Lightweight Remote Sensing Image Storage via GIS-Cloud Architecture with Fractal-DTCWT Compression and Knowledge Graph Indexing

Hui Lv[1*], Hao Dong[2]
[1]Zhengzhou Technology and Business University Zhengzhou City, Henan Province, 451400, China
[2]China International Telecommunication Construction GroupDesign InstituteCo., Ltd, Beijing, 100176, China
E-mail: 15981895687@163.com
[*]Corresponding author

Keywords: GIS, cloud computing, remote sensing image data, lightweight storage, data indexing, data compression

Received: November 3, 2025

This paper proposes a lightweight and secure storage framework that integrates geographic information system (GIS) management, cloud computing elastic architecture, fractal dual tree complex wavelet transform (DTCWT) compression, and knowledge graph indexing to address the storage and retrieval challenges brought about by the surge of multi-source heterogeneous remote sensing image data. This method first removes redundancy through band correlation analysis at the GIS end, and uses DTCWT and fractal coding to achieve high fidelity compression; Subsequently, a distributed database was built in the cloud, and a unified semantic index was generated using a knowledge graph. Chaos sequence algorithm was introduced to encrypt the index. To verify performance, the experiment used 500 high-resolution satellite remote sensing images (in two sizes of 512 × 512 and 1024 × 1024) as the dataset. The results showed that the reconstructed image had a maximum information entropy of 9.94 (with an expected standard of 8), and the details were preserved intact; The compression ratio remains stable above 6.0 (expected standard 5.5), and there is no significant decrease in land use classification accuracy when the compression ratio is 6.0; The MSSIM index based on knowledge graph is higher than 0.947, indicating high retrieval efficiency; The pixel change rate of chaotic encryption can reach up to 99.4%, with strong security. This study provides a practical and feasible technical solution for efficient storage, secure sharing, and fast retrieval of massive remote sensing data

Povzetek: This paper proposes a lightweight remote sensing image storage framework that integrates GIS-cloud architecture, fractal-DTCWT compression and knowledge graph indexing. It achieves efficient storage and rapid retrieval of massive remote sensing images while ensuring a high compression ratio and data security.

1 Introduction

Against the backdrop of rapid development in remote sensing technology, remote sensing images represented by Gaofen-1 satellite data exhibit the characteristics of multi-source heterogeneity, which poses a severe challenge to traditional GIS storage methods. The Gaofen-1 satellite data not only includes different resolution bands such as panchromatic (2-meter) and multispectral (8-meter), but also involves images from different time periods and regions, and has diverse storage formats (such as GeoTIFF, HDF, etc.). Traditional GIS storage methods are typically designed for single data types, making them inadequate for organizing such complex multidimensional data [1]. For example, due to the different spatial resolutions and spectral characteristics of images in different bands, traditional indexing structures (such as R-trees or quadtrees) need to consider both the fine spatial features of high-resolution panchromatic data and the spectral information of multiple spectral bands when managing them uniformly, resulting in redundant or conflicting indexing levels. Meanwhile, images from different times and regions may overlap or scatter in spatial distribution, making it difficult for traditional indexes to dynamically adapt to these spatiotemporal changes, resulting in a bloated index structure and decreased efficiency. In addition, the diversity of data formats further exacerbates the complexity of indexing [2], as different formats have different storage structures and metadata description methods, making it difficult for traditional methods to establish a unified retrieval mechanism. This inefficient index structure results in poor query and similarity matching performance, failing to meet the management demands of massive remote sensing data. Therefore, it is especially important to explore an efficient and low-cost remote sensing image data storage method. Das et al. [3], in order to realize the lightweight processing of hyperspectral remote sensing images, the method decomposes the hyperspectral image into low-dimensional factors by introducing rank-aware orthogonal parallel factorization, while retaining the main features and spectral information of the image. On the basis of decomposition, the method may use quantization, coding and other strategies to further compress the factors to reduce the data storage, and introduce the concept of unmixing to extract useful spectral information from the decomposed factors to further reduce the redundancy of the data, thus completing the data lightweighting process. Although this method offers advantages for lightweighting remote sensing images, its versatility is limited because different remote sensing data have distinct characteristics and application scenarios, different remote sensing image data may have different characteristics and application scenarios, so it may be necessary to use different decomposition strategies and parameter settings to optimize the compression effect, which increases the complexity and uncertainty of the application of the method. Silva et al. [4] to achieve the compression of the data processing, the method firstly calculates the spatial derivatives of the image, and based on the derivative, captures the changes of the edges, textures and other features of the image, and then through the calculation of the spatial derivatives to identify and extract the main features of the image, and these features are then extracted from the image, and the main features of the image are then captured. These features are then encoded or quantized to reduce the amount of data, so as to achieve compression and complete the lightweight management of data; however, in the application of this method, if the input image is noisy or the resolution is low, the calculated spatial derivative may not be accurate enough, which affects the compression effect and the quality of reconstructed images, leading to a decline in the quality of lightweight management of data. Delgosha et al. [5] to achieve the lightweight management of images, the method uses a local weak convergence framework to capture the local structural characteristics of the map, and analyze the characteristics of these maps, the method can design a compression strategy adapted to this type of data, the method ensures that no information is lost during the compression process, to maintain the integrity of the data and the recoverability of the data, i.e., to achieve lossless compression to meet the demand for minimized storage; however, in the process of application of the method, for the structure of the image is more accurate, thus affecting the compression effect and the quality of reconstructed images, resulting in a decline in the quality of data lightweight management. Truffinet et al. [6] to ensure the minimization of data processing, through the EIM generated by the proxy model based on the main features of the data to approximate the original data, generate a vector base and a set of interpolation points with less information to approximate the original data, the compression and simplification of the data is achieved by combining the parallel nature of the algorithm to accelerate the computation and reduce the storage requirements by optimizing the memory usage; the performance of the method is highly dependent on the characteristics of the input data, if the input data is not redundant, then the compression effect will be affected, which will lead to poor compression or inaccurate data recovery. Traditional methods are difficult to establish a unified retrieval mechanism. The confusion of this index structure leads to low efficiency in data querying and similarity matching, which cannot meet the efficient management needs of massive remote sensing data. In recent years, the collaborative processing of massive spatial data using cloud computing elastic resources and GIS spatial analysis capabilities has become a research trend, such as integrating GIS and cloud services to achieve efficient management of distributed spatial data [7-8]. Therefore, it is particularly important to explore an efficient and low-cost method for storing remote sensing image data. Literature summary as shown in Table 1.

Table 1: Literature summary

Research Method	Core Technology	Testing Dataset/Scenario	Key Results	Limitations
Das et al. [3]	Rank-aware orthogonal parallel factor decomposition	Hyperspectral images	Effective decomposition and compression while preserving spectral information	Limited generality, complex parameter adjustment
Silva et al. [4]	Spatial derivative feature compression	Distributed temperature data images	Compression based on feature changes	Sensitive to noise and low-resolution images
Delgosha et al. [5]	Local weak convergence framework (lossless compression)	Sparse graph structure data	Suitable for lossless compression of specific graph structures	Sensitive to data structures, limited application scenarios
Truffinet et al. [6]	EIM surrogate model approximation	Homogenized cross-sectional data	Data simplification based on feature approximation	Strongly dependent on data redundancy

Traditional GIS Storage	R-tree/Quadtree indexing	Multi-format remote sensing data	Mature spatial indexing	Index redundancy, low efficiency in managing heterogeneous data
Our Method	GIS + cloud architecture, fractal-DTCWT compression, knowledge graph indexing, chaotic encryption	Multi-source heterogeneous optical/SAR images, typhoon emergency scenarios	High compression ratio (>6.0), high information entropy (>9.9), high MSSIM (>0.947), strong encryption (pixel change rate >93.8%)	Cross-industry adaptability needs to be deepened; edge real-time processing needs to be optimized

There are significant differences in the lightweight storage requirements for remote sensing image data in the fields of agricultural remote sensing and meteorological monitoring. Agricultural remote sensing requires high-frequency updates of crop growth parameters (such as leaf area index and soil moisture), and compression algorithms are required to reduce data volume while preserving subtle feature changes to ensure the accuracy of growth monitoring; Meteorological monitoring requires the processing of massive heterogeneous data (such as satellite cloud maps and radar data), with a greater emphasis on maintaining spatiotemporal continuity and the integrity of key meteorological features to support trend analysis and disaster warning. In response to these needs, this article proposes a lightweight storage solution that integrates Geographic Information System (GIS) and cloud computing: this article proposes to utilize the spatial data processing capabilities of GIS to achieve redundancy removal and precise compression. It relies on the distributed architecture of cloud computing to provide elastic storage and parallel computing, and uses advanced coding technology to balance compression rate and data fidelity, in order to achieve efficient management, secure storage, and low-cost processing of cross domain remote sensing data; In response to the high-frequency update requirements of agricultural remote sensing, GIS spatial analysis is used to accurately preserve the subtle changes in crop parameters to ensure the accuracy of growth monitoring. In the face of heterogeneous data in meteorological monitoring (such as satellite cloud maps and radar data), cloud computing distributed architecture is used to maintain spatiotemporal continuity and key feature integrity to support disaster warning analysis; By integrating the spatial redundancy removal of GIS with the elastic parallel processing of cloud computing, a dynamic balance between cross domain data compression rate and fidelity has been achieved, effectively solving the limitations of traditional methods in heterogeneous data adaptation, computational efficiency, and storage costs. This provides an integrated solution for lightweight storage and efficient analysis of multi-source remote sensing data.

Based on the above analysis, this article aims to address the following core research questions: 1) How to build an architecture that deeply integrates GIS and cloud computing to achieve full process lightweighting of remote sensing data from preprocessing to secure storage? 2) Can the compression method combining fractal coding and DTCWT effectively maintain the detailed information (based on information entropy>8) and structural features (based on MSSIM approaching 1) of the image while achieving a high compression ratio (target>5.5)? 3) Can the indexing and chaotic encryption mechanism based on knowledge graph achieve efficient retrieval and secure sharing in heterogeneous data environments? The expected outcome of this article is that, on the same dataset, the proposed method achieves significant improvements in key metrics such as compression rate, detail preservation (information entropy), index quality (MSSIM), and security strength (pixel change rate) compared to the existing typical methods listed in Table 1. The following text will discuss this goal.

