A Blocky and Layered Management Schema for Remote Sensing Data

BEIBEI YANG1, RUI WANG1, WEN ZHANG1, CHENHAN WU2, XUJIN WANG1, AND LINGKUI MENG1

1School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China
2Wuhan Digital Engineering Institute, Wuhan 430070, China

Corresponding author: Lingkui Meng (lkmeng@whu.edu.cn)

This work was supported by the National Key Research and Development Program of China under Grant 2017YFC0405806.

ABSTRACT In the era of rapid data expansion and computer technology development, discrete storage, multiband push and fuzzy query remote sensing data management methods are no longer suitable for the data analysis needs of users, including the needs for long time series, global regions and multidata fusion. After analyzing traditional data management techniques, this paper discusses the existing achievements and development trends of current technologies. This paper aims to solve the problem of data sharing difficulties and organizational inconsistency caused by the use of different formats for the same spatial object. Based on a discrete global grid, this paper studies the blocky division method and coding specification of Google S2 and then accomplishes the layered storage of remote sensing data in HBase. Finally, Kylin is used to build a cube model to discuss the information mining analysis changes in the new data management model. Experiments show that the blocky and layered management schema (BLMS) can realize the unified management of global remote sensing data with multisource, heterogeneous, multiscale, and long-term characteristics and provide accurate data services on demand.

INDEX TERMS Data management, Google S2, HBase, remote sensing data.

I. INTRODUCTION During the rapid growth of remote sensing images, various storage centers have accumulated large amounts of image data in different storage formats according to the database system’s stage, technology, economy, physical load and application subjects, making remote sensing data appear diverse, high-dimensional and complex [1]. The existing remote sensing data management process is as follows: First, the band sequential (BSQ), band-interleaved-by-line (BIL) and band-interleaved-by-pixel (BIP) formats represent different methods of storing image pixels in memory or on disk [2]. Then, the pixels of these records are divided into satellite orbital strip or scene forms and packaged into various image formats, such as HDF5, GeoTIFF, and JPEG [3]. Finally, different data management methods, such as filesystems, database systems, and hybrid systems of files and databases, are adopted to store and manage the data.

These management systems provide multiple solutions for multisource data management. However, different data storage centers or systems adopt different organizational strategies and storage models, which limit the synergistic and integrated development of multisource remote sensing data [4]. They also often provide a rough way to push data. This kind of roughness is mainly reflected in the full-band push and the low retrieval accuracy for the region of interest (ROI). In addition, the current situation of multisource heterogeneous management hinders the development of multidimensional analysis of remote sensing data. To meet the demand of unified, standardized and orderly management of global data and provide users with more convenient analysis services, increasingly more scholars have begun to pay attention to the application of the geographic grid in image data management. According to certain rules, the surface of the Earth is divided into grids. Each grid area has its own associated points. The grid of each scale has the upper and lower level covering the scope. The Earth grid obtained by multilevel division will be used as the carrier of spatial data to effectively describe the spatial location relationship of global remote sensing data.

A. RELATED WORK The idea of combining grid technology with spatial data was proposed by Goodchild in 1992 [5]. Subsequently, Dutton [6]...
proposed a global hierarchical coordinate grid system based on triangle segmentation in 1996 to form a discrete global grid system (DGGS). Sahr and White [7] and Kimerling et al. [8] further improved the grid types and modeling rules of DGGS. Subsequently, domestic and foreign researchers successively proposed a variety of global segmentation models, such as tetrahedron [9], hexagon [10], octahedron [11] and icosahedron [12] models, and these models have been widely applied in many fields based on polyhedral structures, such as vector data structures [13], point cloud data [14], spatial data indexes [15], [16], data archiving and distribution [17], and global dynamic data structures [18]. DGGS is used by Wolfe et al. [19] and Shelock et al. [20] to manage Landsat and MODIS data, which proves the value of the grid data organization scheme for multiple spatial observation data storage and analysis. Scholars have also explored a variety of multilevel gridding schemes for remote sensing images. Li et al. [21] proposed a multilevel grid division scheme that provided theoretical support for the integration and sharing of multisource remote sensing data. Cheng et al. [22], [23] proposed the geographical coordinates subdividing grid with one dimension integral coding on 2n-tree (GeoSOT). Subsequently, Lu et al. [24], Zhou et al. [25] and Shan et al. [26] carried out in-depth research from three aspects: coordinate transformation, seamless splicing and data compatibility. Li et al. [3] and Wang et al. [27] applied GeoSOT coding to remote sensing image data integration, which verified the feasibility and efficiency of Oracle and HBase, respectively. Huang et al. [28] designed a five-layer fifteen-level (FLFL) structure for satellite remote sensing data management and used it for the precise management of agricultural remote sensing data. Wang et al. [29], Hou [30] and Zuo et al. [31] successively explored the algorithm, image collection processing and image data organization model of FLFL. Huang et al. [32] showed that the GeoHash remote sensing data organization strategy can effectively improve the retrieval efficiency of remote sensing data. For data storage, with the wave of informatization and the rise of the Internet, the NoSQL distributed database management system (NDDBMS) is becoming increasingly popular due to its high scalability, high availability, large data capacity, and fast performance [33], [34]. NDDBMS has several storage methods, including columnar storage, document storage, and key-value storage [35]. Bigtable [36], HBase [37], CouchDB [38], MongoDB [39], Redis [40], and Memcached [41] are all examples of this type of NDDBMS. Agarwal and Rajan [42] and Boehm and Liu [43] evaluated the performance of the MongoDB database in spatial object storage. Das [44] and Liu [45] tried to use HBase to store large amounts of image data.

These models solve the localization problem of global spatial information and retrieval on the technical level but focus on spatial division and the consolidation of the spatial index. The research on grid application modes is far from sufficient. In addition, the problem of how to manage and share remote sensing data in different formats has not been solved.

B. RESEARCH CONTRIBUTIONS

To solve the above problems, this paper proposes the blocky and layered management schema (BLMS). In this schema, the image data are first divided into spatial ranges (blocking from the transverse dimension) according to different spatial resolutions, and then band layers (layering from the longitudinal dimension) are divided according to different spectral resolutions. When the image is divided into blocks, Google S2 [46], [47] is used as the dividing standard. When the image blocks are layered, a single band after stratification is stored in the HBase distributed database. Furthermore, the metadata of the image are built by Kylin to provide the multidimensional analysis function.

The main contributions of this paper are summarized below:

a) This paper designs a blocky and layered management schema (BLMS), which has a higher precision of the index rate and efficiency than the existing system.

b) The method we proposed supports remote sensing images to be stored in the form of pixel values by band, which resolves the limitation of the file format and provides new ideas for unified management and management of multisource remote sensing data.

c) According to the statistical analysis needs of users, the secondary index scheme designed in this paper can provide analysis services with multiple dimensions and different granularities.

C. ORGANIZATION

The rest of the paper is organized as follows: Section II introduces the related background of discrete global grid, HBase and secondary index. Section III describes the design scheme of BLMS in detail, including blocky division on the sphere and layered separation between bands. Section IV discusses the changes that this model brings to data analysis services, and Section V provides a summary and conclusions.

