are completely drawn. According to the region in the figure, the boundary of the WRCD in 3D tiles format is consistent with the theoretical boundary of the original data, which can prove that the data are visualized in the correct place. The graph in Figure 14 is the result of hiding the no value part by some transparency processing, which makes the entire visualization effect more beautiful and clear. It shows the three-dimensional structure of the WRCD and multiple grid layers in the vertical direction when viewed from the side of the entire 3D tiles model. In order to make the structure of the

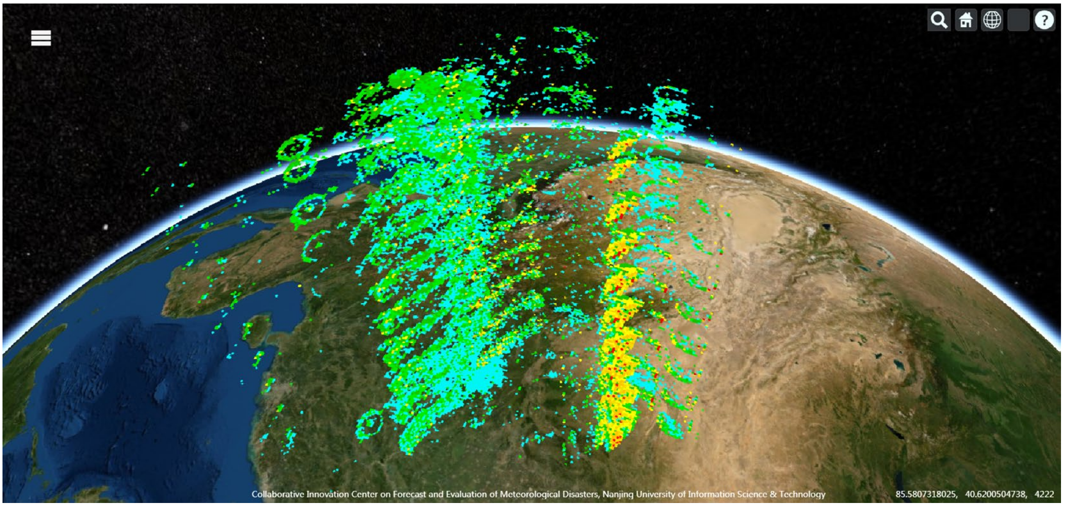

FIGURE 14 Rendering result with grid layers

WRCD in the vertical direction more obvious, we set the interval between adjacent data layers to 100 km during visualization. Through the graphs above, it can be seen that 3D tiles can not only load large-scale WRCD, but also express the structure of the data. After successfully drawing the WRCD, it can be seen that it is feasible to use the 3D tiles data format for the 3D visualization strategy of large-scale WRCD.

To verify that 3D tiles data can load large-scale data that cannot be loaded by ordinary methods, the composing data is loaded in a patched manner in the MDVP (Figure 15). As a result, the maximum number of patches that can be loaded when loading the patch is no more than 90,000, which cannot visualize a single grid layer, while the 3D tiles data can successfully load the full data with the LOD technology.

Therefore, it can be concluded that when using 3D tiles to process large-scale WRCD visualization, it can correctly load larger data.

4.3.2 | Results analysis

When the WRCD is converted into a single patch, it is necessary to follow the format of the grid data and record each grid. In 3D tiles data, each grid is stored as a patch, so there is no need to store all the grids. There are many no data grids in the WRCD. They can be omitted in 3D tiles and recorded as zero in a single patch. Therefore, in data conversion, the single patch solution uses 47,297 ms to convert a layer of the WRCD into 52.09 Mb data, while the 3D tiles solution uses 16,020 ms to generate a layer of 48.54 Mb data.

Data rendering speed is also an issue of concern when visualizing. Shorter loading speeds give users a better experience, while making dynamic visualization possible. In the following sections, the speed of radar composing data visualization using 3D tiles will be studied. Since each grid layer of the WRCD has the same structure, this allows each data layer to process the same number of grids when visualizing. Therefore, one of the grid layers is selected to be visualized, and the visualization speed of data layers with different precision is compared with the rendering speed of the direct loading patch scheme. Since the number of patches that can be loaded by the direct loading patch scheme is less than 90,000, only the first data layer meets the requirements, so only the difference of the first layer is compared.