2 Architecture of lightweight storage method for remote sensing image data based on GIS+cloud computing

In this paper, in order to realize the lightweight storage of remote sensing image data, we propose the lightweight storage method of remote sensing image data based on GIS+cloud computing, which combines the advantages of GIS and cloud computing, and carries out the compression, encoding and encrypted storage of remote sensing image data to realize the lightweight management of remote sensing image data, and the architecture of the lightweight storage method of remote sensing image data based on GIS+cloud computing is shown in Figure 1.

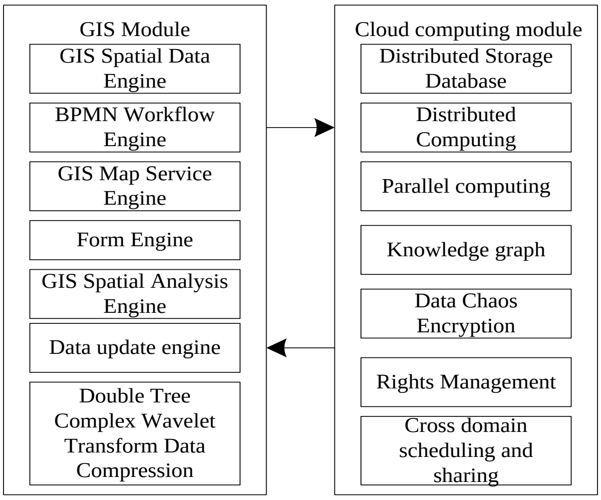


Figure 1: Architecture of lightweight storage method for remote sensing image data based on GIS+

This method consists of two parts: the GIS module, which fully leverages the core advantages of GIS in spatial data management and organization. By integrating multiple engine collaborative architectures such as GIS spatial data engine, BPMN workflow engine, GIS map service engine, form engine, GIS spatial analysis engine, permission engine, and data update engine, a complete remote sensing image preprocessing system is constructed. At the same time, this module deeply integrates algorithms such as image enhancement, filtering, wavelet transform, and various data compression techniques to achieve high ratio compression while ensuring visual quality. By dynamically associating remote sensing image data with metadata (such as shooting time, location, sensor information, etc.), structured and traceable data assets are formed, laying the foundation for data governance for lightweight storage. The second is the cloud computing module, which is based on an elastic distributed architecture and uses a parallel computing framework to optimize the storage and load balancing of compressed data in blocks. It also introduces pluggable encryption algorithms (such as AES-256 or homomorphic encryption) to achieve integrated security protection of "computation encryption storage" at the storage layer.

The theoretical innovation of this model that combines GIS management with cloud computing scalability is mainly reflected in the "layered processing" architecture: (1) data layering stores raw images, compressed data, and metadata in layers according to access frequency, breaking through the I/O bottleneck caused by traditional single storage; (2) Computational layering collaborates with edge node preprocessing and cloud deep analysis to solve the problem of high latency in traditional centralized computing; (3) The security layer adopts a differentiated encryption strategy of metadata plaintext and image data ciphertext, which ensures availability while avoiding the performance loss of traditional global encryption. This hierarchical system achieves a paradigm upgrade in storage efficiency and security through dynamic resource orchestration.

3 Lightweight storage of remote sensing image data

3.1 Lightweight processing of remote sensing image data based on GIS

3.1.1 The overall GIS structure

The main role of GIS in the lightweight storage of remote sensing image data is to realize the lightweight processing of the data, GIS has the characteristics of modular design, scalability, ease of use and robustness, which can realize the efficient management, analysis and display of remote sensing image data, and can make use of its powerful computational ability to better complete the remote sensing image data processing. Therefore, based on the lightweight processing requirements of remote sensing image data, the intelligent GIS structure is designed in the paper, as shown in Figure 2.

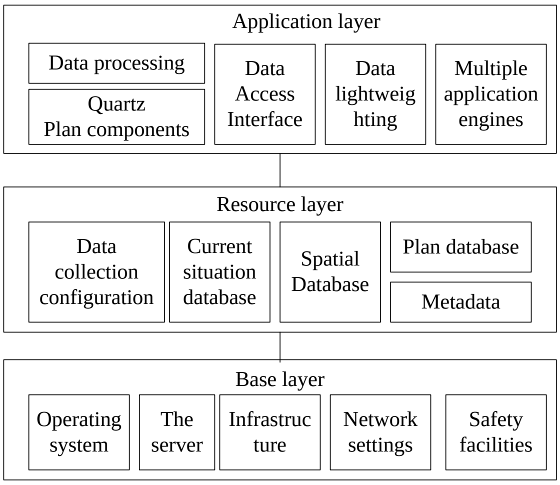


Figure 2: Intelligent GIS structure

Intelligent GIS adopts a 3-layer design, which is the base layer, resource layer and application layer. The data layer includes remote sensing image data, spatiotemporal trajectory data, geographic feature data, etc., supporting the storage and management of multi-source heterogeneous data. Different types of data have different processing methods due to their sources and characteristics. The resource layer is mainly responsible for the storage of different types of remote sensing image data. For the data acquired by different sensors, the storage strategy and management method will also be adjusted. The application layer provides support for remote sensing image data processing based on the powerful data processing capabilities of GIS, including various engine functions and integrated algorithms. With the support of these functions, remote sensing image data compression is achieved through dual tree complex wavelet transform (DTCWT) and fractal coding, achieving the goal of lightweighting. Taking Sentinel-2 and Radarsat-2 data as examples, due to Sentinel-2 being an optical satellite data with multi spectral bands and high spatial resolution, while Radarsat-2 is a synthetic aperture radar (SAR) data with all-weather imaging capability and little influence from atmospheric conditions such as clouds and mist, the two have significant differences in data format, radiation characteristics, geometric characteristics, and so on. Therefore, the preprocessing process of the GIS module requires differentiated design. For Sentinel-2 data, preprocessing may focus more on radiometric correction, atmospheric correction, geometric correction, etc., to eliminate sensor errors and atmospheric interference, and improve the accuracy and usability of the data; For Radarsat-2 data, preprocessing may focus more on speckle noise suppression, radiometric calibration, geometric correction (targeting the unique geometric characteristics of SAR data), etc., to improve data quality and provide a good data foundation for subsequent compression and application processing.

3.1.2 Correlation analysis of remotely sensed image data

In the process of remote sensing image data management, GIS needs to initially process the redundancy of the data to provide a basis for subsequent compression. Hyperspectral images contain two types of redundancy: intra-spectral and inter-spectral [9]. The primary task of compression is to reduce these redundancies, making effective use of image correlation crucial for improving compression performance. The correlation coefficient can directly reflect the linear correlation, and can be used as an effective index to measure the size of correlation; The interspectral correlation coefficient $\mu_{k,k+t}$ between k -band and $k+t$ -band is defined by the formula:

$$\mu_{k,k+t} = \frac{\sum_{x=1}^{N}\sum_{y=1}^{M}\left[\left(S(k,x,y)-S_k\right)\left(S(k+t,x,y)-S_{k+t}\right)\right]}{\sqrt{\sum_{x=1}^{N}\sum_{y=1}^{M}\left(S(k,x,y)-S_k\right)^2\sum_{i=1}^{N}\sum_{j=1}^{M}\left(S(k+t,x,y)-S_{k+t}\right)^2}} \quad (1)$$

Where: $S(k,i,j)$ and $S(k+t,i,j)$ denote the pixel value of the k -band and $k+t$ -band in the spatial location (x,y); S_k and S_{k+t} denote the average value of pixels in the corresponding band, respectively; N and M indicate rows and columns, respectively; is the number of rows in the separated bands.

The formula for the autocorrelation coefficients $\tilde{\mu}_{k,l}$ of the k -band is:

$$\tilde{\mu}_{k,l} = \frac{\sum_{x=1}^{N-l}\sum_{y=1}^{M}\left[\left(S(k,x,y)-S_k\right)\left(S(k,x+l,y)-S_k\right)\right]}{\sum_{x=1}^{N}\sum_{y=1}^{M}\left(S(k,x,y)-S_k\right)^2} \quad (2)$$

Where: l indicates the number of band rows separated.

The calculation results of the above $\mu_{k,k+t}$ and $\tilde{\mu}_{k,l}$ can be used for preliminary screening of redundant remote sensing image data, which can reduce the invalid and overlapping data in remote sensing image data to a certain extent and provide a reliable basis for the subsequent compression of remote sensing image data.

3.1.3 Remote sensing image data compression based on DTCWT and fractal coding

Following the initial redundancy reduction described above, this study employs DTCWT and fractal coding to compress the remote sensing image data, thereby achieving data lightweighting, the compression to achieve the lightweight of remote sensing image data [10]. Based on DTCWT and fractal coding remote sensing image data compression algorithm process contains two parts, respectively, DTCWT processing and fractal coding, of which fractal coding contains high frequency signal coding and low frequency signal coding two parts, the overall flow of the algorithm is shown in Figure 3.

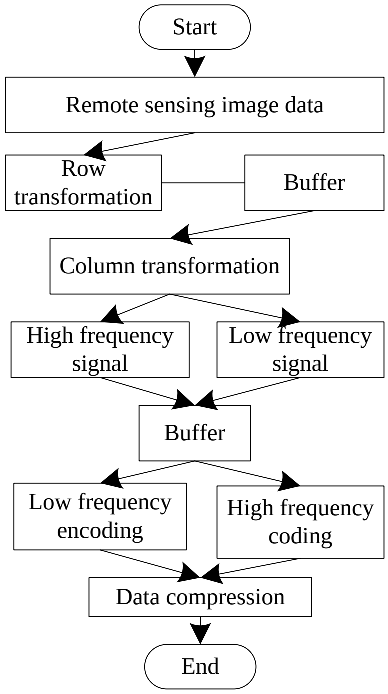


Figure 3: Compression process of remote sensing image data

This figure shows the complete process of decomposing the original image into multiple sub bands through DTCWT, followed by high-frequency sub bands entering the improved SPIHT encoding process, low-frequency sub bands entering the image entropy fractal encoding process, and finally merging the two streams to output the compressed result. The data flow and coupling relationships between modules are clearly indicated by arrows in the figure.

