IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 25, 2020, accepted October 4, 2020, date of publication October 13, 2020, date of current version October 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3030700

A Data Allocation Strategy for Geocomputation Based on Shape Complexity in A Cloud Environment Using Parallel Overlay Analysis of Polygons as an Example

KANG ZHAO[1], BAOXUAN JIN[2], HONG FAN[1], AND MEI YANG[1]
[1]State Key Laboratory for Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan 430079, China
[2]Information Center, Department of Natural Resources, Kunming 650224, China

Corresponding authors: Baoxuan Jin (jbx@yngc.org) and Hong Fan (hfan3@whu.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 41661086, and in part by the Yunnan Fundamental Research Projects under Grant 202001AT070093.

ABSTRACT Given the explosive growth of geospatial data, parallel computing technologies have become widely used in the spatial analysis of these massive types of data. The data used in geographic computing often exhibit a complex graphic structure, which is an important cause of data skew in parallel computing. The shape complexity is crucial to the task allocation strategy of parallel computing. The effect of polygon shape features on the performance of spatial analysis was investigated in this study. A quantitative polygon-shaped complexity evaluation model was established through regression analysis. The Hilbert data partition strategy weighted by shape complexity was used as a spatial data allocation method for parallel spatial analysis. This study established a shape complexity evaluation model for overlay analysis and used the Spark parallel computing paradigm to carry out a comparative experiment of a massive, complex polygon. Experimental results showed that the spatial data allocation strategy based on the complexity of polygon shape computing effectively solved the problem of data skew in the parallel spatial analysis of massive complex polygons.

INDEX TERMS Big data, shape complexity, overlay analysis, data skew, parallel computing.

I. INTRODUCTION

In recent years, geospatial science has faced challenges in data-intensive computing with the explosive growth of geospatial data [1]. High-performance parallel computing is generally considered an effective solution for improving massive spatial data processing and large-scale computing applications to solve the challenges brought by complex algorithms and massive data [1]–[5]. The efficient processing of a large number of complex graphics is the objective of high-performance geographic computing. The spatial analysis of millions or even tens of millions of complex geographic graphics has become a common demand. For example, the number of geological patches in many Chinese provinces often reaches tens of millions. Many polygonal patches exhibit considerable vertices and complex hole structures. The difference in the shape complexity of geographic graphics often leads to serious data skew in parallel spatial analysis, thereby greatly reducing the efficiency of parallel computing.

Overlay analysis is a common task in geographic computing that is widely used in geographic information systems, computer graphics, and computer science. This study conducted an in-depth analysis of the effect of the polygon features on overlay analysis based on the classical Greiner–Hormann algorithm [6], and quantified the factors that affect overlay evaluation analysis, established a polygon shape complexity evaluation model. Moreover, through the combination of Hilbert space-filling curve [7], this study designed a calculation task allocation and optimization strategies on the basis of polygon shape complexity, which effectively solved the data skew problem and greatly improved the efficiency of parallel overlay analysis.

The associate editor coordinating the review of this manuscript and approving it for publication was Daniel Grosu.

The remainder of this work is organized as follows. Section 2 reviews the research background and related studies. Section 3 analyzes and quantifies the effect of typical shape features on the computational complexity of spatial analysis. The complexity measurement model of the polygon shape is established using overlay analysis as an example. Subsequently, the design of the computing task allocation strategy of parallel overlay analysis on the basis of the measurement model is described. Section 4 presents the experimental verification and results analysis. Section 5 provides the conclusion drawn from this research, followed by potential future work.

II. RELEVANT WORK

A. DATA BALANCED DISTRIBUTION IN PARALLEL GEOGRAPHIC COMPUTING

The reasonable distribution of spatial data is a key factor for improving the query and calculation performance of geographic big data [29]. The basic principle of parallel computing divides a complete data block into relatively small and independent multiblock datasets, improves the I/O performance of data through parallel access, and provides the basis for distributed or parallel data operation. When MPI, MapReduce, Spark, and other parallel computing frameworks for distributed clusters are applied in spatial data processing, the rational allocation of spatial data is widely considered. Spatial big data processing systems, such as spatial Hadoop [30], Hadoop GIS [31], and MD HBase [32], extend Hadoop. These systems realize the spatial data partition method on the basis of grids, R-trees, R+-trees, Z-curves, Hilbert curves, quad trees, and k–d trees. The aforementioned systems also provide a method for assigning the specified partition and multipartition repeated storage for the cross-partition objects [30]–[34]. GeoSpark [35] and LocationSpark [36] expanded the resilient distributed dataset (RDD) to the spatial resilient distributed dataset (SRDD) that supports spatial object storage on the basis of Apache Spark. Grid partition and coding are used for all SRDDs, and index construction, such as quad trees and R-trees, are supported for SRDDs [37], [38]. These optimization methods improve the parallel nature of spatial data processing on the basis of spatial partitioning. However, the equal number of spatial objects or the same storage capacity of each partition are taken as the basis of spatial data partition in these studies, ignoring the effect of individual differences of each spatial object on computing time. The result is that although each partition has the same amount of data, the calculation intensity varies greatly.

The allocation of spatial data is involved in specific parallel spatial analyses. Wang [39] and Zheng [40] used grid division for spatial data distribution in parallel overlay computing. Zhao [28] used Hilbert division and R-trees for the division of spatial data in parallel overlay computing, which has higher parallel efficiency than grid division. Fan [41] and Zhao [28] regarded the number of polygon vertices as the factor of data allocation in parallel overlay analysis. However, the number of polygon vertices alone cannot fully reflect the effect of polygon shape features on overlay analysis time. Obviously, a comprehensive evaluation of the effect of polygon shape features on overlay analysis performance is helpful to optimize spatial data partitioning strategy.

B. SHAPE COMPLEXITY

Graphics are ubiquitous in nature. The concept of shape complexity has been proposed [9], [10] and widely used in many fields to describe the differences in the complexity of graphic shapes, however, no unified shape measurement standard has been developed to date. Under a specific research level and granularity [11], existing studies introduce different measurement methods for shape complexity.

Geometric formulas are widely used to describe the linear features of object shape, and statistical physical factors, such as the fractal dimension [12], are used to describe the complex nonlinear features of the real world. The fractal dimension quantifies the self-similar features [13] of spatial objects through convolution, which is not limited to the quantification of 2D graphics. In follow-up research, the fractal dimension and image pattern recognition [14], [15] have been extended to measure the spatial differences of various complex shapes [16], [17] in nature. The fractal dimension quantifies the local features of an object; however, the overall shape does not affect the fractal dimension [13]. Fractal dimension has low differentiation with shape complexity when few polygon vertices are present. In the field of computer graphics, shape complexity exhibits a global shape and local features. Chen [18] used global distance entropy, local angle entropy, and the shape factor to quantify shape complexity. Su [19] used graphic boundaries, global structures, and symmetry to measure shape complexity. Saleem [20] used a similarity matrix of 2D graphics to describe the complexity of 3D graphics. Matsumoto [21] used the absolute curvature to quantify the surface shape complexity of the object. Duan [22] utilized the combination of the area perimeter ratio of the graphics and the number of blocks to represent the shape complexity. Although these methods require graphical display and design, they do not directly reflect the relationship between shape complexity and calculation efficiency.