II. BACKGROUND

A. DISCRETE GLOBAL GRIDS

The predecessor of Google S2 is Google’s multidimensional spatial point indexing algorithm. Similar to GeoHash, the Google S2 algorithm converts 3D geographic latitude and longitude coordinates into encoded strings similar to URLs, thereby reducing the redundancy generated by traditional methods while ensuring accurate retrieval of target locations. Therefore, this algorithm is applied to the spatial search service on search engines such as Google Maps, MongoDB and Foursquare. In this paper, we use the discrete grid of Google S2 as the constraint specification because of its technical characteristics. First, Google S2 has a small span between different scales, which provides 30 gradient
intervals spanning the 4th power curve to meet the “pointarea-global” remote sensing service requirements from 0.7 cm2 to 85,000,000 km2. Second, the length of the grid code is constant and small. Google S2 encoding storage requires uint64, which saves 4 bytes of storage compared to GeoHash’s 12 bytes. These four bytes are especially valuable when reducing the storage space required by large data sets. Finally, Google S2 uses a Hilbert curve to fill the spatial grid, which makes it dimensionally stable and continuous.

The encoding principle of Google S2 is shown in Fig. 1. First, the Earth is projected onto the six faces of a circumscribed cube, and then each surface is slightly projected and corrected so that the cells on the sphere are approximately the same size. Then, each face is iterated in the latitudinal and longitudinal directions and divided into four subsquares to form a 30-level grid. Finally, each grid level is encoded by the Hilbert curve. In Google S2, the encoding of different grid levels is 64-bit. As shown in Fig. 1, the first 3 bits of the CellID represent the 6 faces when the cube is projected. When reading, starting from the last bit of the code, the first bit that is not 0 is used as the flag bit. The first digit to the fourth digit of the flag bit represents the one-dimensional index for the effective digit of the plane grid coordinate i, j, and the index mode refers to the Hilbert curve.

**FIGURE 1.** The encoding principle of Google S2.

**FIGURE 2.** Data storage structure in HBase.

### B. HBase AND SECONDARY INDEXES

The underlying HBase uses the HDFS file storage system, which is managed by Zookeeper and exhibits high reliability, high performance, and scalability. This system is suitable for PB-level image data storage and supports unstructured multisource data structures. This schema can be used as a blueprint to build the storage system for the image data. A simplified storage structure of an HBase storage unit is shown in Fig. 2. RowKey is the unique ID for each record. Each RowKey can correspond to multiple column families. Each column family can contain several columns, and the value of each column attribute has its own version, which is represented by a timestamp. In HBase, the logical layering of remote sensing data and the millisecond-level fast response
FIGURE 3. Schematic diagram of the method.

Based on RowKey can be realized by means of column families. Before downloading data, users need to perform a multiconditional query on massive data to find data of interest. Although HBase can store a large amount of remote sensing data, it supports only the millisecond response based on RowKey and does not support the secondary index of non-RowKey fields, so it cannot meet the user’s large-scale, multiconditional query requirements. Therefore, response time has become a major shortcoming of grid data management, and a robust retrieval mechanism is needed to support more fine-grained rapid retrieval. In general, the user’s general query is based on the index conditions that the system has designed. As the amount of data increases, optimization is achieved through the establishment of indexes, database partitions, and table partitioning. However, in the era of big data, partition optimization efficiency is not obvious, so online analytical processing (OLAP) technology is needed to provide users with ad hoc queries [48]. Ad hoc query and analysis complement the two computing modes of batch processing and stream computing in the era of big data and have outstanding advantages in terms of timeliness and flexibility. Therefore, to achieve the rapid response of massive image data and meet the complex and diverse business requirements of remote sensing data, we combined HBase’s general query and Kylin’s ad hoc query to calculate the data of the data warehouse in advance and provide a query interface [49].

In this paper, we build a Kylin multidimensional analysis model based on the HBase first-level index to provide a multiconditional refined query and analysis for the band-level pixel matrix. Apache Kylin, developed by eBay, provides complex queries and multidimensional analysis of data [50], [51]. Kylin is a distributed engine built on Hadoop that solves HBase’s inability to scale, process, and analyze very large amounts of data horizontally. The engine provides SQL query interfaces and multidimensional analysis in the Hadoop architecture. When dealing with terabytes and petabytes of data, Kylin can achieve a second-level response and support high data concurrency. The core concept behind Kylin’s design is “space-for-time”. By precomputing multiple dimensions, the results are stored in HBase to improve the query speed. On the one hand, Kylin can analyze data in multiple dimensions and different granularities. On the other hand, it can achieve accurate data deduplication and fast response. In the application of remote sensing data, Kylin constructs data cubes through precomputation to realize fast retrieval and multidimensional analysis of images.

III. METHODS

During the analysis of spatial objects, new images are obtained by splicing or cropping between the regions from the transverse dimension and performing matrix calculation between the bands from the longitudinal dimension. Therefore, the refined management of remote sensing data includes the study of the blocky division of spheres and layered separation of bands (Fig. 3). When block segmentation is carried out in the region, Google S2 is introduced as the grid management model for the spatial data, which is applied to the integration of multisource remote sensing data. A geospatial grid can be used to decompose the image data generated within a specified range or associated with high intensity according to the rules into several remote data sets (blocky images) stored in the same grid. Since Google S2 has 30 different levels, it can be used to match remote sensing images with different spatial resolutions according to certain mapping specifications. When the band is layered for storage, if we deviate from the data structure limitation of the image and re-examine the remote sensing data, all the remote sensing images can be decomposed into the “metadata + pixel matrix” structure. This phenomenon means that multisource heterogeneous remote sensing data have a unified and standardized expression. The “metadata + pixel matrix” data management idea has high general-purpose characteristics and can be extended to the organization of raster data with different resolutions, such as Landsat, Sentinel and MODIS. The existence of the column family facilitates the “metadata + pixel matrix” structure. As a column-oriented storage mechanism, HBase provides horizontal scaling. Therefore, the image blocks can be layered according to the band and stored in different columns of the column family in the form of pixel values.
In the BLMS, the remote sensing data use the Google S2 coordinate system as the reference coordinate system, which is the World Geodetic System (WGS84). Then, the remote sensing data managed in the experiment are a product subjected to preprocessing, such as radiation correction and band registration. In addition, this management model has several core attributes that distinguish it from other management patterns. Below, we provide a description of the core attributes:

d) Remote sensing data from different sources are mapped to the corresponding grid level of Google S2 according to the specification, and each level can be mapped at multiple spatial resolutions.

e) Remote sensing images are strictly limited to image blocks of uniform size. In this experiment, 512×512 pixel blocks are used. In principle, there is data redundancy between adjacent image blocks.

f) Image block coding, which is also a key value in the database, is designed according to the space, time and source attributes of the image block.

g) The image blocks are stored in the database in units of rows, the respective band layers of the image block are stored in pixel values in the columns of the column family, and the affine transformation parameters for data reconstruction are assigned.

A. BLOCKY DIVISION ON THE SPHERE

To integrate remote sensing images with different sources, resolutions and coverage areas, it is necessary to study the multiscale spatial division at the global scale. The Google S2 grid proposed in this paper needs to solve two main problems during the organization of global remote sensing data:

The first is that the geographical coverage of the Google S2 grid cannot be directly used as a standard for blocky division. A grid-based blocking method is required between the grid and the image block. The Google S2 grid is not strictly rectangular, which will result in some trouble. During the process of regularized and standardized segmentation of time-series remote sensing data, these parallelogram-shaped grids will increase the computational complexity during image mosaicking and spatial calculations.