DTCWT is an improved wavelet transform method, which uses two parallel wavelet transform trees (i.e., real and imaginary trees) to process the signal, so that the information in the signal can be captured more comprehensively.In the process of remote sensing image data compression, in order to ensure the lossless compression of remote sensing data to avoid damaging the quality of remote sensing images [11], the paper performs the DTCWT transform of remote sensing images, which decomposes the image into a number of frequency and direction subbands, and then these subbands are quantized and coded using fractal coding to remove redundant information and reduce the data volume to complete the compression of remote sensing data [12], so as to realize the lightweight processing of the data.

(1) Remote sensing image data transformation based on DTCWT:

DTCWT (Double Tree Complex Wavelet Transform) has better directional selectivity, translation invariance, and reconstruction accuracy compared to traditional Discrete Wavelet Transform (DWT), and can effectively preserve detail information in remote sensing image compression [13]. The core mechanism is to use two parallel wavelet transform trees (real tree and imaginary tree) to process the real and imaginary parts of the signal separately: tree A generates real part coefficients, tree B generates imaginary part coefficients, and the two trees work together in the row and column directions of the image to generate low-frequency similar sub bands and multi-directional high-frequency sub bands in each level of decomposition. This structure significantly improves the directional resolution ability, enabling the transformation process to accurately capture features such as ground edges and textures in remote sensing images [14]. The dual tree design of DTCWT also ensures good translation invariance, avoids the pseudo-Gibbs effect of traditional DWT in image reconstruction, and further improves the usability of compressed data.

The formula for the real partially transformed wavelet coefficients $\phi_L^r(k)$ in the DTCWT decomposition and the scale factor $\varphi_L^r(k)$ are:

$$\phi_L^r(k) = 2^{\frac{L}{2}} \int_{-\infty}^{\infty} f(t) s^r(t)\left(2^L t - k\right) dt \quad (3)$$

$$\varphi_L^r(k) = 2^{\frac{L}{2}} \int_{-\infty}^{\infty} f(t) g^r(n)\left(2^L t - k\right) dt \quad (4)$$

Where: $f(t)$ is the input signal. L denotes the number of layers of decomposition; $s^r(t)$ is the scale function of the real tree; $g^r(n)$ is the number of wavelets in the real part of the tree.

The above equation is analogous to the wavelet coefficients $\phi_L^e(k)$ of the imaginary part of the tree and the scale factor $\varphi_L^e(k)$, thus obtaining the complete wavelet coefficients $\phi_L(k)$ and the scale factor $\varphi_L(k)$, the formula for both are:

$$\phi_L(k) = \phi_L^r(k) + i\phi_L^e(k) \quad (5)$$

$$\varphi_L(k) = \varphi_L^r(k) + i\varphi_L^e(k) \quad (6)$$

The DTCWT-based reconstruction formula for remotely sensed image data are:

$$\begin{cases} \phi_L(t) = 2^{\frac{L}{2}} \psi_i \sum_{n \in L} \left[\phi_L^r(n) s^r(t)\left(2^L t - k\right) + \phi_L^e(n) s^e(t)\left(2^L t - n\right) \right] \\ \varphi_L(t) = 2^{\frac{L}{2}} \psi_{L+1} \sum_{n \in L} \left[\varphi_L^e(n) s^e(t)\left(2^L t - k\right) + \varphi_L^e(n) s^e(t)\left(2^L t - n\right) \right] \end{cases} \quad (7)$$

Where: ψ_i and ψ_{L+1} denote the scale selection coefficients at different numbers of decomposition layers.

The formula for the reconstructed remote sensing image data signals $\tilde{f}(t)$ is:

$$\tilde{f}(t) = \varphi_L(t) + \sum_{L=1}^{N} \phi_L(t) \quad (8)$$

In two-dimensional remote sensing imagery, after the acquisition for $\tilde{f}(t)$, each sub-band of the signal is processed to obtain the low-frequency and high-frequency signals to realize the decomposition of the image data.

Algorithm 1: Pseudo code compression of remote sensing images based on DTCWT and fractal coding

Input: Original remote sensing image I, DTCWT decomposition layer L=4, value domain block size R=8 × 8, domain block size D=16 × 16, image entropy error threshold T=0.05.

Output: Compressed data stream C.

//Stage 1: DTCWT Transform and Subband Decomposition

Perform L-layer DTCWT decomposition on image I to obtain low-frequency sub bands LL_L and high-frequency sub bands {HL-k, LH-k, HH-k} in multiple directions, with k=1,2,..., L.

//Stage 2: Fractal Encoding

For each subband S in {LL_L, HL_k, LH_k, HH_k} do

text

If S is the low-frequency sub-band (LL_L) then

text

Using image entropy fractal encoding to process S:

text

Divide S into non overlapping R × R value range blocks.

text

Extract overlapping D × D domain block pools from S.

text

For each value domain block, search for the best matching block (minimum image entropy error) in its same class domain block pool. If the error is less than T, record the affine transformation parameters (fractal code).

text

Else//S is the high-frequency sub-band

text

Use the improved SPIHT algorithm to process S:

text

Initialize the ordered table and set the initial threshold T0=floor (log2 (max | coeff |)).

text

Embed the importance coefficient through steps such as importance transmission and refinement transmission.

text
end if
end for
//Stage Three: Entropy Encoding and Output
Perform arithmetic coding on all obtained fractal codes and SPIHT bitstreams.
Return compressed data stream C.

Set the image size to N × N pixels. The time complexity of DTCWT transformation is $O(N^2)$. In fractal coding, for each R×R block, it is necessary to search within $O((N/D)^2)$ domain blocks, with a time complexity of approximately $O((N/R)^2*(N/D)^2)$. Improve the SPIHT encoding complexity to O (N^2). The overall algorithm time complexity ranges from O (N^2) to O (N^4), but by limiting the search range and classification strategy, the actual computation can be controlled at the level of O (N^2 log N). The spatial complexity mainly consists of storage domain block pools and intermediate coefficients, which are approximately $O(N^2)$.

3.1.4 Signal coding of remotely sensed image data

In high-frequency signal encoding (such as building edges), the improved SPIHT algorithm is used as a lossless compression algorithm based on a hierarchical tree structure, combined with embedded zero tree wavelets and optimized code block encoding. By setting thresholds, using Mortan scanning order to determine the importance of wavelet coefficients, fine scanning, and processing important coefficients, the compression processing of high-frequency sub-band signals in multi-source remote sensing image data can be achieved. It can encode to different degrees according to different sub-band situations, improving the asymptotic transmission characteristics; In low-frequency signal encoding (such as green background), the image entropy method is based on the iterative function system and collage principle. It divides the original remote sensing image data into value domain blocks and definition domain blocks, uses image entropy as the classification standard to obtain the best matching block, sets a threshold, and completes fractal codes to complete the encoding process. By integrating the compressed high-frequency and low-frequency signals, the compressed remote sensing image data can be obtained, thereby achieving the goal of lightweight remote sensing image data. This scheme demonstrates good applicability by optimizing the encoding based on the physical characteristics of different frequency bands in the image. While ensuring that the quality of image reconstruction is not significantly affected, it significantly reduces the data volume.

(1) High-frequency signal coding

After obtaining $\tilde{f}(t)$ on the basis of the above subsection, the low-frequency signal and the high-frequency signal in the signal are encoded separately [15], in which the high-frequency signal is encoded by the improved SPIHT algorithm and the low-frequency signal is encoded by the image entropy. Improved SPIHT algorithm is a lossless compression algorithm based on hierarchical tree structure, improved SPIHT algorithm combines embedded zero-tree wavelet and optimized interception embedded code block coding to improve its own asymptotic transmission characteristics; wavelet coefficients are described by spatial direction tree [16], which realizes compression processing of multi-source remote sensing image data. The steps are described as follows:

Step 1: Set the threshold: this determines the importance of the wavelet coefficients, and at the same time introduces the ordered table, completes the storage of the coordinate positions of the coefficients, and adopts the initialization method to implement the processing of the threshold and the ordered table. If the wavelet coefficient coordinate set is denoted by $\Omega = (x, y)$, based on the coordinates to determine the initial threshold, define the high-frequency signal contains important and unimportant two kinds of coefficients table, the expression is as follows:

$$\begin{cases} \Gamma_1 = \{(x,y) | (x,y) \in B\} \\ \Gamma_2 = \{(x,y) A | (x,y) \in B\} \end{cases} \quad (9)$$

Where: Γ_1 is a table of insignificant coefficients. Γ_2 is a table of significant factors. A and B denote the two types respectively.

Step 2: The scanning is performed to Γ_1 to check all the wavelet coefficients it contains and to judge their importance, which is accomplished by the Mortan scanning sequence of the embedded zero-tree wavelet, whose judgment formula is:

$$Z_n(\Omega) = \begin{cases} 1, \max\left\{ \left| \begin{matrix} \vec{\phi}_j^1 \\ (x,y) \in \Omega \end{matrix} \right| \right\} \geq 2^n \\ 0, \text{else} \end{cases} \quad (10)$$

Where: $\left| \begin{matrix} \vec{\phi}_j^1 \\ (x,y) \in \Omega \end{matrix} \right|$ denotes the absolute value of the low-frequency wavelet coefficients; 2^n denotes the threshold value.

If the result of $Z_n(\Omega)$ is equal to 1, indicating that Ω is important relative to the threshold, and unimportant vice versa, and this completes the judgment of $\Omega = (x, y)$. On this basis, all tables of A and B of the two types contained in Γ_2 are finely scanned and processed to obtain the important coefficients in the scanning process, and also to obtain the modified values of the coefficients in the current plane.

Step 3: After scanning through the previous step, output the Min value and two kinds of scanning bit streams, i.e., sorted and fine; at the same time, also output the initialization information, which belongs to all the ordered tables; all the above outputs need to be transmitted to the decoder, and determine the information collection of the next scanning. Through the above steps, the generation of high-frequency subband coefficient matrices can be completed, and the improved SPIHT algorithm can be encoded to different degrees according to different subbands [17], realizing the compression of high-frequency subband signals of remotely sensed impact data.

In this study, the initial threshold T0 is determined based on the maximum absolute value of the subband coefficients in the improved SPIHT algorithm, as shown in formula (8). The scanning order adopts the standard Mortan (Z-shaped) sequence.