In geographic computing, computational efficiency is considered when expressing shape complexity, which is often expressed as a function of the problem size and computational cost [10]. Chazelle [23] described the shape complexity of a polygon on the basis of the curvature (i.e., the frequency at which the boundary alternately changes in the opposite direction). This method does not involve the number of polygon vertices and can effectively solve the polygon triangle. Brinkhoff [13] selected the relevant structure, measurement, and statistical parameters to create a complexity model for the polygon. Guo [24] used the number of polygon features and that of vertices of each feature to quantify the complexity of the object. Accordingly, reasonable data division in parallel computing has been realized. Ying [26] used the concept of Brinkhoff complexity to simplify the expression of the

polygon to improve the speed of graphic data transmission. Li [27] defined the complexity calculation for line and surface elements with two factors (e.g., area coefficient and angle coefficient) and quantified the geometric information on sensitive elements in maps. Zhao [28] used the number of polygon vertices to estimate the complexity of polygons in parallel overlay analysis. The aforementioned method solved the problem of complexity measurements in specific geographic computing but is limited to specific questions.

In summary, existing methods for measuring shape complexity mainly focus on how to express complex graphics, while the research on the effect of shape features on the efficiency of spatial analysis is lacking. Therefore, the application of existing shape complexity measurement methods in parallel spatial calculations is difficult. The difference in the graphic shape causes variation in the time cost of the spatial analysis, which is a key content of this article.

III. METHODOLOGY

The information of spatial data includes two parts: spatial graph and attribute information. Among them, the attribute information has the same field structure and storage space, but the spatial graph may be quite different. Therefore, the difference in execution time of different spatial objects processed by the same spatial analysis algorithm is mainly affected by the differences in spatial objects' graphics. In the parallel overlay analysis, we found out the shape feature factors that affect the time cost of the algorithm, and created the polygon shape complexity evaluation model by using the stepwise regression method, so as to quantified the computational strength in the overlay analysis of each polygon. Moreover, we designed a data allocation strategy based on Hilbert filling curve and polygon shape complexity.

A. POTENTIAL EFFECT OF POLYGON SPATIAL FORM ON SPATIAL ANALYSIS

Polygonal graphics have different shape features. A simple polygon only has a few vertices with regular shape (Fig. 1 [a], [b], and [c]), and a geographic polygon has a complex shape (Fig. 1[d]) that is irregular and can have tens of thousands of vertices, uncertain concave or convex boundaries, many complex holes and island structures, and different locations. The principle of Greiner – Hormann algorithm is to get the intersection part of two polygons by finding the intersection points of two polygons. Therefore, the polygon shape features that affect the number of polygon intersection operations are the potential factors determining the time cost of Greiner – Hormann algorithm. We selected the potential factors that affect the calculation of shape complexity from the aspects of local features and spatial distribution to fully describe the effect of the feature differences of complex polygons on the performance of overlay analysis.

- Number of polygon vertices

A vertex is the basic element of a graphic and consists of a pair of coordinates. According to the number of vertices, the number of edges can be determined, and the spatial measurement calculation (e.g., calculation of length and area) can be carried out on this basis. The intersection relationship of the graphs (e.g., whether points intersect lines and faces) can also be determined. Therefore, the number of vertices is an important factor that affects the performance of the algorithm.

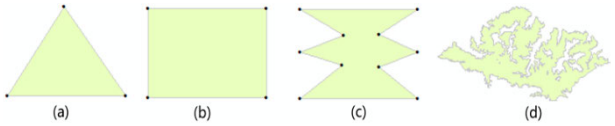

FIGURE 1. Different shapes of polygons.

- Density of polygon vertices

The polygon vertex density reflects the number of vertices per unit area. The polygon structure is generally highly complex when the density of polygon vertices is high. If the vertex number and area of polygon P are n and A_p, then the vertex density of the polygon can be expressed as follows: $D_p = \frac{n}{A_p}$.

- Spatial aggregation of polygon vertices

We used the average nearest neighbor (ANN) ratio to measure the spatial clustering of vertices. The polygon is highly complex when the ANN ratio of the vertices is small, and the degree of spatial aggregation is high. ANN can be expressed as follows: $ANN = \frac{\bar{D}_0}{\bar{D}_E}$, where $\bar{D}_0$ is the ANN distance [42], $\bar{D}_0 = \frac{\sum_{i=1}^{n} \mathrm{d}_i}{n}$, $\bar{D}_E$ is the expected average distance, $\bar{D}_E = \frac{0.5}{\sqrt{n/A}}$, d_i is the nearest distance from each point to other points, n is the number of vertices of the polygon, and A is the area of each polygon.

- Number of ring structures

A complex polygon often has several holes, islands, and other ring structures. In the intersect operation and spatial graphical differences, the number of intersections and difference operations will be increased by the number of holes and islands, thereby increasing the calculation complexity and reducing the performance of spatial analysis.

- Number of concaves

The processing of concave polygons in the spatial analysis that involves polygons is more complex than that of convex polygons [43]. The vibration variation of the polygon edge and the potential computational complexity are great when many concaves are present.

- Concavity

Considering that the number of concaves does not reflect the degree of concavity of the polygon, this study uses Brinkhoff's definition of convexity [13] for reference and defines this relationship with concavity to quantitatively describe the concavity of the polygon.

Definition 1: If the area of polygon P is A_p, and the convex hull is A_{pch}, then the concavity of the polygon is expressed as $Con_{pg} = \frac{A_{pch} - A_p}{A_{pch}}$.

When the polygon concavity is large, the potential effect on its spatial analysis is also great.

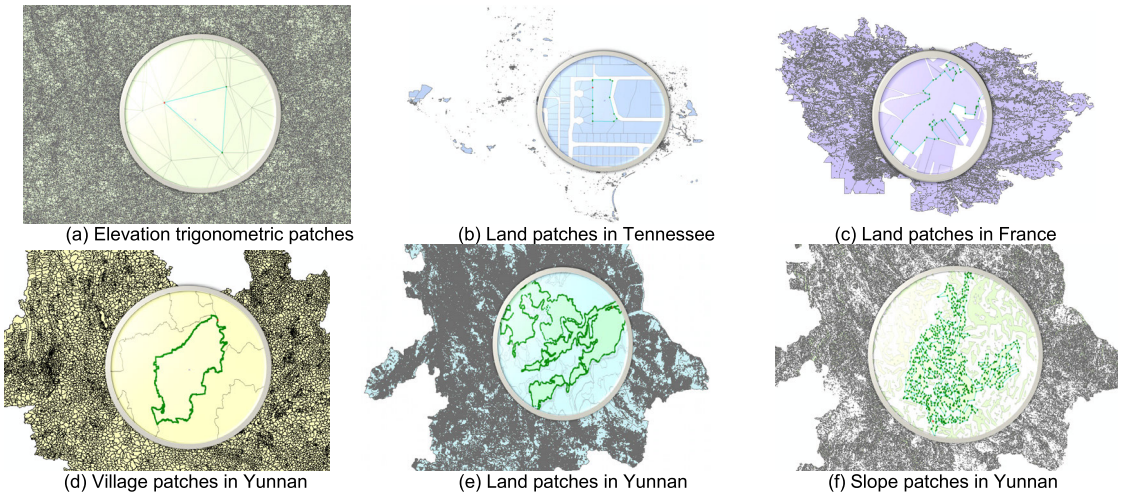

FIGURE 2. Experimental data.