The other is that it is difficult to obtain an optimal match for the Google S2 level and image resolution. Google S2 offers 30 levels of different scales. This condition guarantees the accuracy of the image block but results in an image management problem. The smaller the image block is, the higher the accuracy of image retrieval and the effective utilization of resources. However, more small image blocks increase the number of disks that address read and write operations, and the response speed slows when the image is rendered. On the other hand, if the image block is too large, it can quickly satisfy large-area and global-scale analysis services, but many of the read data are not in the demand range, resulting in low transmission efficiency and wasted network resources. When we look at satellite imagery, we need both a global overview to achieve macroscopic monitoring and microscopic map information to provide detailed identification. Therefore, we will also discuss the optimal grid levels for the different resolutions.

1) GRID-BASED BLOCKING METHOD

As explained above, the Google S2 geographic grid is not a regular parallelogram, and standard rectangular or square remote sensing images are generally used in the field of remote sensing. Thus, there is a certain level of natural confrontation. To this end, we consider decomposing the remote sensing image into “square blocks” with fixed pixels, which contain the corresponding geographic grid. Such an image block can ensure that the storage size remains consistent and can accomplish spatial retrieval by means of grid coding. First, we need to choose a suitable size for the “square block”.

An image block with the same length and width will maximize the uniform operation efficiency, so a “square block” was considered the best choice for this study. Then, as a result of the computer architecture, the numbers must be stored and calculated in binary form in the computer. Therefore, the length and width of the “square block” must also be 2 to the power n (2, 4, 8, 16, 32, 64 ...). Finally, we refer to these three factors when selecting the size of the image block.

In storage centers and systems, 256×256, 512×512, 1024×1024, and 2048×2048 are relatively popular sizes. For example, these sizes are used by World Wind [52] (512×512), Google Maps [53] (256×256), and Google Earth [1] (256×256, 512×512, 1024×1024, 2048×2048). Similarly, in deep learning studies, 512×512 pixels has become the first choice for tiles [54]-[57]. Moreover, in terms of storage and rendering performance, JT Sample et al. [58] used an integer power of two between 256 and 2048 pixels for the accessed tiles during wasted pixel experiments. The results showed that both 256 and 512 performed well, and 512×512 was the best choice when a large number of image blocks was created. Finally, human vision should be considered when choosing the appropriate image size to ensure that the user can correctly identify the information in the image.

For a 27-inch screen with a 2560×1440 resolution, we obtain 108.8 pixels per inch (PPI). The recommended distance between human eyes and a 27-inch monitor is approximately 60 cm. Therefore, at a vertical distance of 60 cm, the best reading range between 10 and 20 degrees on the plane is 10.4-21 cm. That is, the length and width of the pixels are suitable in the range of 457 px-947 px. In addition, Liu’s research [59] showed that the normal resolution of the human eye for a computer screen is 0.25 mm; that is, 4 pixels can be distinguished in a 1 mm unit. Assume that the scale accuracy L is the physical distance represented by the smallest scale on the graph, which is equivalent to the spatial resolution of a remote sensing image. M is the scale denominator, and the following function is true:

\[
\frac{1}{M} = \varepsilon \frac{L}{R} = \varepsilon 
\]  

(1)

The minimum range of the 512 × 512 image matrix is 0.25 mm × 512 = 12.8 cm, which is in accordance with the range given above of 10.4-21 cm. The optimal spatial ranges of the 256 × 256 and 1024 × 1024 pixel matrices are 6.4 cm and 25.6 cm, respectively, which exceed the optimal reading range.

In conclusion, considering the discussion given above, we will use 512 × 512 pixels for the image block in this study. After selecting the size of the image block, the mapping range of the image block and the Google S2 grid becomes an urgent problem to be solved. This problem must be solved to achieve spatial indexing by uniformly coding the image slice with the help of cellID of Google S2.

The center point of the Google S2 grid is used as a reference point for mapping. A total of 256 pixels were read from the center point in the four directions of up, down, left and right to obtain an outsourced image block with a fixed size of 512 × 512 pixels (Fig. 4). The geographic extent covered by the image block is larger than the maximum latitude and longitude range of the grid, which satisfies Equation 2:

\[ \Delta \lambda_{\text{slice}} > \Delta \lambda_{\text{grid}} \text{ && } \Delta \Phi_{\text{slice}} > \Delta \Phi_{\text{grid}} \]  

where \( \Delta \lambda_{\text{slice}} \) is the slice longitude, \( \Delta \lambda_{\text{grid}} \) is the grid longitude, \( \Delta \Phi_{\text{slice}} \) is the slice latitude, and \( \Delta \Phi_{\text{grid}} \) is the grid latitude.

In image segmentation, if there is an edge block smaller than 512 × 512 pixels, it should be made up before segmentation to ensure that all image blocks are equal in size.

This segmentation method breaks the scale concept of conventional segmentation and takes the resolution and grid as the data scale to facilitate the rapid mosaicking of adjacent data blocks. The square dicing of the outsourced grid results in data redundancy.

2) MAPPING RULES FOR GRIDS AND IMAGE BLOCKS

The relationship between the geographic extent covered by the image block (BlockCover) and the Google S2 grid (GridCover) is shown in Fig. 5, which includes BlockCover \( \subseteq \) GridCover, GridCover \( \cap \) BlockCover, and GridCover \( \subseteq \) BlockCover. In Fig. 5a, GridCover covers the BlockCover, which means that full coverage of the image data cannot be achieved, and some data are missing when the image is mosaicked. When GridCover and BlockCover overlap (Fig. 5b), a certain extent of image data coverage can be achieved, but the coverage situation is relatively complex, which increases the difficulty of data management. When BlockCover overlays GridCover (Fig. 5c), there will be a large number of overlapping sections, which can achieve full coverage of the image data, but there will be data redundancy.

In the previous section, we chose the image discretization idea of cropping 512 × 512 pixels with the outer square of the grid center as the reference (Fig. 5c). As seen from Fig. 5, the boundaries of adjacent image blocks are covered one or more times. Partial overlap is essential for in-depth analysis of image data, but redundancy will be discussed in the following section. Next, we need to determine the most appropriate grid level to reach the desired spatial resolution.

Because BlockCover and GridCover both include two dimensions of longitude (\( \lambda \)) and latitude (\( \Phi \)), they are compared separately in the discussion of coverage relationships.