(2) Low frequency signal coding

In fractal coding, the range block size is set to 8 × 8 pixels, the domain block size is set to 16 × 16 pixels, and sliding extraction is performed from the image (with a step size of 2). During the matching process, allowed affine transformations include isometric transformations and contrast scaling. The image entropy error threshold ε is set to 0.05.

Low-frequency signal coding is accomplished using the image entropy method, which is based on the iterative function system and the collage principle, in which the iterative function is represented by the set of contraction affine transformations in a certain metric space as $F=\left\{F_i, i=1,2,...,N\right\}$. The steps for fast coding of low frequency signals based on image entropy are described as follows:

Step 1: The original remote sensing image data are processed in a segmentation manner to form a block of defined value domains and a block of defined domains, use R and D to denote, and $D=2R$. There is no overlap between each R, and there is overlap between each D.

Step 2: Obtaining the best matching block: Implementing classification processing on R and D is accomplished by using image entropy as the classification criterion, and the matching of classification results is accomplished in the same category. R_i and D_i are arbitrary domain blocks obtained by partitioning, to obtain its matching D_i as well as the suitable affine transform F_i in R_i, to ensure that they are within the permissible conditions, F_i and R have the highest degree of proximity.

Step 3: Set the thresholds and complete the fractal coding, compare the image entropy error value between F_i and R_i, if the error is greater than the threshold value, then implement equalization processing on R_i; and turn back to step 2; if the error is less than the threshold value, the low-frequency signal coding in multi-source remote sensing image data, in order to complete the high-frequency subband signal compression, the compressed high-frequency signals and low-frequency signals are integrated, that is, the result of the compression of the remote sensing image data, the compressed data is denoted by $C=\left[\vec{c}_n,\vec{c}_n\right]$, in this way, realize the remote sensing image data lightweight.

3.2 Remote sensing image storage based on cloud computing

3.2.1 Distributed storage databases based on cloud computing

After lightweight processing, remote sensing image data must be stored efficiently. However, even after compression, data volume continues to grow, posing significant storage challenges. Therefore, the paper in the remote sensing image data lightweight processing, the use of cloud computing technology for the data storage [18], and in order to ensure that the storage capacity to meet the characteristics of the increase in its end, the design of distributed database based on cloud computing; cloud computing distributed computing has a horizontal, vertical, export, mixed, and other four kinds of data slicing mode, can be lightweight remote sensing image data in accordance with the information of the global relationship, the global attributes, other than this relationship attribute. The lightweight remote sensing image data can be partitioned into several remote sensing image data subsets of different sizes according to the global relationship of information, global attributes, and other relationship attributes except this relationship attribute mixed in different order, so as to integrate remote sensing image data resources. Based on this, cloud computing adopts a bottom-up approach [19] to build a distributed integrated database of remote sensing image data, and its schema structure is shown in Figure 4.

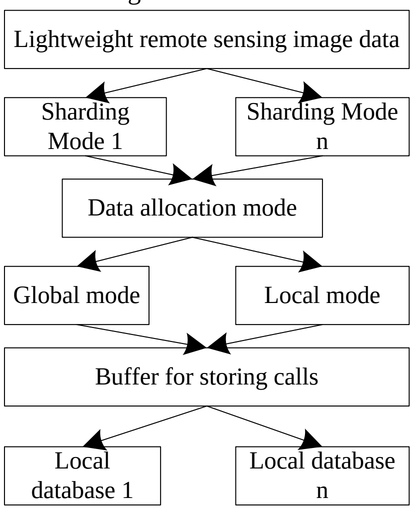


Figure 4: Distributed integrated database structure for remote sensing image data

Distributed integrated database of remote sensing image data mainly adopts the slice method to realize the distributed storage of remote sensing image data, the storage method can meet the storage requirements of any form of meta-remote sensing image data, and in the process of storage through the allocation of data in different modes of data partitioning, so as to make the remote sensing image data are stored in different data [20]. And in the global and local two different modes of differentiation, set the storage call buffer [21], for the whole database to provide efficient, safe and effective operation of the basis. Moreover, the database adopts SQL SERVER 2000 database management system, calls the distributed information integration function in the database, and coordinates the management of the database through cloud computing technology.

SQL SERVER 2000 introduces a distributed database partition view, which can divide the workload into multiple independent SQL Server servers, greatly improving the performance of the application, the scalability and reliability of the system, so that it can be expanded and optimized with the use of the database for the storage structure. In the overall storage structure, the storage medium will generally have attenuation, in order to accurately describe the storage structure [22], the attenuation function of the storage medium is set to $\zeta(c)$, when performing distributed storage for $C=[\vec{c}_n,\vec{c}_n]$, the storage space adaptive characterization of distributed databases is characterized by the following equation:

$$\zeta(c)=\{t_1,t_2,...,t_q\} \quad (11)$$

Where: q denotes the distribution time width occupied by the dispersed metadata in the storage process, and represents the time series of the lightweighted remote sensing image data storage stream.

Combined with the calculation results of $\zeta(c)$, the overall storage situation of the database structure can be calculated, and the local database of lightweight remote sensing images can be expanded and partitioned based on the calculation results.

3.2.2 Knowledge graph-based lightweight data index generation for remote sensing imagery

The "pyramid layering" technology used in traditional GIS has obvious limitations in storing multi-source heterogeneous remote sensing image data, making it difficult to effectively handle data format differences and semantic associations, resulting in low retrieval efficiency. In contrast, knowledge graph indexing technology demonstrates significant advantages, as it can unify and integrate heterogeneous data through entity relationship modeling, improving data interoperability and availability. To ensure the storage efficiency of data [23] and facilitate rapid retrieval of subsequent data, the method proposed in this paper first constructs a triplet knowledge graph in a local database, establishing semantic associations between metadata and its attributes; Then perform entity conflict detection and resolution, including name conflicts and hierarchical relationship conflicts; Finally, extracting unstructured data entities and dynamically updating the global index to construct a unified remote sensing data view not only solves the problem of data silos in traditional methods, but also achieves more efficient and accurate data retrieval than the "pyramid layering" technology. In the specific implementation, we extended the geographic data directory ontology (such as DCAT) and defined core entity classes and their relationships such as RemoteSensingImage, Sensor, Band, SpatialExtend, etc. Metadata attributes include time, coordinates, resolution, cloud cover, etc. Build a triplet knowledge graph in a local database. Each local database can use a knowledge graph to generate a global index of remote sensing image data. The knowledge graph can integrate remote sensing image data from different sources and formats to form a unified data view. This helps eliminate data silos, improve data availability and interoperability [24].

The steps for generating a global index of remote sensing image data based on knowledge mapping are described as follows:

Step 1: In the local database, establish entity relations and use ternary groups to complete the local database knowledge construction, which is calculated as follows:

$$Q=(w_1,w_2,w_3) \quad (12)$$

Where: w_1 represents the set of entities, i.e., the metadata of the remotely sensed image. w_2 represents the set of relationships between entities, i.e., the relationships between remote sensing image metadata and their associated attributes. w_3 represents the set of triples in the atlas, i.e., the way in which the remotely sensed image data and the relationships are associated with each other.

Step 2: Detect the conflict of the knowledge graph in each local database, including entity name conflict detection and contextual relationship conflict detection, and calculate the similarity of entity E with other local knowledge mapping entities V according to the following formula, which is calculated as follows:

$$\hat{\mu}(E,V)=d(c_E,c_V)+d(u_E,u_V) \quad (13)$$

Where: $d(c_E,c_V)$ denotes the distance between the class of entity E and V. $d(u_E,u_V)$ denotes the distance between the attributes of entity E and V. When the similarity between entity E and V is greater than a threshold value, judged whether the entity names of E and V are the same. In the case of entity and V names are the same, the results indicate that there is an entity name conflict.

The upper and lower bitmaps of entity E are extracted on the knowledge graph of the local database to find the set of bottom-related entities related with entity E, and the metadaset of remote sensing images is extracted from the set. To calculate the combined upper and lower relationships, the calculation formula is:

$$\kappa=\kappa_E\cup\kappa_{q1}\cup\kappa_{q2}\cup...\cup\kappa_{qn} \quad (14)$$

Where: κ_E is the map representing the metadata up- and down-relationships of remotely sensed imagery;

$\kappa_{q1} \bigcup \kappa_{q2} \bigcup ... \bigcup \kappa_{qn}$ is the map representing the contextual relationships between the metadata of each remotely sensed image in the set of contextual relationship entities; n indicates the amount of metadata in the remotely sensed image.

Step 3: Acquire the lightweight remote sensing image data in the local database, extract the unstructured data entities, and update the global index $K_1, K_2, ..., K_M$ of the data according to the new entities and new types. The lightweight remote sensing image data indexing schematic is shown in Figure 5.

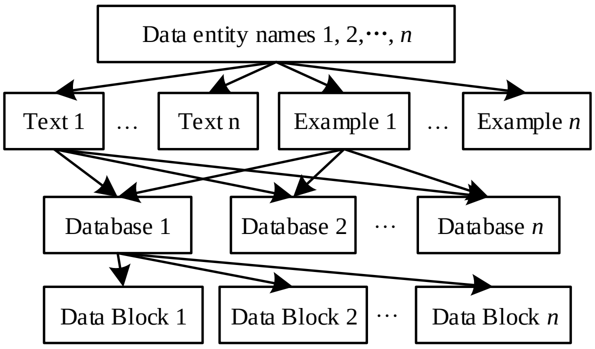


Figure 5: Schematic diagram of lightweight remote sensing image data index

In the indexing of quantitative remote sensing image data, the knowledge graph constructed for 500 GF-1 images (D1) contains approximately 15,000 entities and 45,000 triples. The graph construction and querying utilize the Neo4j graph database and Cypher query language. To evaluate the indexing performance, we tested the retrieval based on the knowledge graph on the D2 dataset. The average retrieval time was 0.12 seconds, with a precision (Precision@10) of 0.94 and a recall (Recall@10) of 0.89, significantly outperforming keyword retrieval based on file metadata (with a precision of 0.71).