TABLE 1. Basic information statistics of experimental data.

Datasets	Number of polygons	Number of vertices	Number of rings	Density of vertices (per square kilometer)	Data sources
Elevation trigonometric patches	486,389	1,945,556	0	0.2	ASTER GDEM
Tennessee land patches	324,564	3,479,160	4444	0.003	OpenStreetMap
Land patches in France	65,214	1,658,994	5371	0.09	OpenStreetMap
Village patches in Yunnan	20,317	37,989,776	0	0.06	National Census Geography of China
Land patches in Yunnan	454,702	79,275,039	65,679	9	The Third National Land Survey of China
Slope patches in Yunnan	108,025	892,878	14,671	0.08	ASTER GDEM

- Edge vibration frequency

Edge vibration frequency refers to the frequency of polygon vertices in the concave–convex variation [13], [44]. In this study, Brinkhoff's definition of edge vibration frequency is quoted.

Definition 2: If a polygon has P_v vertices and P_c concaves, then $P_c \leq P_v - 3$. The concave rate of the polygon can be normalized to $Notches_{norm} = \frac{P_c}{P_v - 3}$. The edge vibration frequency of the polygon $Freq_p$ can be expressed as follows: $Freq_p = 16(Notches_{norm} - 0.5)^4 - 8\,(Notches_{norm} - 0.5)^2 + 1$.

Brinkhoff's research showed that when $Notches_{norm}$ was close to 0.5, the vibration frequency of the edge was high, and the smoothness of the edge was low. The edge of the polygon was smooth when $Notches_{norm}$ was close to 0 or 1 [13].

- Coverage ratio of polygon area

The coverage ratio of the polygon area is used to express the ratio of its area to the area of the study area, which can indirectly reflect the size of the spatial distribution of the polygon. The coverage ratio of the polygon area is a meaningful indicator. When the coverage ratio of the polygon area is great, it will likely participate in the spatial analysis and calculation. For example, if the coverage ratio of the polygon area is large, the polygon may intersect with many polygons.

B. COMPLEXITY EVALUATION MODEL OF POLYGON SHAPE

1) SELECTION OF MODEL FACTORS

Since it is not clear how much effect a potential polygon feature has on the time cost of overlay analysis, we choose to use the correlation coefficient [25] between polygon features and the time cost of overlay analysis to measure the effect. This study selected five representative complex geographical graphics and one simple geographical graphic dataset to determine the value of the aforementioned potential factors (Fig. 2). Table 1 describes the data.

These data were obtained at different scales and with various complexities. Aside from convex polygons, concave polygons, simple triangles, and quadrilaterals, complex polygons with hole structures were also observed. The number of vertices of different polygons ranged from several to tens of thousands. These data were representative. Specifically, these data samples were sufficient, and the common geographical polygon shapes were covered.

TABLE 2. Correlation coefficient of potential factors.

	Vertices	Concaves	Parts	Concavity	DP	ANN	freq	RC	CT
Vertices	1.00								
Concaves	0.97	1.00							
Parts	0.56	0.67	1.00						
Concavity	0.21	0.21	0.09	1.00					
DP	0.12	0.10	−0.01	0.31	1.00				
ANN	−0.14	−0.15	−0.10	−0.52	−0.24	1.00			
freq	0.05	0.06	0.04	0.37	0.23	−0.31	1.00		
RC	0.37	0.38	0.26	0.11	0.01	−0.08	0.04	1.00	
CT	0.95	0.93	0.57	0.46	0.11	−0.11	0.13	0.46	1

We used the above-mentioned polygons as experimental data for overlay analysis and the clipping layer to clip the subject layer. We also recorded the time cost. The clipping layer had only one complex polygon. We randomly selected 100,000 polygons (if the dataset was less than 100,000, then the features were repeatedly extracted) from each of the datasets to merge into the subject layer, which eventually contained 600,000 polygons. The specific steps are presented as follows:

Step 1: The geometric center of the unique polygon in the clipping layer was computed.

Step 2: A polygon was taken from the subject layer sequence as the subject polygon, and the subject polygon was translated to the geometric center of the clipping polygon.

Step 3: The timer was started, and the subject polygon was clipped with the clipping polygon. Thereafter, the timer was stopped, and the time cost was recorded.

Step 4: Steps 2 and 3 were repeated until all polygons in the subject layer are clipped.

We used an ArcPy script to count the eight factors according to the factor calculation method. These factors are number of polygon vertices (vertices), number of concaves (concaves), number of ring structures (parts), concavity (concavity), density of polygon vertices (DP), coverage ratio of polygon area (RC), edge vibration frequency (freq), and spatial aggregation of polygon vertices (ANN). We used the data analytical tool in Excel software to analyze the correlation between the eight factors and the clipping cost–time (CT) and obtain the correlation coefficient table (Table 2).

A correlation coefficient is a numerical measure of some type of correlation, meaning a statistical relationship between two variables. The value of correlation coefficient in range from -1 to 1, where ± 1 indicates the strongest possible agreement and 0 the strongest possible disagreement [8]. In Table 2, the correlation coefficients between the factors and the CT were greater than 0.1, thereby indicating that these factors all affected the overlay analysis performance. The correlation coefficient between the number of concaves and vertices was greater than 0.7, thereby indicating a strong correlation between the number of concaves and vertices. The correlation among other factors was less than 0.4, except for the CT factor, that means such correlation among other factors was small. Therefore, concave and vertex factors cannot be simultaneously selected. After the analysis, this study used seven factors, namely, the number of polygon vertices (vertices), number of ring structures (parts), concavity (concavity), density of polygon vertices (DP), coverage ratio of polygon area (RC), edge vibration frequency (freq), and spatial aggregation of polygon vertices (ANN), as potential factors to create a model assessing the computational complexity on the basis of the polygon shape.

2) MODEL FOR EVALUATING POLYGON SHAPE COMPLEXITY

The polygon shape complexity C is expressed as follows:

$$C = a_i f\ (x_i) = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n \tag{1}$$

where x_i corresponds to the different shape feature factors, n is the number of shape feature factors used, and a_i is the weight of the effect of the shape factors on the complexity of shapes.