The process of selecting the optimum grid level includes the following three steps:

Step 1: According to the spatial coverage of the grid, we can obtain the maximum coverage GridCover_{\text{max}} at each level. By traversing all grids (assume the number is k) under the ith level, the maximum (MAX[\( \lambda \)]) and minimum (MIN[\( \lambda \)]) longitude and the maximum (MAX[\( \Phi \)]) and minimum (MIN[\( \Phi \)]) longitude of each grid are obtained. Then, we calculate the longitude and latitude differences of the grids. Finally, the maximum value is selected as the reference for GridCover_{\text{max}}.

\[
\text{GridCover}_{\text{max}} = \begin{cases} 
\text{Grid}_\Delta \lambda_{\text{max}} = \text{Max} [\text{Max}[\lambda] - \text{Min}[\lambda]]_k \\
\text{Grid}_\Delta \varphi_{\text{max}} = \text{Max} [\text{Max}[\varphi] - \text{Min}[\varphi]]_k 
\end{cases}
\]  

Step 2: The BlockCover is then calculated based on the spatial resolution of the image block. Since the image block size is the same, its coverage can be obtained by multiplying the spatial resolution by 512 pixels.

\[ \text{BlockCover} = \begin{cases} 
\text{Block}_\Delta \lambda = \text{Res} \times 512 \\
\text{Block}_\Delta \varphi = \text{Res} \times 512 
\end{cases} \]

Step 3: we obtain the i_level when the longitude of the image block contains the longitude of the largest grid and the j_level of the latitude range of the image grid containing...
TABLE 1. The table of corresponding remote sensing image resolution and segmentation level.

| Level | Min S2 Area/km² | Avg S2 Area/km² | Max S2 Area/km² | Demo Res/m | BC/km² |
|-------|-----------------|-----------------|-----------------|------------|--------|
| 9     | 194,643         | 324,291         | 408,121         | LE7        | 60     | 973,440 |
| 10    | 48.661          | 81,073          | 102,030         | LT5        | 30     | 235,930 |
| 11    | 12.165          | 20,268          | 25,508          | LC8        | 15     | 58,982  |
| 12    | 3.041           | 5.067           | 6.377           | GF1        | 8      | 16,777  |
| 13    | 0.760           | 1.267           | 1.594           | GF2        | 4      | 4,195   |
| 14    | 0.190           | 0.317           | 0.399           | GF1        | 2      | 1,049   |
| 15    | 0.0475          | 0.0778          | 0.100           | GF2        | 1      | 0.262   |

Sat—Satellite, Res—Resolution, BC—BlockCover, Landsat 5—LT5, Landsat 7—LE7, Landsat 8—LC8.

According to the above process, the remote sensing data are regularly cut into individual image blocks. The most suitable grid levels matched by remote sensing images with different spatial resolutions are shown in Table 1.

Taking the GF1 satellite PMS sensor as an example, it contains 4 2-m multispectral bands and 1 8-m panchromatic band, covering a range of 44 km × 43 km. According to the rules, the optimum level for the 2-m panchromatic band is level 14. At level 14 segmentation, a GF1 scene image is segmented into 3923 blocks. However, the multispectral 8 m band is divided into 293 blocks by level 10 of Google S2.

3) DATA MOSAIC AND REDUNDANCY

To ensure data integrity, security and high availability, remote sensing image data, especially high-resolution image data, require cross-border coverage between image blocks for distributed storage and analysis. This condition will waste some border pixels. This overlap is a trade-off between the integrity of data analysis and data redundancy. Therefore, comparison and analysis of data redundancy is essential. During redundancy analysis, we use two evaluation indicators: the pixel ratio method and maximum coverage frequency.

We assume that the data set of the pixels in the whole image scene is \([P] = \{P_1, P_2, P_3 \ldots P_n\}\). After image dicing, if the \(i\)th pixel (\(P_i\)) exists in \(m\) image blocks, the coverage frequency of the pixel is \(N_i = m\). The maximum coverage frequency \(M\) in the whole image is:

\[
M = \max \{N_1, N_2 \ldots N_n\} \tag{7}
\]

The pixel ratio method is the ratio of the sum of the pixel size of the image block and the amount of source image data of the same data source. We assume that \(k\) is the total number of image blocks, \(r\) is the source remote sensing image, and \(PH\) and \(PW\) represent the total number of pixels in the length and width directions of the source image or image block. Then, the formula of pixel redundancy \((C)\) is derived as follows:

\[
C = \frac{\sum_{i=0}^{k} Q_i}{\sum_{i=0}^{k} P_{Wi} \times PH_i} = \frac{k \times 256 \times 256}{PW_r \times PH_r} \tag{8}
\]

Three remote sensing satellite data sets from GF1, Landsat and MODIS are taken as examples. The spatial resolutions of these data sets are 2 m, 8 m, 16 m, 30 m, 60 m and 500 m. According to the calculations of Equations 5 and 6, the data redundancy at different resolutions is shown in Table 2.

From Table 2, after the remote sensing data are segmented at the optimum grid level, the pixel ratio redundancy is relatively constant. The redundancy fluctuates approximately 2, and the maximum coverage redundancy is 4 or 5. These redundant data will be stored as backup data for the image data.

In addition, the segmentation of this boundary overlap facilitates the mosaicking of data. Since the images are unified under the same coordinate system, there is a certain redundancy in the boundaries of the images, which guarantees the quality of the mosaic of the images. Therefore, we can completely process the data from the ROI in parallel, create image mosaics, and finally obtain the final visualization result for the image.

B. LAYERED SEPARATION BETWEEN BANDS

Current remote sensing data management is usually a file-path mapping method. In fact, the remote sensing data generated by this method are files rather than data values. Remote sensing data are a raster data set consisting of one or more bands. Each band consists of a series of pixel matrices, each of which is a data value. Therefore, we consider storing these pixel data values directly in a distributed database. According to the “metadata + pixel matrix” structure shared by the image blocks, a large number of discrete pixel values can be organized and stored in an ordered and structured manner.

This section uses the HBase database as an example, combining the common characteristics of image blocks and the HBase column storage model to study the layered separation between bands. The main research content of this model includes the following two points: (1) Design image block
coding. When the HBase database is stored, the entire image block is stored in the same row. Therefore, it is necessary to design a unique encoding system (BlockID) for the image block to implement a fast index. Considering that HBase is based on RowKey to provide queries, the RowKey value of HBase mentioned below is designed to be the same as the BlockID. (2) Band-based layered storage model. The metadata and image matrix of the image block are stored separately in different column families and are sequentially stored in each column of the column family according to the metadata attribute and the image band. When stored, the values of the same band in the image block are stored together in numerical values.

1) REMOTE SENSING IMAGE BLOCK CODING

Considering the data retrieval and equalization allocation problem, BlockID and HBase’s RowKey values are consistent. Therefore, BlockID is designed to meet the three RowKey principles: unique, concise, and high-level hash.

According to the specified data storage design in HBase, the structure design of BlockID/RowKey (hereinafter all referred to as BlockID) is as shown in Fig. 6. BlockID contains two types of codes: grid code (GridID) and image metadata code (MetaID). Among them, GridID is the Google S2 grid code obtained by image block mapping, which is used to describe the geographical space where the data are located. The user can quickly retrieve the data from the ROI based on the GridID; MetaID is a coding combination of image metadata attributes, including the satellite platform (Sat_id), the sensor (Sen_id), the resolution (Res_id), the level of the image (Level_id), the shooting timestamp (Time_id), and the scene number (Scene_id).

The total length of the BlockID code is designed to be 24 bytes, expressed as “GridID+Sat_id+Sen_id+Res_id+Level_id+Time_id+Scene_id.” The GridID, Time_id, and Scene_id directly select the encoded Google S2 values or the values from the remote sensing image metadata, while the Sat_id, Sen_id, Res_id, and Level_id need to design a set of mapping rules. The detailed principles of BlockID coding include the following:

a) GridID: long type with a size of 8 bytes. The value is the GridID corresponding to the center point of the image block; the CellID of different levels is composed of 64 units.
b) Sat_id: byte type with a size of 1 byte. According to the satellite mapping specification, the satellite platform is mapped to a unique numerical code, and the user can use this numerical code to confirm the satellite platform of the source data.
c) Sen_id: byte type with a size of 1 byte. The value is based on the sensor mapping specification and is used to distinguish the internal sensor settings of the platform.
d) Res_id: byte type with a size of 1 byte. Values are based on the resolution map specification to mark the spatial resolution of the image. A sign of 0 means that the resolution is higher than 1 m, and a sign of 1 indicates that the resolution is lower than 1 m.
e) Level_id: byte type with a size of 1 byte. Satellite data and product data are distinguished according to the product category mapping specifications. A sign of 0 indicates the band after image preprocessing; a sign of 1 indicates the product data.
f) Time_id: long type with a size of 8 bytes. The code value is the number of seconds from the start of January 1, 1970 (midnight of UTC/GMT) to the image capturing time.
g) Scene_id: int type with a size of 4 bytes. The encoded value is the same as the image scene number in the source database.