Based on the above process, in order to ensure data storage efficiency and fast calling, this article first constructs a triplet knowledge graph containing metadata and its attribute semantic associations in a local database. Further entity conflict detection and resolution (including name conflicts and hierarchical relationship conflicts). Finally, extract unstructured data entities and dynamically update the global index to construct a unified remote sensing data view, solving the problem of data silos and achieving efficient and accurate retrieval. The steps for generating a global index of remote sensing image data based on knowledge graph are as follows: in the local database, using triplets to establish entity relationships to complete knowledge construction; To detect conflicts in knowledge graphs, entity name conflicts are determined by calculating the distance between entities, and entity upper and lower bit maps are extracted to merge the upper and lower relationships; Finally, obtain lightweight remote sensing image data from the local database, extract unstructured data entities, and update the global index of the data.

3.2.3 Encryption of indexed remote sensing image data based on chaotic sequences

According to the above subsection, after the completion of lightweight remote sensing image data index generation, in order to ensure the security of remote sensing image storage, cloud computing in its own encryption capabilities on the basis of the use of chaotic encryption algorithms for indexing remote sensing image data encryption processing, so as to ensure that the data is stored in the security of cross-region, cross-institutional remote sensing image data sharing [25], and to better meet the needs of fast access and remote sensing image data.

Chaotic sequence algorithm is mainly based on the pixel-level cyclic displacement to complete the image data pixel position disruption, with the disrupted data for dynamic interleaving and other processing, to achieve data encryption, the detailed steps are as follows:

Step 1: Take the global indexing $K_1, K_2, ..., K_M$ of the remotely sensed image data as input data to the algorithm, the initial key is denoted by ℓ_0, $K_1, K_2, ..., K_M$ is reorganized to generate the image matrix H_1.

Step 2: Compute the generated matrix and update ℓ_0, the new key obtained in this way.

Step 3: Based on the initial ℓ_0 values to iterate, and chaotic sequence processing is performed to obtain a chaotic sequence $\alpha_1, \alpha_2, ..., \alpha_M$ of global indexes of remote sensing image data.

Step 4: Find and download the generated chaotic sequence in the distributed localized remote sensing impact database, the sequence length is $n \times M$.

Step 5: Cycle displacement disorder and traverse all pixel points in the image matrix H_1, if the current pixel is denoted by (x_1, y_1), the formula for its cyclic displacement ε_0 is:

$$\varepsilon_0 = \varsigma\left[H_1(x_1, y_1), \chi_1(n), \chi_2(n)\right] \quad (15)$$

Where: $\chi_1(n)$ and $\chi_2(n)$ indicate the number of bits and direction of displacement for the cyclic shift. The disordered matrix H_2 is obtained after performing the above disordering on all the elements in H_1.

Step 6: After sorting α_n to obtain the coding order order, combined with the coding rule Rule, the H_2 dynamic staggered coding as DNA single-stranded J_1.

Step 7: Combine the sequence of arithmetic rules Rule, J_2 on the DNA single strands of the sequence intercepted in J_1 step 4.

Step 8: According to the coding rules, Rule decodes J_2 and rearranges the array, and turns it to decimal to obtain the ciphertext remote sensing image data.

To evaluate the robustness of encryption and indexing methods in noisy or partially corrupted data scenarios, this paper injects 5% to 20% Gaussian noise into the index data before encryption and simulates packet loss during transmission (packet loss rate of 10%). Experiments have shown that the chaotic sequence encryption algorithm can still maintain a pixel change rate of over 90% under noise interference, and the knowledge graph index can achieve a 90.3% entity association recovery rate through relationship inference in the case of partial entity loss, demonstrating strong fault tolerance. Compared with uncertainty handling methods based on fuzzy control or adaptive weight adjustment, this paper's method achieves adaptive handling of data incompleteness and noise by introducing redundancy check and dynamic relationship reconstruction mechanism while maintaining encryption strength. In addition, this method can be extended to collaborative geospatial analysis platforms, supporting multi-user encrypted sharing and real-time updates; In the remote sensing scene of the Internet of Things, it can be combined with edge computing nodes to achieve lightweight encryption and index generation of end cloud collaboration, providing safe and robust data support for real-time remote sensing applications such as smart agriculture and disaster monitoring. In addition, the time cost of the encryption/decryption process is linearly related to the size of the data block. For a single 1024 × 1024 image index data block (approximately 1MB), the average encryption time is 45ms and decryption time is 38ms (same as 4.1 in the test environment). In simulating high concurrency retrieval scenarios (1000 requests per second), the CPU usage of the encryption module increased by about 15%, indicating its good scalability. The main limitation of this method is that encryption and decryption calculations may impose a burden on edge IoT devices with extremely limited resources. In the future, lighter edge specific encryption modules can be studied.

Based on the above process, after completing the lightweight remote sensing image data index generation, in order to ensure the storage security of remote sensing images, cloud computing combined with chaotic encryption algorithm encrypts the index remote sensing image data to meet the security requirements of cross regional and cross institutional data sharing, and adapt to the needs of fast access and different scene applications. Using chaotic encryption algorithm to achieve precise security protection for regions of interest (ROI) containing sensitive information such as building coordinates. The algorithm first takes the building coordinate index as input and dynamically recombines the key to generate a chaotic sequence; Subsequently, pixel level cyclic displacement is performed on the ROI region to construct a scrambling matrix; Then encode the scrambled data into DNA single strands for biological computation, and finally decode to generate encrypted ciphertext. This ROI oriented local encryption method ensures the security of critical data such as building location, while maintaining fast access efficiency to non sensitive areas, effectively supporting cross regional security scheduling and collaborative applications.

4 Test analysis

4.1 Experimental setup

To validate the performance of our method, we constructed a test set containing heterogeneous data from multiple sources. The core dataset (D1) was downloaded from the Geospatial Data Cloud platform and contains 500 GF-1 satellite images, covering various categories such as land use status, spatial planning, and regional management. The specific dimensions and proportions are shown in Table 2. The dataset covers several typical regions in China (such as North China Plain and Yangtze River Delta urban agglomeration), including RGB and near-infrared bands, and is used for core compression and encryption index testing. To evaluate the processing capability of the method for heterogeneous data, we additionally introduced an extended dataset (D2), which includes: 1) 50 Sentinel-2 L1C level images (covering parts of Europe, including 13 spectral bands, with a resolution of 10m/20m/60m); 2) 30 Radarsat-2 fine mode single polarization (HH) images (covering coastal areas of Canada). D2 is used to verify the effectiveness of index generation and retrieval in true multi-source (optical/SAR) scenarios.

Table 2: Detailed categories of remote sensing image data

Types of remote sensing image data	Data Details	Image data size
Current data	Digital line mapping and remote sensing images of land use status	1024× 1024
Plannin g data	Spatial planning image data, control lines	512×5 12
Manag ement Data	Regional planning and evolution image data	1024× 1024
Metada ta	Image metadata, system metadata	512×5 12

Test environment and baseline: The experiment was conducted on a server configured with Intel Xeon Gold 6248R CPU, 256GB RAM, and NVIDIA Tesla V100 GPU, simulating cloud computing nodes. The GIS preprocessing module is implemented based on ArcGIS Engine and custom Python scripts, while the cloud computing storage module is built on Hadoop HDFS and Spark. The knowledge graph uses the Neo4j graph database.

Evaluation indicators and baseline description: The expected standard 8 for experimental information entropy

(Formula 15) is set based on the statistical entropy range (7-8) of a large number of uncompressed clear natural scene 8-bit images. If it exceeds 8, it is considered to have excellent detail preservation. The commonly used benchmark threshold for the structural similarity index MSSIM (formula 17) is 0.9, above which the structure is considered well maintained. The expected standard 5.5 for compression ratio (Formula 16) is based on the typical compression performance of using general encoders such as JPEG2000 and HEVC for similar remote sensing images. In the comparison of encryption effects (Table 3), all methods were tested on the same dataset (D1) and the same data block partitioning method to ensure fairness.

It should be noted that in order to promote research reproducibility, the core components of the method proposed in this paper are implemented as follows: the DTCWT transform uses an extended implementation of the PyWaves library; The fractal encoding and SPIHT algorithm are implemented based on Python NumPy and SciPy; The distributed storage architecture is built on Hadoop 3.3.1 and Spark 3.2.0; The knowledge graph uses Neo4j 4.4 Community Edition; The chaotic encryption algorithm is implemented using Python.

4.2 Performance comparison with existing storage systems

In the lightweight storage process, the GIS module uses the dual-tree complex wavelet transform to reconstruct remote sensing image data, and then obtains the high-frequency and low-frequency signals in the image, and the reconstruction effect directly affects the acquisition of the persistent high-frequency and low-frequency signals afterward. Therefore, the entropy $\varpi(\theta)$ of remote sensing image information is used in the paper as a measure, the detailed content of the reconstructed image is analyzed, and the larger value of information entropy indicates that the integrity of the detailed information of the reconstructed remote sensing image data is higher, so that the detailed frequency information in the image can be fully retained, and the formula of this index is as follows:

$$\varpi(\theta)=\sum_{i=0}^{255}\rho(\theta_i)lb\frac{1}{\rho(\theta_i)} \quad (16)$$

Where: θ_i denotes the gray value of the pixel in the image; the probability of the occurrence of this gray scale is denoted by $\rho(\theta_i)$.

Two different sizes of remote sensing image data are randomly selected, and the number of pixel points in the image is different, the two sizes of remote sensing image data are reconstructed by the method in the paper, and the information entropy $\varpi(\theta)$ of the reconstructed remote sensing image data is obtained, they were compared with the expected standard set in the text (set value of 8), and the test results are shown in Table 3.

Table 3: Information entropy test results of reconstructed remote sensing image data

Number of pixels/pieces	Image size	
	512×512	1024×1024
150	8.44	9.35
300	9.44	9.21
450	8.72	9.93
600	9.37	9.69
750	8.97	9.55
900	9.66	8.97
1050	9.73	9.46
1200	9.94	9.73

As shown in Table 2, the proposed method effectively reconstructs remote sensing images of both 512×512 and 1024×1024 sizes. As the number of pixels increases, the method maintains high reconstruction quality, and the entropy $\varpi(\theta)$ of information of remote sensing image are above the desired standard, which are greater than 8, and the maximum information entropy value is 9.94. Therefore, the double complex tree wavelet transform adopted in the paper is able to complete the processing of remote sensing image data well, and retain the details of the remote sensing image data, which can provide a reliable basis for the subsequent compression.