This study adopted stepwise regression analysis to determine the weight of the effect of each feature factor and fit the polygon shape complexity model. The abnormal values of the obtained data are eliminated and normalized. Subsequently, CT is taken as the dependent variable, and other factors were taken as the variables for stepwise regression analysis.

In the stepwise regression, the F value of the variables that were entered into the model was set to 0.05. If the F value was larger than 0.1, then the variable was dropped. The analytical results are presented in Tables 3, 4, and 5:

The adjusted *R-square* value in Table 3 is greater than 0.9, and the value of *sig.* in Table 4 is less than 0.05, thereby indicating that the model was statistically significant and well fitted. Table 5 shows that the vertex, parts, RC, and ANN were finally selected in the model. By contrast, DP and freq were dropped because the sig. values were greater than 0.05. Therefore, the shape complexity can be expressed as follows:

$$C = 0.692x_1 + 0.027x_2 - 0.023x_3 + 0.001x_4, \tag{2}$$

TABLE 3. Model summary.

Model	R	R-square	Adjusted R-square	Std. Error of the Estimate
1	.996[a]	.993	.993	.00165
2	.997[b]	.993	.993	.00160
3	.997[c]	.993	.993	.00159
4	.997[d]	.993	.993	.00159

[a]Predictors: (Constant), vertex
[b]Predictors: (Constant), vertex, and parts
[c]Predictors: (Constant), vertex, parts, and RC
[d]Predictors: (Constant), vertex, parts, RC, and ANN

TABLE 4. ANOVA[a].

Model		Sum of squares	df	Mean square	F	Sig.
1	Regression	9.167	1	9.167	3385874.830	.000[b]
	Residual	.067	24665	.000		
	Total	9.233	24666			
2	Regression	9.170	2	4.585	1788411.976	.000[c]
	Residual	.063	24664	.000		
	Total	9.233	24666			
3	Regression	9.171	3	3.057	1206914.153	.000[d]
	Residual	.062	24663	.000		
	Total	9.233	24666			
4	Regression	9.171	4	2.293	908381.279	.000[e]
	Residual	.062	24662	.000		
	Total	9.233	24666			

[a]Dependent variable: Time
[b]Predictors: (Constant) and vertex
[c]Predictors: (Constant), vertex, and parts
[d]Predictors: (Constant), vertex, parts, and RC
[e]Predictors: (Constant), vertex, parts, RC, and ANN

TABLE 5. COEFFICENTS[a].

Model		Unstandardized coefficients		Standardized coefficients	t	Sig
		B	Std. Error	Beta		
1	(Constant)	−8.990E−5	.000		−8.540	.000
	Vertex	.697	.000	.996	1840.075	.000
2	(Constant)	−9.730E−5	.000		−9.496	.000
	Vertex	.688	.000	.983	1541.319	.000
	Parts	.027	.001	.024	37.174	.000
3	(Constant)	−9.409E−5	.000		−9.238	.000
	Vertex	.691	.000	.988	1425.128	.000
	Parts	.027	.001	.024	37.324	.000
	RC	−.023	.001	−.010	−17.371	.000
4	(Constant)	.000	.000		−12.376	.000
	Vertex	.692	.000	.988	1422.900	.000
	Parts	.027	.001	.024	37.592	.000
	RC	−.023	.001	−.010	−17.215	.000
	ANN	.001	.000	.005	9.353	.000

where x_1 indicates the number of polygon vertices, x_2 is the number of polygon parts, x_3 is the coverage ratio of the polygon area, and x_4 is the spatial aggregation of polygon vertices. The coefficient of the polygon vertex number factor was the largest. Therefore, the number of polygon vertices had a great effect on the efficiency of overlay analysis.

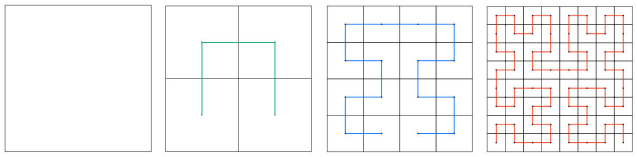

FIGURE 3. Hilbert cells and curve generation.

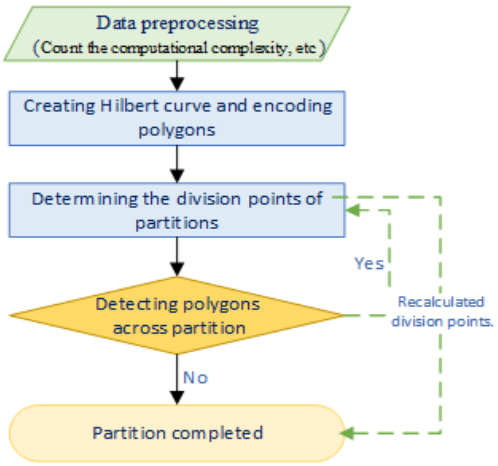

FIGURE 4. Process of data partitioning.

C. DATA DISTRIBUTION STRATEGY BASED ON THE SHAPE COMPLEXITY

1) PROCEDURE FOR DATA DISTRIBUTION STRATEGY

Given that the random and grid partition cannot reflect the spatial proximity of elements [28]–[45], this study proposed the Hilbert partition strategy with weighted shape complexity. The core idea was to calculate the shape complexity of each polygon on the basis of the evaluation model of the shape complexity. This study also takes the equal total shape complexity of each data partition as the measurement basis for the reasonable allocation of parallel computing tasks to ensure that the computing tasks of each data partition were simultaneously completed.

Fig. 3 shows the Hilbert partition. Hilbert partitioning divides a spatial region into $2^N \times 2^N$ cells. During the iteration, N is the order of the Hilbert curve (i.e., the number of iterations). In general, N is determined by the number of spatial objects, and the amount of spatial data (n) is required $n < 2^{2\times N}$.

The implementation of the Hilbert partition strategy with weighted shape complexity is detailed in Fig. 4.

Step 1: Data preprocessing

The average shape complexity of all polygons ($\bar{c}$) was calculated before partitioning. The sum of the shape complexity of all polygons was counted, and the ideal shape complexity of each partition (C_{ideal}) was obtained by dividing it by the number of partitions. C_{ideal} was taken as the partition basis.

Step 2: Creating Hilbert curve and encoding the polygons

The center points of the polygon MBR (minimum boundary rectangle) are normally distributed, and the optimal cell

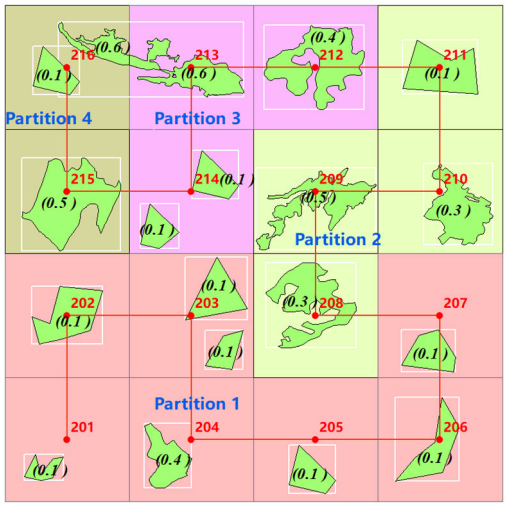

FIGURE 5. Example of allocating a group of complex polygons.

side length was determined. Thereafter, the value of N was obtained. A cell was allowed to contain multiple polygons, that is, repeated coding.