Taking the geometrically corrected image data (level 2) from the GF1/2 satellite as an example, we list some of the coding mapping specification tables for the satellite, sensor, resolution, and image categories in Table 3.

| Sat_id | Sen_id | Res_id | Cat_id |
|-------|--------|--------|--------|
| Sat coding | Sensor coding | Res/m coding | Cat coding |
| GF1 | byte (1) | WFV | byte (1) | byte (1) |
| GF2 | byte (2) | MSS | byte (1) | byte (1) |

Sat=Satellite, Res=Resolution.
2) BAND-BASED LAYERED STORAGE METHOD

In the previous section, we explained that remote sensing images can be decomposed into the structure of the “metadata + pixel matrix”. If this structure is mapped to the storage unit structure of HBase, the remote sensing image unit can be decomposed into two parts: “metadata column family” (MetaCF) and “pixel matrix column family” (PixelCF). The BlockID of Section III.B.1 provides HBase’s RowKey as a unique identifier, and together with “MetaCF” and “PixelCF”, constitute a layered storage model of remote sensing data, as shown in Fig. 7. “MetaCF” records a variety of basic attribute information for remote sensing data, including image space description information, affine transformation parameters, and image restoration information. Because the different layer matrices in the remote sensing image matrix represent different spectral band information, the different spectral information is recorded in different column attributes in “PixelCF”.

According to the layered storage model, we design the table structure of the image block in HBase (Table 4), showing the block and layered refinement management idea proposed in this study. In the table, the value of RowKey is the RS_GeoID code in Section III.B.1. Various descriptions of the remote sensing images are recorded in each column of MetaCF, including satellite name, sensor name, spatial resolution, shooting time, band information, grid coding, and cloud coverage. PixelCF stores the pixel values in each column of the column cluster in units of band layers.

The hierarchical storage model helps preserve analytical product data for remote sensing images. During the information analysis of remote sensing data, the structure of the pixel matrix itself is not modified, but the internal values are adjusted to form a new pixel matrix. Although the pixel matrix has new numerical information and attribute meanings, it is used to display the same corresponding spatial information. Therefore, we do not have to restore the product data as a separate file but instead add the processed data directly to the data storage unit of the original image. For example, product data such as NDWI, NDVI, and NDBI are generated after calculations using different bands to generate a new band matrix, which can also be additionally recorded in PixelCF in the form of new column attributes.

The design of the table structure is in line with the setting of the number and attributes of the column family for big data management of remote sensing images. The spatial information attribute and the image information are stored in different column families, and the management of the remote sensing data information from different sources is managed by freely controlling the columns in the column family.

The HBase table structure design of the layered storage model is the implementation of the “metadata + pixel matrix” storage structure proposed in this paper. Through this implementation, the attributes of multiple dimensions of remote sensing images can be uniformly expressed, and the data structure of multisource and heterogeneous remote sensing image data can be realized through discrete storage of data bands.

3) SECONDARY INDEX IN KYLIN

Kylin’s data are built on Hadoop, and with the help of the big data computing framework, the data in the data source are calculated according to the dimensional model and saved in the form of multidimensional arrays. During the query, the RowKey value is found in Kylin’s Cube through the entered query conditions, and it is then submitted to HBase. Based on the returned RowKey values, the HBase database can quickly find the image data.

The principle of the “HBase+Kylin” indexing scheme is shown in Fig. 8. The main flow of the query is simple. The RowKey and metadata fields involved in the HBase table that require conditional filtering are stored in the form of the cube dimension values of Kylin. This process implements the association of HBase’s RowKey and Kylin multidimensional cubes. In data retrieval, the RowKey values that satisfy the filter condition are first obtained in Kylin’s precomputed cube through conditional queries, and these values are then returned to HBase. HBase queries the HBase table according to the returned RowKey value, obtains the data set of the specified RowKey, and returns the data set of the final query to the upper client.

IV. EXPERIMENTAL RESULTS

This experiment verifies the changes obtained by the new management method to the data service model from the two aspects of remote sensing data retrieval visualization and multidimensional data analysis. We selected GF1 and GF2 satellite data from the China Centre for Resources Satellite Data and Application (http://www.cresda.com/CN/) as...
TABLE 5. The table structure of the image block in HBase.

| Sat  | Sensor | Res | Data                          | Size   | Level |
|------|--------|-----|-------------------------------|--------|-------|
| GF1  | PMS    | 2   | 2013.4.28-2018.11.22          | 12,205,794 | 14    |
| PMS  | 8      | 16  | 34,717,376                   | 12     |
| WFV  | 1      | 11  | 8,913,226                    | 11     |
| GF2  | PMS    | 10  | 2014.8.21-2018.11.20         | 9,813,226 | 15    |
| MSS  | 4      | 13  | 9,813,226                    | 13     |

Sat=Satellite, Res=Resolution.

the experimental data, referred to as CCRSDA the follow-
ing. The experimental data statistics are shown in Table 5. Since the image data cannot be acquired in large quantities, we obtained the metadata only from 2013 to 2018, and the image matrix was a random numerical matrix generated by real metadata simulation without the real information of the image. In the experimental data preprocessing stage, we first segmented the remote sensing data set to generate image blocks in different regions of the world, uniformly coded these image blocks, and finally layered them in the HBase database according to the band layer. In the data analy-
sis, we also need to extract some of the attribute fields of HBase’s MetaCF and build cubes to provide multidimen-
sional analysis services. (The code and data used in this study are available at https://figshare.com/ under the identi-
ifier https://figshare.com/s/a382708677c8e1fa14cb. GF1 and GF2 image data cannot be made publicly available to protect research participant privacy and consent).

The data management structure of GF1/2 adopts the Geo-
TIFF specification. The organization of image data adopts the BSQ format, and the data are pushed in a compressed tar format file. Taking GF1 PMS data as an example, the data compression package includes 4 multispectral bands with a resolution of 8 m and a panchromatic band with a resolution of 2 m. The composition of the compressed file of the data product is shown in Fig. 9.

A. IMAGE VISUALIZATION AND RETRIEVAL ACCURACY EVALUATION

1) IMAGE RETRIEVAL VISUALIZATION

Data management is implemented on a granular spatial scale, so query function changes will occur in the new data man-
agement model. More accurate data coverage analysis can be obtained when indexing data, and the coverage by the original image of the area can be calculated. The data query and visualization are shown in Fig. 10. This management method directly provides users with an analysis of the remote sensing data coverage in the ROI, helping users to better find the data they truly need.

The above figure shows a detailed comparison of the retrieval of the Poyang Lake area before and after data management optimization under the same query conditions. As seen from the figure, the image retrieval method before optimization returns a substantial amount of noninteresting data. The blocky and layered management system can ana-
yze the coverage of each area slice while providing more refined search results.