In order to ensure the accuracy of the experiment, the drawing speed comparison experiment was carried out under the same network conditions. To further reduce the impact of network variation, the drawing speed will be

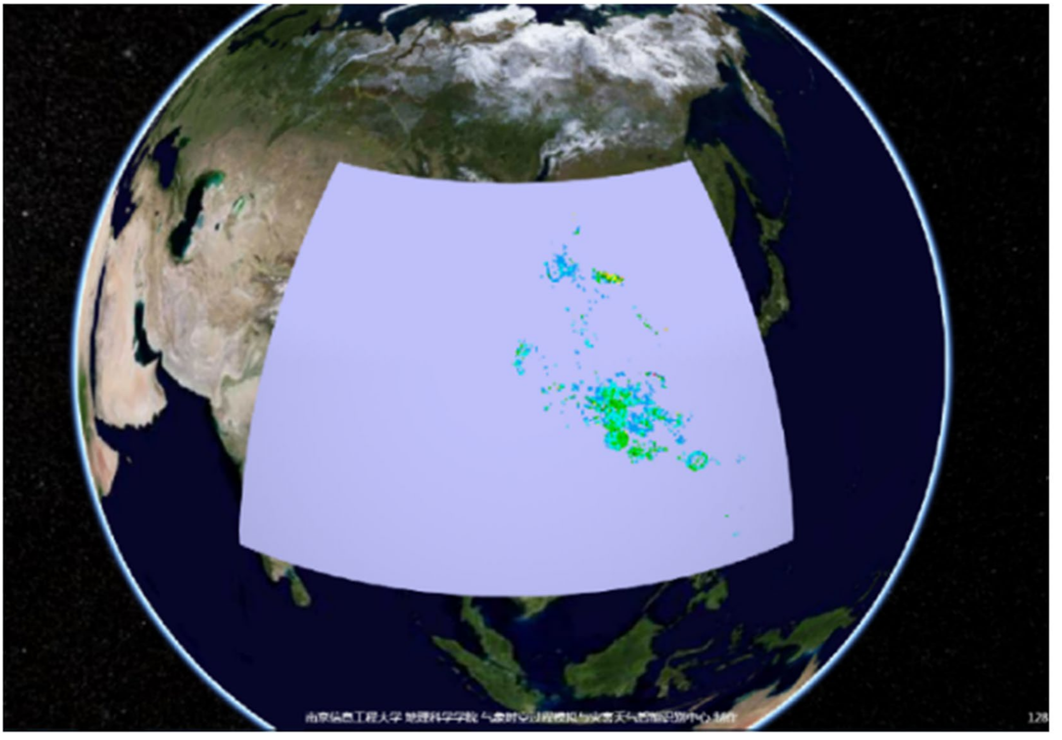

FIGURE 15 Rendering result of first data layer in a patched manner

TABLE 1 Loading speed table

		3D tiles average time (ms)			Patch rendering average time (ms)		
Layer number	Face number	Tileset reading	Rendering	Total	File reading	Rendering	Total
1	65,100	8.5	223	231.5	191.4	865.3	1,056.7
2	65,100		221	–	–	–	–
3	542,500		1,087.5	–	–	–	–

collected as much as possible in a short period of time. After eliminating the abnormal data caused by network problems, the average of multiple experiments will be obtained. The final result is shown in Table 1.

Since only a portion of the data blocks are loaded during the actual visualization process, the visualization speed of individual data blocks in each data layer is studied here. The face number given in the table is the number of patches for a single data block, and since the data index structure is merged into one index file, only one tileset.json is loaded. Since the patch drawing scheme cannot stably draw more than 100,000 patches, the first data layer is used here to make a comparison. As shown in Table 1, when loading 65,100 grids, 3D tiles take only a quarter of the time drawn by the patch. When the number of patches reaches more than 500,000 in the third data layer, the patch drawing scheme cannot solve the 3D visualization of these data at all. Therefore, when 3D tiles draw the WRCD, it can not only load more patches, but also load the WRCD faster. It makes real-time visualization of large-scale WRCD feasible.

5 | CONCLUSIONS

This article proposes a web-based real-time 3D visualization framework for large-scale weather radar data using 3D Tiles. An experiment using the weather radar data across the whole of China is implemented. Based on the experiments and analysis, it is obvious that 3D tiles data has unique advantages in drawing large-scale 3D WRCD. It can significantly improve the drawing speed of 3D data without affecting the final effect. This is actually due to its design philosophy: 3D tiles was designed for large-scale 3D model data. The pipeline from receiving a 3D Tile to rendering it with WebGL is streamlined to be fast and simple, and to minimize client-side processing. Although 3D tiles data have only been widely used for 3D modeling of oblique photogrammetry and point cloud reconstruction, its design can also serve 3D visualization in the meteorological field.