Step 3: Determining the division points of partitions.

We suppose that the shape complexity of the current partition was C_{cur}, and the complexity of the polygon to be distributed was c. When a polygon was added, $C_{cur} = C_{cur} + c$, and the values of C_{cur} and C_{ideal} were compared. When $C_{cur} \geq C_{ideal}$, the complexity of the current polygon c and the average complexity of the polygon $\overline{c}$ were compared. If $c < \overline{c}$, then the current polygon was taken as the end point of the current partition. If $c > \overline{c}$, then the current polygon was taken as the start point of the next partition.

Step 4: Detecting polygons across partitions

The MBR of the current polygon was used to determine whether the polygon was a cross partition. If the MBR crossed other partitions, then the object was simultaneously distributed to all the crossed partitions. At this time, the computational complexity of each partition was changed, so the division points needed to be calculated again.

2) EXAMPLE FOR DATA DISTRIBUTION STRATEGY

An example of allocating a group of complex polygons with our algorithm is given below (Fig. 5). We will allocate the polygons into four partitions with equal shape complexity.

The complexity indicators of each polygon, which have been marked on the polygons in Fig. 5, have been calculated in data preprocessing. We also counted the average complexity of all polygons ($\overline{c}$) and the ideal complexity of partitions (C_{ideal}).

First, the spatial region was divided into $2^N \times 2^N$ cells, and a Hilbert fill curve was generated. Fig. 5 shows that each cell was coded by Hilbert filling curve, and the codes range was from 201 to 216. Thereafter, we determined the cell where the center point of the polygon's MBR laid. The Hilbert code of a cell was the Hilbert code of the polygons that laid in the cell.

TABLE 6. Equipment configuration.

Equipment	Num	Hardware configuration	Operating system	Software
X86 Server	6	DELL R720, 24 core, 64 G RAM, hard disk drive	Centos7	Hadoop 2.7 and Spark 2.3.1

The MBR of the polygon is used to instead of it to simplify the calculation of the central point of a polygon.

We calculated the dividing points according to the method of step 3 in the previous section. Fig. 5 demonstrates that the dividing points are 207, 211, and 214. The number of polygons in the four partitions is not the same. However, the shape complexity of each partition is roughly the same.

In certain cases, polygons crossed multiple partitions, such as the polygon crossed partitions 3 and 4.cells We can judge whether the polygon crossed the partition 4 or not by comparing the MBR's coordinates of the polygon with the coordinates of cells in partition 4, thereby avoiding the operation of polygon intersection. If the polygon crossed partition 4, then it was also allocated to partition 4.

In parallel spatial analysis, ideal task allocation suggests that the shape complexity of each data partition is the same, and all data partitions simultaneously finish the computational task. According to the partition strategy in this study, the numbers of polygons in each partition were not necessarily the same. However, the shape complexity of each partition was basically the same; thus, the time–cost of each partition was basically the same. In comparison with the random partition and grid partition methods, this method considered the spatial proximity of polygons and the dimensionality reduction of coding.

IV. EXPERIMENTAL STUDY

In this section, the actual parallel overlay analysis was taken as an example. We compared the weighted Hilbert partition method for calculating shape computational complexity that was proposed in this study with the Hilbert partition method that uses vertex and polygon quantity balances.

A. EXPERIMENTAL ENVIRONMENT AND DATA

The data, experimental environment, and some algorithms in the previous paper [28] were used in this experiment to make the study comparative. The general process of spark overlay analysis was used to map spatial data to several RDDs, and each calculation thread performed overlay calculations and finally gathered the results. For more details, please refer to our previous paper [28]. The experimental environment was a combination of Hadoop and Spark and consisted of six Dell R720 servers (Table 6). The data were stored in HDFS in GeoJSON format. The size of the HDFS data block was 64 MB, and the number of file copies was 3.

In this work, the patches of the National Census Geography and those with a slope greater than 25 degrees in Yunnan

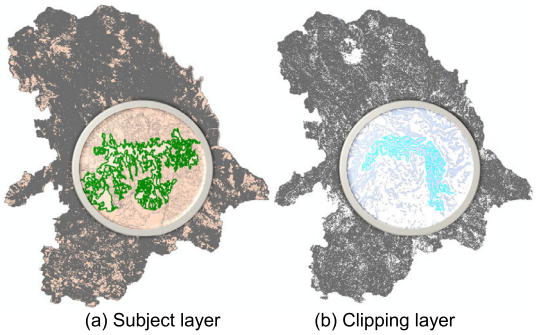

FIGURE 6. Experimental data.

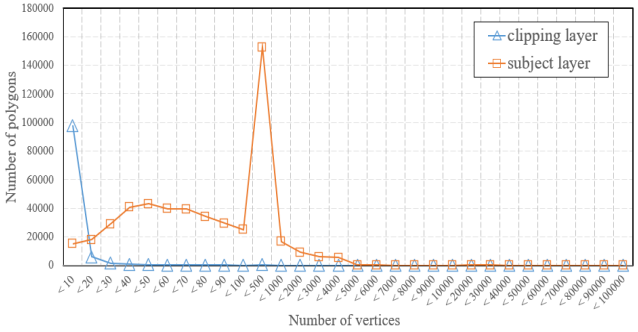

FIGURE 7. Statistical graphs of polygon distribution with different vertex numbers.

Province (generated by ASTER DEM data) were used as the subject and clipping layers for parallel overlay analysis, respectively (Fig. 6).

The statistics of the number of polygon vertices in the subject and clipping layers are illustrated in Fig. 7 Approximately 88,000,000 vertices were found in 500,000 patches of the subject layer through data checking. Among all polygons of the subject layer, the simplest polygon has four vertices, whereas the most complex polygon has 99,500 vertices. A total of 890,000 vertices were recorded in 10,800 patches of the clipping layer. Among all polygons of the clipping layer, the simplest one has eight vertices, whereas the most complex one has 5572 vertices. The subject and clipping layers are typical complex geographic data.

We created 10 subject layers by randomly thinning and duplicating the original 500,000 land-type patches. The numbers of patches in the subject layers were 50,000, 100,000, 250,000, 500,000, 1 million, 2 million, 4 million, 6 million, 8 million, and 10 million. Meanwhile, the number of patches in the clipping layer was 108,000.