Since each band in the database is a separate index attribute, the user can directly select the desired band layer to download. As shown in Fig. 11, the user can directly query and output multiple band combinations. The water
FIGURE 10. The data query and visualization.
body classification map and the vegetation classification map of the corresponding space can be directly combined with the original image, which greatly improves the flexibility of data application.

2) EVALUATION OF QUERY ACCURACY
In this paper, two kinds of precision evaluation indexes of recall (R) and precision (P) are used to evaluate the difference in the precision evaluation formula of retrieval results. \( I_{ROI} \) is an ROI, \( I \) is all regions indexed, and \( I_R \) and \( I_N \) are indexed regions of interest and noninterest regions, respectively. The R and P are derived as follows:

\[
R = \frac{I_R}{R} \tag{9}
\]

\[
P = \frac{I_R}{I} = \frac{I_R}{I_R + I_N} \tag{10}
\]

In the accuracy evaluation experiment, we selected 10 different regions as variables, the image time was limited to 2017, and the cloud coverage of the image was \( \leq 100\% \). The study area contains 10 geographical areas with different traits and different ranges, as shown in Fig. 12. The data source is a selection of 4 scales of data of 1 m, 2 m, 8 m and 16 m of GF1/2.

To ensure the parallelism of the experiment, we compared the query results of CCRSDA and BLMS. It is worth noting that under the experimental query conditions, some regions of interest cannot be completely covered by the image.

It can be seen from Fig. 13 that when CCRSDA and BLMS are retrieved, the recall rates of the two systems are the same. This shows that BLMS, similar to CCRSDA, can retrieve all the data of interest in the database. The precision ratio of the two systems shows a consistent growth trend. They increase as the area of interest increases and as the spatial resolution increases. Statistical analysis of the four data retrieval results shows that the average retrieval accuracy of CCRSDA is 42.23%, the highest is 57.18%, while the average accuracy of BLMS retrieval is 85.90%, and the lowest is 76.53%. The results show that CCRSDA’s data retrieval is similar to a fuzzy query, and the retrieval accuracy is limited by the push range of the data ontology. However, BLMS uses the multilevel characteristics of the grid to decompose the image into countless small matrices of standard structure, avoiding the spatial scale of data being too large, reducing the excessive query and push of valuable data, and effectively improving image query accuracy.
B. MULTIDIMENSIONAL DATA ANALYSIS AND EFFICIENCY VERIFICATION

1) MULTIDIMENSIONAL DATA ANALYSIS

Kylin’s data cubes allow users to view and analyze cube data from multiple angles and directions to acquire data information. We extracted five types of field data, including satellite, sensor, resolution, time, and grid id, from the MetaCF column family of the HBase database to build a data cube for multidimensional data analysis experiments. The five-dimensional tables in the cube are derived from the data from five fields. Each of these dimensions is designed with a smaller dimension division interval to support multidimensional analysis operations of roll-up, drill-down, slice, pivot, and segmentation of remote sensing metadata. Fig. 14 below shows the flow chart for building the cube with these five fields and performing statistical analysis.
According to this process, we conduct multidimensional analysis from three dimensions: space, time and image attributes.

The analysis of the spatial dimension is the statistical process of the geographic grid covered by the ROI. Fig. 15a is a coverage map obtained by counting the data blocks over the Poyang Lake range in all the inbound data. Fig. 15b shows the statistics for GF2 data in January 2017 in the database. Fig. 15b can be regarded as Fig. 15a’s roll-up operation in the region dimension and drill-down operation in the time and satellite source dimensions. In the query, the time required for the statistical analysis shown in Fig. 15a is 1.53 seconds on average, and Fig. 15b requires 23.43 seconds.

Statistical analysis from the time dimension can provide granularity results at yearly, monthly and daily scales, as shown in Fig. 16. The results are displayed on three-color scales to better show the timing distribution of the image data. In the figure, red, yellow and blue indicate high, medium and low coverage, respectively. We calculated all the image blocks in the database. The statistics of the image blocks in the database in 2013-2018 are shown in Fig. 16a, which takes 0.88 seconds. Fig. 16b represents 12 months from 2013-2018, and the image statistics take 1.27 seconds. Finally, we analyzed the image from January 2016, as shown in Fig. 16c, which took approximately 0.69 seconds.

Different dimensions can also interact to form more complex statistical analyses. In the statistical analysis of the image source dimension, we choose the three dimensions of satellite, sensor and cloud. Through cube operations, such as segmentation and slicing, the image coverage distribution of different image sources and different cloud coverage is analyzed. The simulation data set includes images from GF1’s PMS and WFV sensors and GF2’s PMS and MSS sensors for the four different sources. Fig. 17 shows the results of our statistical analysis of the cloud coverage of these four sources, which took 2.88 seconds. Fig. 17b shows the statistical result of the GF1 satellite with cloud coverage between 1% and 50%, which was obtained by segmenting and pivoting the cube through Fig. 17a, which took 1.51 seconds.

2) EVALUATION OF QUERY EFFICIENCY
We compared the query performance in the two modes of the Hive index and BLMS index. According to the design of the data model, SQL statements are used in the query. The test methods include a single-table query and multitable query. We tested 10 groups of SQL queries, of which the first 6 groups are single-table queries and the last 4 groups are multitable queries. To ensure the accuracy of the test, the average of three times is used as the final query result for the same query. The results are visualized in Fig. 18.

In the 10 sets of queries, the results of the two sets of experimental queries are consistent, and the query time is shown in the figure. BLMS’s query efficiency in both single-table queries and multitable queries shows greater advantages than Hive. Compared with the single-table query, the multitable query times of both the Hive and BLMS indexes show an upward trend. Among them, except for Q4 and Q5 (Limit statements), Hive takes more than 100 seconds to query. Hive’s average times for single-table and multitable queries are 207.30 seconds and 526.58 seconds respectively. However, the query time of the BLMS index is usually very short and stable. The BLMS single table query takes an average of 4.73 seconds. The average time for multitable queries is 2.44 seconds.

The results show that Hive full table scanning combined with filters to filter out the target data has low query
This reduces the time-consuming full-disk scan. In addition, BLMS uses Kylin secondary indexes to convert high-complexity aggregation operations and multitable join operations into queries on precomputed results. It provides direct access during query, which determines that BLMS has a good fast query and high concurrency ability.

V. DISCUSSION

In this study, the Google S2 global discrete grid was used to organize global regional remote sensing source data, and standardized blocky image data were managed using distributed databases and multidimensional analysis of new technologies. This management mode enables the observation of remote sensing data from different dimensions while maintaining stable data redundancy 2-3 times. Experiments show that remote sensing data can be used for standard grid processing of data and stored in the form of pixel values in a database for management. This change in management mode will improve the efficiency of data usage. In addition, the metadata and pixel matrix of the image block are stored separately in different column families, which facilitates the
B. Yang et al.: Blocky and Layered Management Schema for Remote Sensing Data

FIGURE 19. Service model of BLMS.

fine-grained statistical analysis of the data, and the feasibility of multidimensional analysis is verified by the cube in Kylin.