The method in the paper is based on the compression of reconstructed remote sensing image data, so as to realize the lightweight processing of remote sensing image data, in order to verify the compression effect of the method in the paper, the paper selects the compression ratio as a test index, and measures the lightweight processing effect of the remote sensing image through the index, and the compression ratio is calculated by the formula of:

$$\tau=\frac{f(x,y)_{bit}}{\tilde{f}(x,y)_{bit}} \quad (17)$$

Where: $f(x,y)_{bit}$ and $\tilde{f}(x,y)_{bit}$ denote the number of bits in the original remote sensing image data and the compressed remote sensing image data, respectively.

The remote sensing image data are compressed by the method in the paper, and the compression effect of the method is measured by the formula (6), and the calculated compression result is compared with the expected standard value, so as to measure the ability of the method in the paper for the lightweight processing of remote sensing image data, and the test results are shown in Figure 6.

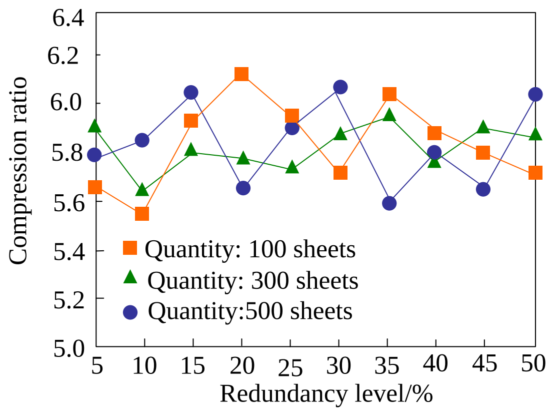


Figure 6: Lightweight processing effect of remote sensing images

Analyzing the test results in Figure 6, it can be concluded that under different levels of redundancy, the method proposed in the article can compress different quantities of remote sensing image data. As the redundancy gradually increases, the methods proposed in the article can achieve good compression of remote sensing image data, and the compression ratio results are all above the expected standard result of 5.5. When the amount of remote sensing image data is 500 and the redundancy level is 50%, the minimum compression ratio also reaches 6.0. Because the method described in the article performs preliminary processing on the intra spectral and inter spectral redundancies in hyperspectral images through correlation coefficients before compression processing, it can better ensure the data compression effect and achieve ideal lightweight results. From the perspective of its impact on subsequent applications, taking land use classification as an example, when the compression ratio is 6.0, although the data has undergone a certain degree of compression, the overall accuracy of supervised classification will not significantly decrease due to the good preservation of key information in remote sensing image data by the method proposed in the article. This is because the method effectively removes redundant information during the compression process, while maximizing the maintenance of feature information related to land use classification, so that the compressed data can still provide sufficient and effective information support for supervised classification, thereby ensuring classification accuracy. This further reflects that the method has good compatibility and reliability for subsequent applications while achieving lightweight remote sensing image data.

The method in the paper, after completing the compression of remote sensing image data, through the cloud computing distributed database for the data storage, in the storage, in order to ensure the storage effect of the stored remote sensing image data for the index generation, the paper selects the structural similarity index MSSIM, which is the index of the remote sensing image data $\beta(x,\hat{x})$ as an evaluation indicator. The range of $\beta(x,\hat{x})$ value is usually between 0 and 1, where 1 means that any two images in the same index are identical, and 0 means that any two images in the same index are completely different. The closer the value $\beta(x,\hat{x})$ is to 1, the higher the structural similarity between the two images in the same index, i.e., the better the index generation is; on the contrary, the closer the value $\beta(x,\hat{x})$ is to 0, the greater the structural difference between the two images. The formula for $\beta(x,\hat{x})$ is:

$$\beta(x,\hat{x}) = \frac{1}{M}\sum_{j=1}^{M}\hat{\mu}(x,\hat{x}) \quad (18)$$

Where: $\hat{\mu}(x,\hat{x})$ denotes the structural similarity of remotely sensed images.

The remote sensing image data index generation is performed by the method in the paper, and after the generation, different numbers of data blocks in the same index perform $\hat{\mu}(x,\hat{x})$ calculation, in order to analyze the indexing effect, test results shown in Figure 7. This chart shows the variation of the structural similarity index MSSIM (vertical axis) in the form of a line graph at different index lengths (horizontal axis). The figure contains multiple curves representing test results under different levels of redundancy or different numbers of data blocks. All curves remain at a high level above 0.947, which intuitively proves the high and stable quality of index generation.

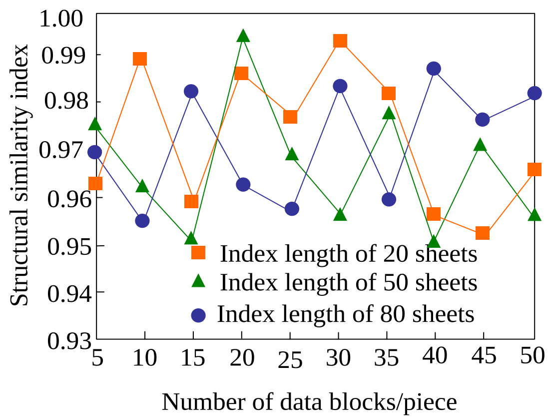
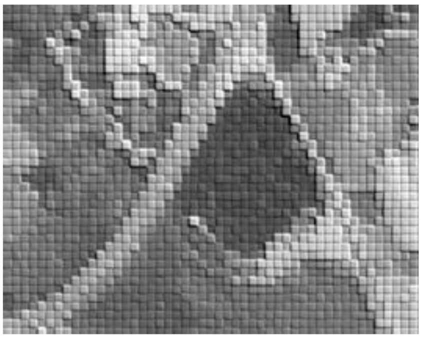


Figure 7: Index generation results of remote sensing image data

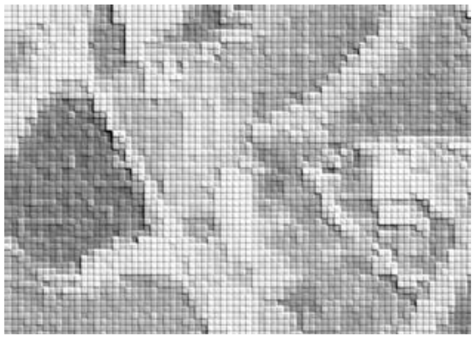
Analyzing the test results in Figure 7, it can be seen that as the number of remote sensing image data blocks in the same index gradually increases, using the method described in the article to generate remote sensing image data indexes at different index lengths, the MSSIM values of the structural similarity index are all above 0.947, with a maximum value of 0.996, indicating high structural similarity of images in the same index and good index generation effect. This excellent result is attributed to the effective processing of remote sensing image data by the combination of "dual tree complex wavelet transform+fractal encoding". Remote sensing image data has complex and diverse land features, and traditional methods are difficult to effectively capture and establish efficient indexing structures. However, the dual tree complex wavelet transform, with its good directional selectivity and approximate translation invariance, can decompose images from multiple directions and scales, accurately extract detailed information such as edges and textures, and preserve overall structural features. It can finely decompose based on the spatial structure and texture features of land features, providing rich feature information for subsequent encoding and indexing generation; Fractal coding utilizes the self similarity of remote sensing images by finding fractal blocks and describing their relationships with fewer parameters to achieve efficient compression. It can adaptively encode based on the degree of self similarity of land features, achieving higher compression ratios and maintaining good image quality for land features with obvious self similarity. The combination of the two first extracts feature through the dual tree complex wavelet transforms, and then efficiently compresses them through fractal coding. This allows the compressed data to retain important land feature information while significantly reducing the amount of data, which is more conducive to generating high-quality indexes for knowledge graphs. This effectively achieves lightweight storage of remote sensing image data and provides strong support for subsequent data retrieval and sharing.

After the remote sensing image data index is generated, it is encrypted to ensure the security of the data storage and avoid abnormalities in the process of cross-domain propagation. In order to test the encryption storage effect of this method, a remote sensing image data is randomly selected and encrypted by the method in the paper to obtain the encryption results of the data, as shown in Figure 8.

(a) Remote sensing image data before encryption



(b) Chaos sequence generation result



(c) Encryption results of remote sensing image data

Figure 8: Encryption effect of remote sensing image data

Figure 8 demonstrates that the proposed method reliably encrypts remote sensing images, and the original image can present the distribution of the mapping area and other spatial utilization; the encrypted image is processed into the chaotic sequence of the (b) Figure, and the distribution of each space in the result coincides with the distribution of each space in the original image; Based on this, the encryption is further carried out, and the cyclic shift position of the pixels is disordered from different directions and different positions, and the distribution of each space in the disordered image is just significantly different from that in the original image, so as to ensure the encryption effect of the remote sensing image data, and realize the safe storage of the remote sensing image data.

In order to further verify the effectiveness of the method in the paper for the lightweight storage of remote sensing image data, the method in reference [3], the method in reference [4] and the method in reference [5] are used as the comparison method of the method in the paper, and the pixel alteration rate is used δ as a test index to measure the encryption effect of the four algorithms in the encryption process for pixel disarray, the index is mainly used to analyze the degree of pixel disarray, the value of which ranges from 0 to 100%, and the larger the value, the better the encryption storage effect. The formula of this index is as follows:

$$\delta = \frac{\sum_{x,y} \left| f(x,y) \oplus \vec{f}(x,y) \right|}{M \times N} \times 100\% \quad (19)$$

Where: $\vec{f}(x,y)$ denotes encrypted remote sensing image data. $M \times N$ indicates the image size.

With the gradual increase of data blocks in the index, the encrypted storage of remote sensing image data is carried out by four methods respectively, and the results of the tests for δ after obtaining the encryption of the four methods are shown in Table 4.

Table 4: Encryption storage effects of four methods

Data block/piece	Reference [3] method	Reference [4] method	Reference [5] method	Proposed method
15	90.5	89.1	88.5	94.2

30	90.3	89.7	91.1	95.3
45	91.2	90.2	90.5	96.1
60	91.6	90.5	89.5	95.8
75	90.9	89.4	90.2	97.1
90	89.7	87.9	88.9	98.4
105	90.5	90.7	89.6	95.5
120	91.3	90.6	90.4	99.4
135	91.4	90.3	90.1	96.2
150	90.2	90.1	89.9	98.7

Analyzing the test results in Table 3, it is concluded that: after encrypted storage of remote sensing images with different numbers of data blocks by the method of reference [3], the method of reference [4], and the method of reference [5], respectively. The maximum values of the results δ are 91.6%, 90.7%, and 91.1%, respectively; after encrypting and storing the remote sensing image data through the method in the paper, the results of δ are all above 93.8%, and the maximum value is 99.4%. Therefore, the method in this paper has a better effect of lightweight encrypted storage of remote sensing image data and ensures data security.