B. EXPERIMENTAL PROCESS

1) EXPERIMENT 1: BALANCE COMPARISON OF THE DATA PARTITIONING STRATEGIES

The data skew was large when the difference between the maximum and the minimum values of the partition computational complexity was also large. We divided the data according to the number of polygons balanced through Hilbert partitioning (strategy 1), the number of vertices

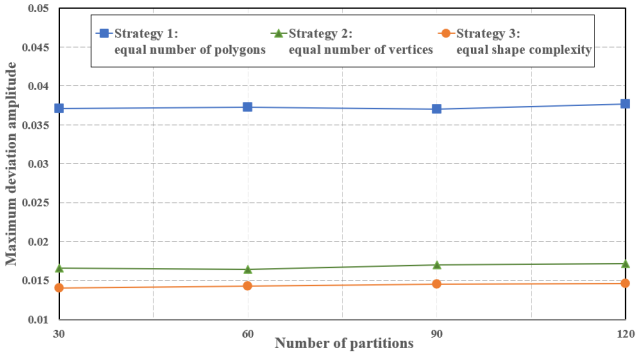

FIGURE 8. Comparison of the maximum deviation range of the three strategies.

balanced through Hilbert partitioning (strategy 2), and the shape computational complexity weighted through the Hilbert partitioning method (strategy 3) proposed in this paper. Strategy 1 is a method commonly used in other literature [31], [38]–[41]. Strategy 2 is the method of spatial computing task allocation in my previous research [28]. Strategy 3 is the method adopted in this study.

Given the different dimensions of the polygon and vertex numbers and computational complexity, we used the maximum deviation amplitude of the shape complexity to express the equilibrium degree of data allocation. This task was conducted to quantify the effect of the different allocation strategies.

Definition 3: Assuming that the data are divided into N partitions, and the maximum, minimum, and average values of an indicator in N partitions are V_{max}, V_{min}, and $\overline{V}$, respectively, the maximum deviation amplitude R_{md} of the indicator is expressed as $R_{md} = \frac{V_{max} - V_{min}}{\overline{V}}$. The equilibrium degree is high when the R_{md} value is small.

The data were divided into 30, 60, 90, and 120 partitions by using the three strategies. The deviation statistics of the computational complexity of each partition are shown in Fig. 8.

Although the numbers of polygons in each data partition in strategy 1 were the same, the maximum deviation amplitude of the shape complexity was the largest and more than twice those in strategies 2 and 3. When strategy 3 was adopted, the deviation degree was small, and the rationality of task allocation was the high under the same number of partitions. When strategy 2 was adopted, the deviation degree was slightly higher than that of strategy 3. When strategy 1 was adopted, the deviation degree was large, and the rationality of task allocation was low.

2) EXPERIMENT 2: PARALLEL OVERLAY ANALYSIS AND COMPARISON OF DIFFERENT COMPUTING TASK ALLOCATION STRATEGIES

In the experiment, the Greiner-Hormann algorithm and Spark framework were used to realize parallel overlay analysis. The Hilbert curve was used to partition the data, and the number of polygons, vertices, and computational complexity of each

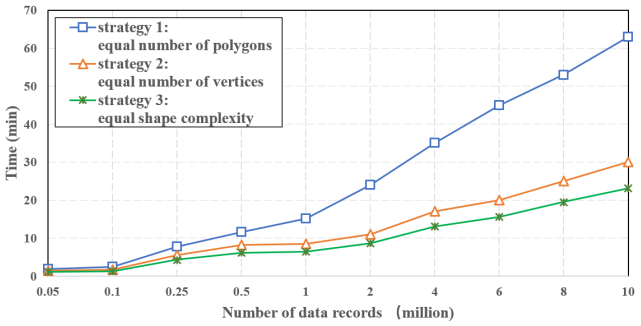

FIGURE 9. Time–cost statistical graphs of different data partitioning methods.

partition were equal to those of the strategy of computing partition points. In the three different data allocation strategies, we used sets of 50,000, 100,000, 250,000, 500,000, 1 million, 2 million, 4 million, 6 million, 8 million, and 10 million polygons for the parallel overlay calculation, and the time costs were recorded. The CT statistics of the overlay analysis under different data allocation strategies are shown in Fig. 9.

a) The three time–cost curves show an upward trend with the increase in data volume. The time–cost of strategy 1 is more than twice that of strategies 2 and 3 under the same number of computing tasks. The complexity difference in the geographic graphics had an important effect on the parallel nature of parallel spatial analysis.

b) The time–cost curve of strategy 3 was the lowest, thereby verifying the effectiveness of the complexity model proposed in this study.

c) The time–cost curve of strategy 2 was slightly higher than that of strategy 3, thereby indicating that the number of polygon vertices was an important factor affecting the performance of the overlay analysis.

C. ANALYSIS OF THE EXPERIMENTAL RESULTS

The three strategies are designed on the basis of the Hilbert filling curve. The time complexity of constructing the Hilbert curve is $O(N^2)$, where N is the order of the Hilbert curve. The parallel overlay analysis is based on the Greiner–Hormann algorithm, and its time complexity is O(logN).

The experimental results showed that the proposed method (strategy 3) had an optimal rationality of the spatial data allocation and short CT, thereby showing that the proposed shape computational complexity model reflected the effect of the polygon shape complexity on the overlay analysis efficiency.

Strategy 1 did not consider the difference in the shape complexity of each experimental data. Although the number of polygons in each partition was the same, the computing intensity of each partition was quite different. The result showed that the data partition with the lowest shape complexity finished first. Meanwhile, the data partition with the highest shape complexity finished last, with a long interval of end time between the two parallel computing operations. The rationality of spatial data allocation of strategy 2 in Experiment 1 and the time–cost of parallel computing of strategy 2 in Experiment 2 were close to those of strategy 3; this finding was also consistent with the shape complexity evaluation model, that the number of vertices was the largest effect factor in the shape complexity evaluation model. In reference [28], from the point of view of the time complexity of the software algorithm, the number of polygon vertices was proposed as the basis of data allocation, and other factors that affected the overlay analysis were ignored. From the point of view of the effect of the shape complexity on the performance of spatial analysis, this study introduced the shape features that affect the performance of spatial analysis and constructed a shape complexity evaluation model for overlay analysis by using a stepwise regression analysis. Here, the shape complexity evaluation model of the overlay analysis, which explained [28], [41] the effectiveness of using the number of polygon vertices as the data allocation factor, was proposed. Multiple factors and effect weights that affected the performance of overlay analysis were comprehensively provided. In combination with the Hilbert partition, a weighted Hilbert partition strategy for the shape complexity was designed in this study. Better effective allocation than that in the referenced citation [28] was achieved through the comparison of data allocation experiments.