The remote sensing image will no longer be restricted by the file structure after blocky and layered management, and the unified data output interface can solve the problem of analyzing and converting different images. Next, we will further elaborate the unique contribution of this schema from three aspects of data retrieval analysis and mining, as shown in Fig. 19.

a) Data retrieval service: In traditional data management schemes, multiple bands of image data are combined and organized by satellite orbital strip or scene forms, which often contain a large amount of uninteresting data. In this schema, remote sensing data are cut into several small blocks of uniform size to achieve different dimensions and different granularity of data observations. For example, we can obtain the Google S2 grid over water based on the vector data and extract only the data blocks with the water target based on the grid coding and data filtering requirements to reduce invalid queries and redundant transmission. This on-demand approach overcomes the traditional fuzzy query and provides refined retrieval services.

b) Data analysis service: On the one hand, we do not need file parsing and can directly obtain the bands required for the image in the ImgCF. On the other hand, mathematical processing analysis is performed on the pixel layers of each column, and the newly obtained matrix data are newly generated and directly added to the storage unit of the original image. This approach reduces the need for repeated processing by multiple users and enables the sharing of data resources. For example, the NDVI data product obtained by the NIR and red bands of the GF image can be regarded as one, but the band information is stored in the image subband column family of the original image storage unit. At the same time, the NDVI descriptive information attribute is added to the MetaCF.

c) Data mining services: Traditional management methods or schemas can obtain only images in the database, and do not provide statistical analysis services for query results. In this model, the time-series accumulated data are stored in similar spaces in chronological order, which facilitates statistical data density. The secondary index we established can reorganize the linear correlation between data of different dimensions, provide in-depth mining services for remote sensing images, and realize data-driven business development.

VI. CONCLUSION

In this paper, we propose a multilevel spatial tiling method for global remote sensing data based on Google S2, which guarantees the global relevance and multiscale of global spatial information. Then, we designed global remote sensing data
encoding and layered storage methods, which enabled the orderly management of source data information in distributed databases and greatly simplified the user’s query and use of data resources. Finally, we extracted the metadata field and combined it with Kylin to build a cube model to explore the changes in the new data management model for information mining analysis. The benefits of this remote sensing data management model are mainly reflected in the following points:

a) The model eliminates the “isolated information island”. The framework reintegrates data based on the segmentation of the Google S2 grid unit, unifying the data format and making the remote sensing data isomorphic, global and multiscale. The model builds a storage architecture for global remote sensing data fusion, providing a unified data resource pool for users. This feature lays the technical foundation for the integration of traditional data storage centers.

b) The model optimizes data services for users. In the new service mode, the range is accurately searched according to the demand, the data are acquired on demand in band units, and the data information is mined in grid units. When data are organized based on a multilevel grid, it becomes possible to provide data according to the retrieval, calculation and application requirements of the data, which also reduces invalid queries and redundant transmissions.

c) The model builds an open ecosystem for remote sensing data. This condition means that the remote sensing data from each data storage center are connected to the storage architecture through a standard data interface, are organized in an orderly and standardized manner, and users can freely select data from the ecosystem. Remote sensing data are clustered and stored in time and space, which is convenient for statistical data density. Fragmented management and effective data analysis can help users efficiently select data of interest.

This paper has performed some research on the integrated management of massive remote sensing data, and it has designed and implemented a refined management model. To better meet the application needs, there are still some issues that need to be explored:

a) Vector and raster integration. This paper discusses and studies the grid organization of only raster data. How to organize spatial vector data and integrate vector and raster data based on a discrete global grid and distributed database will be the focus of future work.

b) Cloud coverage recalculation. This experiment is conducted only to verify the service mode of the data, so the cloud coverage attribute of the image block is the analog data. Cloud coverage density is calculated from a satellite orbital strip or a scene of the remote sensing data, which means that after the image is segmented, its cloud coverage will be meaningless. Therefore, the renormalized blocky data require additional cloud coverage calculations to achieve attribute synchronization.

The method of analyzing and reorganizing global remote sensing data based on spatial grids and bands provides a new idea for the interaction and sharing of remote sensing data. With the development of big data technology, remote sensing data will gradually integrate globally discrete grids suitable for large-scale spatial data organization and distributed database technologies suitable for PB-level big data storage. Innovations in traditional data storage devices and architectures are undoubtedly imperative.

ACKNOWLEDGMENT

(Beibei Yang and Rui Wang contributed equally to this work.) The authors would like to thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] X. F. Lu, C. Cheng, J. Gong, and L. Guan, “Review of data storage and management technologies for massive remote sensing data,” Sci. China Technol. Sci., vol. 54, no. 12, pp. 3220–3232, Dec. 2011.

[2] J. D. Althaussen, “What remote sensing system should be used to collect the data,” in Manual of Geospatial Science and Technology. New York, NY, USA: Taylor & Francis, 2002, pp. 276–297.

[3] S. Li, C. Cheng, B. Chen, and L. Meng, “Integration and management of massive remote-sensing data based on GeoSOT subdivision model,” J. Appl. Remote Sens., vol. 10, no. 3, p. 34003, 2016.

[4] L. Liu, “Research on some key technologies of mass data management based on data grid,” Ph.D. dissertation, Univ. Chin. Acad. Sci., Beijing, China, 2005.

[5] M. F. Goodchild, “Geographical grid models for environmental monitoring and analysis across the globe (panel session),” in Proc. GIS/LIS, Phoenix, AZ, USA, vol. 94, 1994.

[6] G. Dutton, “Improving locational specificity of map data a multi-resolution, metadata-driven approach and notation,” Geographical Inf. Syst., vol. 10, no. 3, pp. 253–268, Apr. 1996.

[7] K. Sahr and D. White, “Discrete global grid systems,” Comput. Sci. Statist., Interface Found. North America, Inc., Fairfax Station, VA, USA, Tech. Rep., 1998, p. 30.

[8] J. A. Kimerling, K. Sahr, D. White, and L. Song, “Comparing geometrical properties of global grids,” Cartography Geographic Inf. Sci., vol. 26, no. 4, pp. 271–288, Jan. 1999.

[9] H. Alborzi and H. Samet, “Augmenting SAND with a spherical data model,” Proc. 1st Int. Conf. Discrete Global Grids, Santa Barbara, CA, USA, Mar. 2000, pp. 26–28.

[10] X. Tong, J. Ben, Z. Qing, and Y. Zhang, “The hexagonal discrete global grid system appropriate for remote sensing spatial data,” in Proc. Geoinformatics Joint Conf. GIS Built Environ., Adv. Spatial Data Models Analyses, Oct. 2009, Art. no. 71460J.

[11] A. S. Szalay, J. Gray, G. Fekete, P. Z. Kunst, P. Kukol, and A. Thakar, “Indexing the sphere with the hierarchical triangular mesh,” 2007, arXiv:cs/0701164. [Online]. Available: https://arxiv.org/abs/cs/0701164

[12] M. Satoh, T. Matsuno, H. Tomita, H. Miura, T. Nasu, and S. Iga, “Nonhydrostatic icosahedral atmospheric model (NICAM) for global cloud resolving simulations,” J. Comput. Phys., vol. 227, no. 7, pp. 3486–3514, Mar. 2008.

[13] X. C. Tong, J. Ben, Y. Y. Liu, and Y. S. Zhang, “Modeling and expression of vector data in the hexagonal discrete global grid system,” Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci., vol. XL-4/W2, pp. 15–25, Oct. 2013.

[14] T. Kim, K.-S. Kim, J. Lee, A. Matono, and K.-J. Li, “Efficient encoding and decoding extended geocodes for massive point cloud data,” in Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp), Feb./Mar. 2019, pp. 1–8.