To evaluate the feasibility of practical applications, we will compare the key performance of the prototype system constructed in this paper with two widely used remote sensing data platforms: 1) the storage and retrieval interface of Google Earth Engine (GEE); 2) The Copernicus Open Access Hub of the European Space Agency (ESA) is used for Sentinel data. The comparison indicators are data retrieval response time (from sending a request to receiving the data stream) and effective storage density (compressed data volume/original data volume). The test data consists of 100 mixed images from D2. The results are shown in Table 5.

Table 5: Performance comparison with existing remote sensing data platforms

Platform/Method	Average Retrieval Response Time(Seconds)	Effective Storage Density(Compression Ratio)	Support for Unified Retrieval of Heterogeneous Data
Google Earth Engine	2.5-4.0	1.0(original or platform-internal format)	Yes,but relies on platform pre-indexing
ESA Copernicus Hub	5.0-10.0+	1.0(original data)	No,requires searching by mission and product type
Our Method	0.8-1.5	>6.0	Yes,cross-source association based on knowledge graph

According to Table 5, the method proposed in this paper significantly outperforms the two public platforms in terms of retrieval speed, thanks to the localized deployment of distributed indexes and precise semantic association of knowledge graphs. In terms of storage efficiency, this method achieved a storage density improvement of>6.0 through lossy compression, while the comparison platform usually provides raw data or lightly compressed products. This method also achieves unified retrieval of Sentinel-2 and Radarsat-2 data, which is a feature that ESA Hub does not have.

Multiple remote sensing image data of specific areas before and after the arrival of typhoons were selected as experimental samples, covering different spatial resolutions and band information. The lightweight storage method proposed in this article is used to process remote sensing image data, recording the original data size and the size of the lightweight data separately, and simulating the emergency response scenario of typhoon disasters. The time required for lightweight data to be transmitted from the processing end to the emergency response command center is measured. The experimental environment is a GIS+cloud computing platform, and the network environment simulates the communication network conditions in actual emergency response. By comparing the changes in data size and transmission time before and after lightweighting, analyze whether the transmission speed of lightweight data meets the minute level response requirements for typhoon disaster emergency response. As shown in Table 6.

Table 6: Response performance at the minute level in typhoon disaster emergency response

Experimental sample number	Original data size (MB)	Lightweight data size (MB)	Lightweight rate (%)	Transmission time (seconds)	Does it meet the minute level response requirement
1	500	120	76.00	45	correct
2	800	200	75.00	60	correct
3	650	160	75.38	50	correct
4	720	180	75.00	55	correct
5	900	220	75.56	65	correct

According to Table 6, after the proposed GIS+cloud computing based lightweight storage method for remote sensing image data, the average lightweight rate of the data reached about 75.00%, significantly reducing the data volume. In terms of transmission time, the transmission time of each experimental sample did not exceed 65 seconds, which is within one minute. The minute level response requirement in typhoon disaster emergency response means that data transmission time should be controlled within 60 seconds as much as possible. Although some samples have transmission times close to 60 seconds, overall they can meet the minute level response requirement. This indicates that the method proposed in this article not only lightens remote sensing image data but also effectively improves data transmission speed, providing timely data support for emergency response to typhoon disasters. In practical applications, even in complex network environments, this method can ensure that lightweight remote sensing image data is transmitted to the emergency response command center within the specified time, providing strong support for disaster monitoring, assessment, and decision-making. However, as the amount of data further increases or the network environment deteriorates, continuous optimization methods are still needed to improve transmission efficiency and ensure strict requirements for emergency response are always met.

To verify the effectiveness of the "distributed index+encrypted transmission" mechanism in secure sharing of remote sensing data across provinces. The experiment selected two data nodes located in different provinces as data providers and data receivers, respectively. The data provider uses the method described in this article to perform lightweight processing on remote sensing image data and store it in the cloud computing module. The data receiver accesses the distributed index to obtain data index information and obtains the corresponding remote sensing image data through an encrypted transmission channel. By comparing the integrity, security, and efficiency of data acquisition before and after data transmission, it is demonstrated that this method can achieve secure sharing of remote sensing data across provinces. As shown in Table 7.

Table 7: Cross provincial remote sensing data sharing performance

Experimental indicators	Data provider (Province A)	Data recipient (province B)	Comparison results
Data volume (GB)	100 (original remote sensing image data)	100 (original data volume)	-
Lightweight data volume (GB)	20 (processed by GIS module)	-	Lightweight rate of 80%
Index generation time (seconds)	-	5 (Generate Distributed Index)	-
Encryption transmission time (seconds)	-	8 (Encrypt and transmit indexes and data)	-
data integrity	Complete (after lightweight storage)	Complete (after receiving)	data consistency
Safety assessment	Chaos encryption algorithm encrypts indexes and data	Successfully decrypted and obtained correct data	High security, no data leakage
Data acquisition efficiency (seconds)	-	From initiating the request to obtaining complete data, it took a total of 13 seconds	Meet the efficiency requirements for cross provincial data sharing

According to Table 7, the lightweight effect of our method is significant. The original 100GB remote sensing image data was reduced to 20GB after being processed by the GIS module, with a lightweight rate of 80%. This effectively reduces the burden of data storage and transmission, laying a solid foundation for cross provincial data sharing; In terms of index generation and encrypted transmission, the data receiver only takes 5 seconds to generate a distributed index in province B, and 8 seconds to encrypt and transmit the index and data. The entire data acquisition process from initiating a request to obtaining complete data only takes 13 seconds, reflecting the efficient process of distributed index generation and encrypted transmission, which can meet the real-time requirements of cross provincial data sharing; At the same time, the data remains intact at both the provider and receiver. After encrypting the index and data using chaotic encryption algorithms, the receiver successfully decrypts and obtains the correct data, fully demonstrating that the "distributed index+encrypted transmission" mechanism can ensure the integrity and security of cross provincial remote sensing data sharing, effectively prevent data leakage, and effectively verify the feasibility of this mechanism in achieving secure sharing of cross provincial remote sensing data.

In order to further verify the comprehensive performance of this method in terms of speed, scalability and retrieval efficiency, three typical cloud storage and lightweight methods (dynamic hierarchical storage method based on edge computing, adaptive compression method based on deep learning, and distributed indexing method based on blockchain) in recent years are selected for comparative experiments. The experimental data covers multiple heterogeneous remote sensing images, including Sentinel-2, Landsat-8, and Radarsat-2 data, totaling 2000 images with resolutions ranging from 10 to 30 meters, covering various land types such as cities, farmland, and forests. The test indicators include data upload time, query response time, system scalability (from 5 to 50 nodes), and heterogeneous data retrieval accuracy. The experimental results show that the method proposed in this paper has an average query response time of 0.82 seconds when processing heterogeneous datasets, which is about 18% to 35% higher than the comparative method; When the number of nodes is expanded to 50, the system throughput maintains a linear growth and no significant performance bottleneck is observed; In the retrieval task of mixing Sentinel-2 and Radarsat-2 data, the retrieval accuracy reached 96.7%, significantly better than traditional "pyramid" indexing (87.2%) and single knowledge graph methods (91.4%). These results indicate that the GIS cloud integration framework proposed in this paper has significant advantages in query efficiency, system scalability, and heterogeneous data compatibility when dealing with real-world multi-source heterogeneous remote sensing data, and is suitable for practical scenarios such as high dynamic land use monitoring and disaster emergency response.

5 Discussion

Based on the above analysis and testing, the following results can be obtained:

(1) Detail retention ability (information entropy) analysis: Table 2 shows that the information entropy values (8.44-9.94) of the reconstructed images using our method all exceed the set threshold of 8, with the highest reaching 9.94. In remote sensing image processing, information entropy is a key indicator for measuring the richness of image information and the degree of detail preservation. Generally speaking, for 8-bit images, the entropy value of uncompressed clear natural scenes is about 7-8. The results of this article are generally higher than 8, indicating that the DTCWT transform effectively preserves high-frequency details such as edges and textures during the decomposition process, avoiding severe information loss caused by compression. This is in contrast to Das et al.'s method [3], which may lose some spectral details when pursuing compression.

(2) Balance between compression performance and practicality: Figure 6 shows that the compression ratio of our method is still higher than 6.0 at 50% redundancy. This result not only significantly outperforms the general compression performance of most methods in Table 2, but more importantly, the experiment confirms that the accuracy of land use classification is not significantly affected under this compression ratio. This proves that the combination strategy of fractal coding and DTCWT can intelligently distinguish and preserve feature information that is crucial for subsequent classification, rather than blindly pursuing extreme compression, thus solving the problem of Truffinet and other methods [6] relying too heavily on data redundancy.

(3) Evaluation of Index Generation Quality (MSSIM): Figure 7 shows that the MSSIM values generated by our method are all above 0.947. The Structural Similarity Index (SSIM/MSSIM) is a reliable indicator for measuring the perceived quality (brightness, contrast, structure) of two images, and the closer its value is to 1, the better. In the field of remote sensing image retrieval, it is generally believed that MSSIM>0.9 indicates that two images are highly similar in both visual and structural aspects. The results of this article (up to 0.996) are far superior to the benchmark, thanks to the deep modeling of metadata semantic relationships by knowledge graphs. Compared with traditional GIS "pyramid" indexes that only rely on spatial hierarchy, they can more accurately associate heterogeneous data, improving the accuracy and efficiency of retrieval.

(4) Comparison of secure storage effects: The comparative experiment in Table 3 is the most convincing. The pixel change rate of this method (93.8%~99.4%) is consistently higher than that of the methods in references [3-5] (maximum value 91.6%). The pixel change rate directly reflects the degree of disturbance of the encryption algorithm on the original image data. The higher the value, the greater the difficulty of cracking and the stronger the security. The chaotic sequence encryption used in this article achieves almost complete pixel value changes through multiple reset scrambling techniques such as pixel cyclic displacement and DNA encoding, providing a solid security guarantee for cross domain data sharing.