V. CONCLUSION AND FUTURE RESEARCH

A. CONCLUSION

In this work, we analyzed the effect of different shape features on the efficiency of spatial analysis and selected the independent shape feature factors. We used parallel overlay analysis as an example and created the complexity model of shape computing for the Greiner–Hormann algorithm through stepwise regression analysis. Moreover, we designed a weighted Hilbert partition strategy for spatial data allocation in parallel Geocomputation. This method achieved effective spatial data allocation of parallel spatial computing.

The core idea of the method is to analyze and quantify the shape feature factors that affect the time–cost of the specified algorithm. Such an initiative is conducted to evaluate the differences of time–cost to calculate each spatial object and guide the allocation of spatial data according to the differences. Since the spatial proximity and dimensionality reduction of coding, Hilbert fill curve was used as the spatial partition method. Spatial algorithms are affected by the shape of spatial objects, and great differences can be observed in the shape of geographic data in reality; thus, the method proposed in this study is universal. The essence of this idea was to measure the time-consuming characteristics of the computing tasks and develop a fine-grained data allocation strategy. However, the shape features that affect the performance of different algorithms are different. The shape feature factors and weights selected by different algorithms are different. The computational complexity of shape will bring some additional time–cost. An effective method to reduce the calculation of the complexity model is to retain only the factor with the largest weight in the complexity model. When

the model is simplified to only one factor, the essence of the model becomes the time complexity in computer science.

B. FUTURE RESEARCH

In the next step, we will construct a set of complexity factor systems and shape complexity models for common geospatial analyses on the basis of the method for assessing the shape complexity proposed in this work to preprocess and categorize the shape complexity for the massive spatial data stored in the database. This step provides an efficient data allocation strategy for the parallel spatial analysis of massive geographic data.

ACKNOWLEDGMENT

The authors are grateful to L. Li for providing their experimental data and to L. Hou for imparting valuable suggestions.

REFERENCES

[1] L. Qingquan and L. Deren, "Big data GIS," *Geomatics Inf. Sci. Wuhan Univ.*, vol. 39, no. 6, pp. 641–644, 2014.
[2] C. Yang, M. Goodchild, Q. Huang, D. Nebert, R. Raskin, Y. Xu, M. Bambacus, and D. Fay, "Spatial cloud computing: How can the geospatial sciences use and help shape cloud computing?" *Int. J. Digit. Earth*, vol. 4, no. 4, pp. 305–329, Jul. 2011.
[3] Q. Huang, J. Li, and Z. Li, "A geospatial hybrid cloud platform based on multi-sourced computing and model resources for geosciences," *Int. J. Digit. Earth*, vol. 11, no. 12, pp. 1184–1204, Dec. 2018.
[4] C. Yang, P. Fu, M. F. Goodchild, and C. Xu, "Integrating GIScience application through Mashup," in *CyberGIS for Geospatial Discovery and Innovation*, S. Wang and M. F. Goodchild, Eds. Dordrecht, The Netherlands: Springer, 2019, pp. 87–112.
[5] Z. Li, "Geospatial big data handling with high performance computing: Current approaches and future directions," 2019, *arXiv:1907.12182*. [Online]. Available: http://arxiv.org/abs/1907.12182
[6] G. Greiner and K. Hormann, "Efficient clipping of arbitrary polygons," *ACM Trans. Graph.*, vol. 17, no. 2, pp. 71–83, Apr. 1998.
[7] D. Hilbert, "Ueber die stetige abbildung einer line auf ein Flächenstück," *Mathematische Annalen*, vol. 38, no. 3, pp. 459–460, 1891.
[8] J. Taylor, *Introduction to Error Analysis, the Study of Uncertainties in Physical Measurements*. New York, NY, USA: Univ. Science Books, 1997.
[9] Z. Gao, X. Yang, J. Gong, and H. Jin, "Research on image complexity description methods," *J. Image Graph.*, vol. 15, no. 1, pp. 129–135, 2010.
[10] Tilove, "Line/Polygon classification: A study of the complexity of geometric computation," *IEEE Comput. Graph. Appl.*, vol. 1, no. 2, pp. 75–88, Apr. 1981.
[11] C. Cheng, P. Shi, C. Song, and J. Gao, "Geographic big-data: A new opportunity for geography complexity study," *Acta Geographica Sinica*, vol. 73, no. 8, pp. 1397–1406, 2018.
[12] B. B. Mandelbrot, *The Fractal Geometry of Nature*. New York, NY, USA: WH Freeman, 1983.
[13] T. Brinkhoff, H.-P. Kriegel, R. Schneider, and A. Braun, "Measuring the complexity of polygonal objects," in *Proc. ACM-GIS*, 1995, p. 109.
[14] Y. Xia, R. Zhao, and Z. Jiang, "Fractal dimension estimation based on mathematical morphology," *J. image Graph.*, vol. 9, no. 6, pp. 760–766, 2018.
[15] N. Jin and S. Chen, "A new method of filtering edges of men-made objects from nature background," *J. Image Graph.*, vol. 5, no. 5, pp. 406–410, 2018.
[16] J. Reichert, A. R. Backes, P. Schubert, and T. Wilke, "The power of 3D fractal dimensions for comparative shape and structural complexity analyses of irregularly shaped organisms," *Methods Ecol. Evol.*, vol. 8, no. 12, pp. 1650–1658, Dec. 2017.
[17] D. Moser, H. G. Zechmeister, C. Plutzar, N. Sauberer, T. Wrbka, and G. Grabherr, "Landscape patch shape complexity as an effective measure for plant species richness in rural landscapes," *Landscape Ecol.*, vol. 17, no. 7, pp. 657–669, 2002.
[18] Y. Chen and H. Sundaram, "Estimating complexity of 2D shapes," in *Proc. IEEE 7th Workshop Multimedia Signal Process.*, Oct. 2005, pp. 1–4.
[19] H. Su, A. Bouridane, and D. Crookes, "Scale adaptive complexity measure of 2D shapes," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2006, pp. 134–137.
[20] W. Saleem, A. Belyaev, D. Wang, and H.-P. Seidel, "On visual complexity of 3D shapes," *Comput. Graph.*, vol. 35, no. 3, pp. 580–585, Jun. 2011.
[21] T. Matsumoto, K. Sato, Y. Matsuoka, and T. Kato, "Quantification of 'complexity' in curved surface shape using total absolute curvature," *Comput. Graph.*, vol. 78, pp. 108–115, Feb. 2019.
[22] X. Duan, X. Deng, and Q. Zuo, "The research on the complexity of progressive die edge," *J. Graph.*, vol. 27, no. 5, pp. 94–97, 2006.
[23] B. Chazelle and J. Incerpi, "Triangulation and shape-complexity," *ACM Trans. Graph.*, vol. 3, no. 2, pp. 135–152, Apr. 1984.
[24] M. Guo, Q. Guan, Z. Xie, L. Wu, X. Luo, and Y. Huang, "A spatially adaptive decomposition approach for parallel vector data visualization of polylines and polygons," *Int. J. Geograph. Inf. Sci.*, vol. 29, no. 8, pp. 1419–1440, Aug. 2015.
[25] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise Reduction in Speech Processing*. Berlin, Germany: Springer, 2009, pp. 1–4.
[26] F. Ying, P. Mooney, P. Corcoran, and A. C. Winstanley, "A model for progressive transmission of spatial data based on shape complexity," *Sigspatial Special*, vol. 2, no. 3, pp. 25–30, Nov. 2010.
[27] A. Li, Y. Chen, M. Yao, and S. Wu, "Quantitative measurement of geometrical information for sensitive features in secret-related vector digital maps," *J. Geo-Inf. Sci.*, vol. 20, no. 1, pp. 7–16, 2018.
[28] Zhao, Jin, Fan, Song, Zhou, and Jiang, "High-performance overlay analysis of massive geographic polygons that considers shape complexity in a cloud environment," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 7, p. 290, Jun. 2019. [Online]. Available: https://www.mdpi.com/2220-9964/8/7/290
[29] L. Zhao, L. Chen, R. Ranjan, K.-K.-R. Choo, and J. He, "Geographical information system parallelization for spatial big data processing: A review," *Cluster Comput.*, vol. 19, no. 1, pp. 139–152, Mar. 2016.
[30] A. Eldawy and M. F. Mokbel, "SpatialHadoop: A MapReduce framework for spatial data," in *Proc. IEEE 31st Int. Conf. Data Eng.*, Apr. 2015, pp. 1352–1363.
[31] A. Aji, F. Wang, H. Vo, R. Lee, Q. Liu, X. Zhang, and J. Saltz, "Hadoop GIS: A high performance spatial data warehousing system over mapreduce," *Proc. VLDB Endowment*, vol. 6, no. 11, pp. 1009–1020, Aug. 2013.
[32] S. Nishimura, S. Das, D. Agrawal, and A. E. Abbadi, "MD-HBase: A scalable multi-dimensional data infrastructure for location aware services," in *Proc. IEEE 12th Int. Conf. Mobile Data Manage.*, Jun. 2011, pp. 7–16.
[33] A. Eldawy, L. Alarabi, and M. F. Mokbel, "Spatial partitioning techniques in SpatialHadoop," *Proc. VLDB Endowment*, vol. 8, no. 12, pp. 1602–1605, Aug. 2015.
[34] A. Eldawy and M. F. Mokbel, "A demonstration of SpatialHadoop: An efficient mapreduce framework for spatial data," *Proc. VLDB Endowment*, vol. 6, no. 12, pp. 1230–1233, Aug. 2013.
[35] J. Yu, Z. Zhang, and M. Sarwat, "Spatial data management in apache spark: The GeoSpark perspective and beyond," *GeoInformatica*, vol. 23, no. 1, pp. 37–78, Jan. 2019.
[36] M. Tang, Y. Yu, Q. M. Malluhi, M. Ouzzani, and W. G. Aref, "Location-Spark: A distributed in-memory data management system for big spatial data," *Proc. VLDB Endowment*, vol. 9, no. 13, pp. 1565–1568, Sep. 2016.
[37] J. Yu, J. Wu, and M. Sarwat, "GeoSpark: A cluster computing framework for processing large-scale spatial data," in *Proc. 23rd SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst. GIS*, 2015, pp. 1–4.
[38] J. Yu, J. Wu, and M. Sarwat, "A demonstration of GeoSpark: A cluster computing framework for processing big spatial data," in *Proc. IEEE 32nd Int. Conf. Data Eng. (ICDE)*, May 2016, pp. 1410–1413.
[39] Y. Wang, Z. Liu, H. Liao, and C. Li, "Improving the performance of GIS polygon overlay computation with MapReduce for spatial big data processing," *Cluster Comput.*, vol. 18, no. 2, pp. 507–516, Jun. 2015.
[40] Z. Zheng, C. Luo, W. Ye, and J. Ning, "Spark-based iterative spatial overlay analysis method," in *Proc. Int. Conf. Electron. Ind. Automat. (EIA)*. Atlantis Press, 2017, pp. 227–232.
[41] J. Fan, T. Ma, M. Ji, Y. Zhou, and T. Xu, "Implementation and optimization of eight parallel polygon overlapping tools with OpenMP at the feature layer level in GIS," *Prog. Geography*, vol. 32, no. 12, pp. 1835–1844, 2013.
[42] A. S. Fotheringham, C. Brunsdon, and M. Charlton, *Quantitative Geography: Perspectives on Spatial Data Analysis*. London, U.K.: Sage, 2000.
[43] R. liu, "A simple and fast algorithm for detecting the convexity and concavity of vertices for an arbitrary polygon," *J. Softw.*, vol. 13, no. 7, pp. 1309–1312, 2002.