As the accuracy of weather radar data continues to increase, the amount of meteorological data is increasing. At the same time, with the development of the information industry, people have higher requirements for the visualization of meteorological data. Not only do they need to draw larger-scale meteorological data, but they also want to be able to visualize these meteorological data dynamically in a real-time manner. In this article, the advantages of 3D tiles in 3D model processing are applied to the 3D visualization process of the WRCD, which enables such large-scale grid data to be drawn completely at faster speed. This not only solves the web-based visualization problem of large-scale weather radar data, but also makes dynamic visualization of large-scale meteorological data possible.

Although 3D tiles data can be used to visualize large-scale weather radar data, there are certain limitations. There are many forms of weather radar data. However, this article only solves one of the visualization problems—large-scale visualization of grid data; other forms of large-scale data also require a viable visualization strategy. In terms of production process, due to the complex data structure of 3D tiles, weather radar data require a lot of processing steps when converting into 3D tiles, which makes the data conversion cost higher. Despite some small flaws, there is no doubt that 3D tiles data is still one of the best choices for large-scale 3D weather radar data. As long as it can fully produce its data production process, it can load a large amount of 3D meteorological data more efficiently.

In summary, applying 3D tiles to 3D visualization of weather radar data can take advantage of 3D tiles data and render weather data more efficiently. This is a viable solution to the problem of visualizing large-scale weather radar data. With the research and improvement of the 3D tiles data structure, it will solve more problems in the field of large-scale weather radar data visualization.

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers and the editors for their valuable comments and suggestions on earlier versions of the manuscript. This work is supported by a research project from the National Natural Science Foundation of China (No. 41871285) and the Academic Program Development of Jiangsu Higher Education Institutions (No. 164320H116).

ORCID

Xintao Liu https://orcid.org/0000-0002-7323-9878
Min Chen https://orcid.org/0000-0001-8922-8789

REFERENCES

Abimael, A., Scheer, S., & Sato, F. (2012). 3D visualization tool for meteorological radar data using WebGL. *Blucher Mechanical Engineering Proceedings*, *1*(1), 3237–3252.

Anon. (2019a). *3D tiles format specification*. Retrieved from https://github.com/AnalyticalGraphicsInc/3d-tiles/tree/master/specification#tile-format-specifications

Anon. (2019b). *Batched 3D model*. Retrieved from https://github.com/AnalyticalGraphicsInc/3d-tiles/blob/master/specification/TileFormats/Batched3DModel/README.md

Bell, V. A., & Moore, R. J. (1998). A grid-based distributed flood forecasting model for use with weather radar data: Part 1. Formulation. *Hydrology & Earth System Sciences Discussions*, *2*(2/3), 265–281.

Cesium. (2016). *Open source 3D mapping: Cesium JS*. Retrieved from https://cesiumjs.org

Chen, M., & Lin, H. (2018). Virtual geographic environments (VGEs): Originating from or beyond virtual reality (VR)? *International Journal of Digital Earth*, *11*(4), 329–333.

Chen, Y., Shooraj, E., Rajabifard, A., & Sabri, S. (2018). From IFC to 3D tiles: An integrated open-source solution for visualising BIMs on cesium. *ISPRS International Journal of Geo-Information*, *7*(10), 393.

Choi, H. W., Kang, S. M., Kim, K. J., Kim, D. Y., & Choung, Y. J. (2015). Development of the visualization prototype of radar rainfall data using the Unity 3D engine. *Journal of the Korean Association of Geographic Information Studies*, *18*(4), 131–144.

Cozzi, P. (2015). *Introducing 3D tiles*. Retrieved from https://cesium.com/blog/2015/08/10/introducing-3d-tiles/

Dobesch, H., Dumolard, P., & Dyras, I. (2013). *Spatial interpolation for climate data: The use of GIS in climatology and meteorology*. London, UK: ISTE.

Gan, L., Li, J., & Jing, N. (2017). Hybrid organization and visualization of the DSM combined with 3D building model. In *Proceedings of the Second International IEEE Conference on Image, Vision and Computing*, Chengdu, China (pp. 566–571). Piscataway, NJ: IEEE.

Goodchild, M. F. (2018). Reimagining the history of GIS. *Annals of GIS*, *24*(1), 1–8.