[15] J. Ben, X. C. Tong, and Y. S. Zhang, “A new index structure for global geospatial data,” in Proc. Geoinformatics Joint Conf. GIS Built Environ., Adv. Spatial Data Models Analyses, Int. Soc. Opt. Photon., 2008, Art. no. 714607.
[16] N. Lu, C. Cheng, A. Jin, and H. Ma, “An index and retrieval method of spatial data based on GeoSOT global discrete grid system,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.* (IGARSS), Jul. 2013, pp. 4519–4522.

[17] F. Z. Wang, N. Helian, S. Wu, Y. Guo, D. Y. Deng, L. Meng, W. Zhang, J. Crowcroft, J. Bacon, and M. A. Parker, “Eight times acceleration of geospatial data processing and distribution on the grids,” *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 5, pp. 1444–1453, May 2009.

[18] K. M. Sahr, *Discrete Global Grid Systems: A New Class of Geospatial Data Structures*, Eugene, Oregon: Univ. Oregon, 2005.

[19] R. E. Wolfe, D. P. Roy, and E. Vermote, “MODIS land data storage, gridding, and compositing methodology: Level 2 grid,” *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 4, pp. 1324–1338, Jul. 1998.

[20] M. J. Sherlock, G. Tripathi, and F. Samavati, “Landsat 8 data modeled as DGGS data cubes,” in *Proc. AGU Fall Meeting Abstracts*, 2016, p. N53B-N1886B.

[21] D. Li, Z. Shao, and X. Zhu, “Spatial information multi-grid and its typical application,” *Geomatics Inf. Sci. Wuhan Univ.*, vol. 29, no. 11, pp. 945–950, 2004.

[22] C. Cheng, F. Ren, and G. Pu, *Introduction to Subdivision and Organization of Spatial Information*, Beijing, China: Science Press, 2012.

[23] C. Q. Cheng and C. Fu, “Earth space reference grid and its application prospect,” *Geomatics World*, vol. 7, no. 3, pp. 1–8, 2014.

[24] X. Lv, Y. Liao, and C. Cheng, “Multi-source remote sensing data organization based on GeoSOT location identification,” *Acta Scientiarum Naturalium Universitatis Pekinensis*, vol. 50, no. 2, pp. 331–340, 2014.

[25] X. Zou, D. Tang, L. Hao, and Y. Song, “Application of earth partition grid theory in image processing,” *Sci. Surveying Mapping*, vol. 44, no. 1, p. 16, 2019.

[26] H. Shan, C. Cheng, and P. Chen, “A method of the storage architecture design of geospatial data based on GeoSOT,” *J. Geomatics Sci. Technol.*, vol. 35, no. 3, pp. 311–318, 2018.

[27] L. Wang, C. Cheng, S. Wu, F. Wu, and W. Teng, “Massive remote sensing image data management based on HBase and GeoSOT,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.* (IGARSS), Jul. 2015, pp. 4558–4561.

[28] Y. Huang, Z.-X. Chen, T. Yu, X.-Z. Huang, and X.-F. Gu, “Agricultural remote sensing big data: Management and applications,” *J. Integrative Agric.*, vol. 17, no. 9, pp. 1915–1931, Sep. 2018.

[29] D. Wang, F. Zheng, J. Lai, T. Yu, J. Li, and S. Guo, “A new parallel algorithm based on five-layer fifteen-level remote sensing data organization,” *Microcomput. Inf.*, vol. 28, no. 1, pp. 116–167, 2012.

[30] A. Hou, “The research on geometry processing system for remote sensing image based on five layers fifteen levels data organization structure,” Ph.D. dissertation, Henan Univ., Kaifeng, China, 2015.

[31] Z. Zuo, D. Shang, and Q. Ge, “Design of GIS system based on GeoSOT’s fifteen-level image organization model,” *J. Henan Normal Univ., Natural Sci.* vol. 46, no. 6, pp. 108–116, 2018.

[32] K. Huang, G. Li, and J. Wang, “Rapid retrieval strategy for massive remote sensing metadata based on GeoHash coding,” *Remote Sens. Lett.*, vol. 9, no. 11, pp. 1070–1078, Nov. 2018.

[33] Y. Ma, H. Wu, L. Wang, B. Huang, R. Ranjan, A. Zomaya, and W. Jie, “Remote sensing big data computing: Challenges and opportunities,” *Future Gener. Comput. Syst.*, vol. 51, pp. 47–60, Oct. 2015.

[34] P. Amirian, A. C. Winstanley, and A. Basiri, “NoSQL database: MongoDB vs cassandra,” in *Proc. Int. Conf. Comput. Sci. Softw. Eng.*, 2013, pp. 14–22.

[35] J. Han, E. Haihong, G. Le, and J. Du, “Survey on NoSQL database,” in *Proc. 6th Int. Conf. Per与vas. Comput. Appl.*, Oct. 2011, pp. 363–366.

[36] J. Jose, H. Subramoni, M. Luo, M. Zhang, J. Huang, M. Wasi-ur-Rahman, N. S. Islam, O. Uoyang, H. Wang, S. Sur, and D. K. Panda, “Mencached design on high performance RDMA capable interconnects,” in *Proc. Int. Conf. Parallel Process.*, Nov. 2011, pp. 743–752.
**RUI WANG** is currently pursuing the Ph.D. degree in remote sensing and information engineering with Wuhan University, Wuhan, China. He has participated in the construction of the Water Responsive Response Remote Sensing Intelligent Platform. His research interests include the use of big data and remote sensing technology for time-series reservoir water changes, remote sensing data management methods for computational analysis, and the application of deep learning techniques in water monitoring. He received the first prize of the Dayu Water Conservancy Science and Technology Award.

**CHENHAN WU** received the Ph.D. degree from Wuhan University, Wuhan, China, in 2007. He is currently the Deputy Director and a Senior Engineer of the Innovation Center of the System Overall Department, Wuhan Digital Engineering Research Institute. Two patents have been applied for and authorized by the first author, with many copyrights. He has published several research articles, such as EI. He presided over a number of key research projects of the general equipment department, the naval equipment department, the National Defense Science and Technology Bureau and the group company, and won two second prizes and one third prize for progress in national defense science and technology. There are two second prizes and one third prize for scientific and technological progress of the group. His main research directions are cloud computing, big data processing, information system overall design, and so on. He is a Party Member.

**WEN ZHANG** received the Ph.D. degree from Wuhan University, Wuhan, China, in 2009. She was a Visiting Scholar with the joint Centre of Cambridge-Cranfield for High Performance Computing, Cranfield University, Cranfield, U.K., from 2007 to 2008. She is currently a Lecturer with the School of Remote Sensing and Information Engineering, Wuhan University. She has authored more than ten peer-reviewed articles. Her research interests include network GIS, remote-sensing applications, and spatial data analysis.

**XUJIN WANG** received the bachelor’s degree from Northeastern University, Shenyang, China, in 2018. He is currently pursuing the master’s degree with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China. His research interests include the management of remote sensing big data, high-performance geography, and the application of deep learning in remote sensing.

**LINGKUI MENG** was a Visiting Professor with the Joint Centre of Cambridge-Cranfield for High Performance Computing, Cranfield University, Cranfield, U.K., from 2006 to 2007. He is currently a Professor with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China. His research interests include remote-sensing applications in hydrology, cloud computing, and big data analysis.