[44] M. Cheng, Q. Sun, L. Xu, and H. Chen, "Polygon contour similarity and complexity measurement and application in simplification," *J. Geodesy Geoinf. Sci.*, vol. 48, no. 4, pp. 91–103, 2019.
[45] S. Wang, E. Zhong, H. Lu, H. Guo, and L. Long, "An effective algorithm for lines and polygons overlay analysis using uniform spatial grid indexing," in *Proc. 2nd IEEE Int. Conf. Spatial Data Mining Geograph. Knowl. Services (ICSDM)*, Jul. 2015, pp. 175–179.

KANG ZHAO is currently pursuing the Ph.D. degree with the State Key Laboratory of Information Engineering Surveying, Mapping, and Remote Sensing, Wuhan University. Previously, he worked as the Director of the Information Development Department, Yunnan Provincial Geomatics Centre, China. He served with the Yunnan geological environment big data management and analysis platform. He is a member of the Chinese Society of Geodesy, Photogrammetry, and Cartography. He has authored several articles, and his research interests include geographic computing, geographic information systems, and spatiotemporal big data analytics.

BAOXUAN JIN received the Ph.D. degree from the Key Laboratory of Surveying and Mapping Remote Sensing Information Engineering, Wuhan University, China. He is currently the Chief Engineer with the Department of Natural Resources, Yunnan, China. He has overseen the construction of natural resource cloud projects and other major projects in Yunnan Province. He is a member of the China Land Science Society. He has authored over 20 articles. His research interests include spatiotemporal big data analysis, land use planning, cloud computing, and geographic information systems.

HONG FAN was born in Hunan, China. She received the B.S. and M.S. degrees in computer science in 1988 and 1991, respectively, and the Ph.D. degree in GIS and remote sensing from Wuhan University, China, in 2001. She is currently a Professor and a Ph.D. Supervisor with Wuhan University. She is also a geographic information system expert. She has been involved in teaching and research in cartography and geographic information systems for a long time. Her research interests include high-performance geographical computation, smart geographic information service, qualitative geographic information retrieval, spatiotemporal big data mining, and natural language spatial information processing.

MEI YANG is currently pursuing the Ph.D. degree with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University. Her research interests include the prediction and retrieval of air pollution levels, human dynamic exposure to contamination, and spatiotemporal big data mining.