Haje, N. E., Jessel, J. P., Gaildrat, V., & Sanza, C. (2016). 3D cities rendering and visualisation: A web-based solution. In *Proceedings of the Eurographics Workshop on Urban Data Modelling & Visualisation*, Liège, Belgium (pp. 95–100). Geneva, Switzerland: Eurographics Association.

Herman, L., Russnák, J., & Řezník, T. (2017). Flood modelling and visualizations of floods through 3D open data. In J. Hřebíček, R. Denzer, G. Schimak, & T. Pitner (Eds.), *Environmental software systems, computer science for environmental protection: ISESS 2017* (IFIP Advances in Information and Communication Technology, Vol. *507*, pp. 139–149). Cham, Switzerland: Springer.

Hibbard, W. L. (1986). Computer-generated imagery for 4-D meteorological data. *Bulletin of the American Meteorological Society*, *67*(11), 1362–1369.

James, C. N., Brodzik, S. R., Edmon, H., Houze, R. A. Jr., & Yuter, S. E. (2000). Radar data processing and visualization over complex terrain. *Weather & Forecasting*, *15*(3), 327–338.

Kulawiak, M. (2016). Web-based 3D processing and dissemination of multibeam sonar data. *Hydroacoustics*, *19*, 259–264.

Kulawiak, M., & Kulawiak, M. (2017). Application of Web-GIS for dissemination and 3D visualization of large-volume LiDAR data. In I. Ivan, A. Singleton, J. Horák, & T. Inspektor (Eds.), *The rise of big spatial data* (Lecture Notes in Geoinformation and Cartography, pp. 1–12). Cham, Switzerland: Springer.

Liang, C. G., & Huang, Y. X. (2007). Weather radar data 3D visualization based on particle system. *Computer Knowledge & Technology*, *2*, 518–527.

Liu, Z., Shi, Z., Jiang, M., Zhang, J., Chen, L., Zhang, T., & Liu, G. (2017). Using MC algorithm to implement 3D image reconstruction for Yunnan weather radar data. *Journal of Computer & Communications*, *5*(5), 50–61.

Lu, M., Chen, M., Wang, X., Min, J., & Liu, A. (2017). A spatial lattice model applied for meteorological visualization and analysis. *ISPRS International Journal of Geo-Information*, *6*(3), 77.

Lu, M., Chen, M., Wang, X., Yu, M., Jiang, Y., & Yang, C. (2018). 3D modelling strategy for weather radar data analysis. *Environmental Earth Sciences*, *77*(24), 804.

Miller, R. L. (1996). *U.S. Patent No. 5,583,972*. Washington, DC: U.S. Patent and Trademark Office.

Nocke, T., Sterzel, T., Böttinger, M., & Wrobel, M. (2010). Visualization of climate and climate change data: An overview. *Journal of Atmospheric & Oceanic Technology*, *27*(4), 667–679.

Schilling, A., Bolling, J., & Nagel, C. (2016). Using glTF for streaming CityGML 3D city models. In *Proceedings of the 21st International Conference on Web3D Technology*, Anaheim, CA (pp. 109–116). New York, NY: ACM.

Song, Z., & Li, J. Y. (2018). A dynamic tiles loading and scheduling strategy for massive oblique photogrammetry models. In *Proceedings of the Third International IEEE Conference on Image, Vision and Computing*, Chongqing, China (pp. 648–652). Piscataway, NJ: IEEE.

Tilford, K. A., Fox, N. I., & Collier, C. G. (2002). Application of weather radar data for urban hydrology. *Meteorological Applications*, *9*(1), 95–104.

Titov, A., Gordov, E., Okladnikov, I., & Shulgina, T. (2009). Web-system for processing and visualization of meteorological data for Siberian environment research. *International Journal of Digital Earth*, *2*(S1), 105–119.

Wilson, J. W. (1970). Integration of radar and raingage data for improved rainfall measurement. *Journal of Applied Meteorology*, *9*(3), 489–497.

Yu, X., Yao, X., Xiong, T. N., Zhou, X., Wu, H., & Deng, B. (2006). *Doppler weather radar principles and business applications*. Beijing, China: China Meteorological Press.

Zhang, Z., Liu, L., & Wang, H. (2010). Application of 3D visualization technology to display of Doppler radar networking products. *Meteorological Science & Technology*, *5*, 4.

How to cite this article: Lu M, Wang X, Liu X, et al. Web-based real-time visualization of large-scale weather radar data using 3D tiles. *Transactions in GIS*. 2021;25:25–43. https://doi.org/10.1111/tgis.12638