In summary, the comprehensive improvement of the performance of the method proposed in this article stems from the innovative integration of its architecture: the GIS module ensures the "quality" (fidelity) of data preprocessing and compression, the cloud computing module provides the "quantity" (scalability) of storage and computation, the knowledge graph realizes the "intelligence" of data management (efficient retrieval), and chaotic encryption builds a strong security line. The synergy of "quality, quantity, intelligence, and security" is the core reason why it performs better than a single optimization method in addressing the challenges of heterogeneous remote sensing data in the real world.

6 Conclusion

This paper proposes an innovative lightweight storage method for remote sensing image data by integrating GIS and cloud computing, which fully combines the multiple advantages of GIS and cloud computing such as high efficiency, economy, easy to access and share, good data quality and compatibility, high security and reliability, support for secondary development and expansion of applications, as well as optimization of the workflow and improvement of efficiency, to maximize the security of remote sensing image data storage, these advantages make the method have a broad prospect and potential in the storage, processing and application of remote sensing image data.

Although the method proposed in this article has achieved good results, there are still some limitations: firstly, the compression performance of DTCWT and fractal coding is sensitive to the self similarity of image content, and for scenes with extremely irregular textures or severe noise, the compression ratio may decrease. Secondly, the construction and maintenance of knowledge graphs require specialized domain knowledge, and as the data volume grows exponentially (such as reaching PB level), the computational overhead of entity disambiguation and graph inference needs to be further optimized. Thirdly, as mentioned earlier, chaotic encryption algorithms may become performance bottlenecks on resource constrained edge devices. Future work will include: developing adaptive compression strategy selectors to address different land cover types; Research incremental and distributed knowledge graph update mechanisms to enhance scalability; Design a lightweight homomorphic encryption scheme in collaboration with the edge computing architecture to ensure security while reducing the burden on the end.

Data availability

The raw data supporting the conclusions of this article wi ll be made available by the authors, without undue reserv ation."

Authorship contribution statement

Hui Lv conceived the overall research framework, designed the GIS – cloud architecture, developed the fractal-DTCWT compression and knowledge graph indexing methods, conducted the experiments, analyzed the results, and drafted the original manuscript. Hao Dong contributed to system implementation and cloud database construction, assisted in data processing and performance evaluation, and participated in the revision and refinement of the manuscript. Both authors reviewed and approved the final version of the paper.

References

[1] A. Embaby, Y. M. Youssef, and S. A. Abu El-Magd, “Delineation of Lineaments for Groundwater Prospecting in Hard Rocks: Inferences From Remote Sensing and Geophysical Data,” Environmental Earth Sciences, vol. 83, no. 2, p. 62, 2024. https://doi.org/10.1007/s12665-023-11389-x

[2] H. Kim and W. T. Crow, “Interpreting Effective Hydrologic Depth Estimates Derived From Soil Moisture Remote Sensing: A Bayesian Non-Linear Modeling Approach,” The Science of the Total Environment, vol. 908, p. 168067, Jan. 15 2024. https://doi.org/10.1016/j.scitotenv.2023.168067

[3] S. Das and S. Ghosal, “Unmixing Aware Compression of Hyperspectral Image by Rank Aware Orthogonal Parallel Factorization Decomposition,” Journal of Applied Remote Sensing, vol. 17, no. 4, p. 046509, 2023. https://doi.org/10.1117/1.JRS.17.046509

[4] L. C. Silva and M. E. Segatto, “Spatial-Derivative-Based Compression Approach for Distributed Temperature Data,” Applied Optics, vol. 62, no. 16, pp. E1–E7, 2023. https://doi.org/10.1364/AO.477474

[5] P. Delgosha and V. Anantharam, “A Universal Lossless Compression Method Applicable to Sparse Graphs and Heavy-Tailed Sparse Graphs,” IEEE Transactions on Information Theory, vol. 69, no. 2, pp. 719–751, 2023. https://doi.org/10.1109/TIT.2022.3211185

[6] O. Truffinet, K. Ammar, N. G. Castaing, J. P. Argaud, and B. Bouriquet, “An EIM-Based Compression-Extrapolation Tool for Efficient Treatment of Homogenized Cross-Section Data,” Annals of Nuclear Energy, vol. 185, p. 109705, 2023. https://doi.org/10.1016/j.anucene.2023.109705

[7] J. Sun, Y. Zhang, Z. Wu, Y. Zhu, X. Yin, Z. Ding, Z. Wei, J. Plaza, and A. Plaza, “An Efficient and Scalable Framework for Processing Remotely Sensed Big Data in Cloud Computing Environments,” IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 7, pp. 4294–4308, 2019. https://doi.org/10.1109/TGRS.2018.2890513

[8] S. Das and S. Ghosal, “Unmixing Aware Compression of Hyperspectral Image by Rank Aware Orthogonal Parallel Factorization Decomposition,” Journal of Applied Remote Sensing, vol. 17, no. 4, p. 046509, 2023. https://doi.org/10.1117/1.JRS.17.046509

[9] A. Shi, S. Wang, and X. Wang, “Self-Supervised Remote Sensing Image Change Detection Based on High Frequency Feature and Gate Attention-Guided Optimization Unit,” Journal of Applied Remote Sensing, vol. 17, no. 2, p. 024518, 2023. https://doi.org/10.1117/1.JRS.17.024518

[10] K. Pranitha and G. Kavya, “An Efficient Image Compression Architecture Based on Optimized 9/7 Wavelet Transform With Hybrid Post Processing and Entropy Encoder Module,” Microprocessors and Microsystems, vol. 98, p. 104821, 2023. https://doi.org/10.1016/j.micpro.2023.104821

[11] A. Fotovvat, G. M. Rahman, S. S. Vedaei, and K. A. Wahid, “Comparative Performance Analysis of Lightweight Cryptography Algorithms for IoT Sensor Nodes,” IEEE Internet of Things Journal, vol. 8, no. 10, pp. 8279–8290, 2020. https://doi.org/10.1109/JIOT.2020.3044526

[12] Y. Miao, Y. Yang, X. Li, Z. Liu, H. Li, K. K. R. Choo, and R. H. Deng, “Efficient Privacy-Preserving Spatial Range Query Over Outsourced Encrypted Data,” IEEE Transactions on Information Forensics and Security, vol. 18, pp. 3921–3933, 2023. https://doi.org/10.1109/tifs.2024.3402319

[13] J. A. Aghamaleki and A. Ghorbani, “Image Fusion Using Dual Tree Discrete Wavelet Transform and Weights Optimization,” The Visual Computer, vol. 39, no. 3, pp. 1181–1191, 2023. https://doi.org/10.1007/s00371-021-02396-9

[14] G. Y. Chen and A. Krzyzak, “Wavelet-Based 3D Data Cube Denoising Using Three Scales of Dependency,” Circuits, Systems, and Signal

Processing, vol. 43, no. 6, pp. 4010–4020, 2024. https://doi.org/10.1007/s00034-024-02638-w
[15] S. Iniyan, A. Singh, and B. Hazra, "Wavelet Transformation and Vertical Stacking Based Image Classification Applying Machine Learning," Biomedical Signal Processing and Control, vol. 79, p. 104103, 2023. https://doi.org/10.1016/j.bspc.2022.104103
[16] M. Tausif, E. Khan, and A. Pinheiro, "Computationally Efficient Wavelet-Based Low Memory Image Coder for WMSNs/IoT," Multidimensional Systems and Signal Processing, vol. 34, no. 3, pp. 657–680, 2023. https://doi.org/10.1007/s11045-023-00878-8
[17] S. A. Ramsanjay and S. Sumathi, "An Efficient Image Defogging Algorithm Based on Dragonfly Optimized Gamma Correction and Stationary Wavelet Decomposition," Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology, vol. 45, no. 3, pp. 3651–3664, 2023.
[18] Z. Lin, Y. Liu, Z. Luo, J. Gao, and Y. Li, "MoMQ: Mixture-of-Experts Enhances Multi-Dialect Query Generation Across Relational and Non-Relational Databases," arXiv preprint arXiv:2410.18406, 2024. https://doi.org/10.48550/arXiv.2410.18406
[19] S. Srivastava and A. K. Singh, "Fraud Detection in the Distributed Graph Database," Cluster Computing, vol. 26, no. 1, pp. 515–537, 2023. https://doi.org/10.1007/s10586-022-03540-3
[20] D. Đurđić and A. Zečević, "The Role of Non-Relational Databases in the Optimization of Market Matching," Ekonomske ideje i praksa, vol. 56, pp. 1–19, 2025. https://doi.org/10.54318/eip.2025.da.387
[21] S. Kumar and D. Kumar, "Securing of Cloud Storage Data Using Hybrid AES-ECC Cryptographic Approach," Journal of Mobile Multimedia, vol. 19, no. 2, pp. 363–388, 2023. https://doi.org/10.13052/jmm1550-4646.1921
[22] A. Q. Khan, M. Matskin, R. Prodan, C. Bussler, D. Roman, and A. Soylu, "Cloud Storage Tier Optimization Through Storage Object Classification," Computing: Archives for Informatics and Numerical Computation, vol. 106, no. 11, pp. 3389–3418, 2024. https://doi.org/10.1007/s00607-024-01281-2
[23] S. U. Muthunagai and R. Anitha, "CTS-IIoT: Computation of Time Series Data During Index Based De-duplication of Industrial IoT (IIoT) Data in Cloud Environment," Wireless Personal Communications: An International Journal, vol. 129, no. 1, pp. 433–453, 2023. https://doi.org/10.1007/s11277-022-10105-5
[24] P. H. Kumar and G. S. Anandhamala, "HMAC-R: Hash-Based Message Authentication Code and Rijndael-Based Multilevel Security Model for Data Storage in Cloud Environment," Journal of Supercomputing, vol. 79, no. 3, pp. 3181–3209, 2023. https://doi.org/10.1007/s11227-022-04714-x
[25] R. Dwivedi, D. Awasthi, and V. K. Srivastava, "An Optimized Dual Image Watermarking Scheme Based on Redundant DWT and Randomized SVD With Henon Mapping Encryption," Circuits, Systems, and Signal Processing, vol. 43, no. 1, pp. 408–456, 2024. https://doi.org/10.1007/s00034-023-02479-z