

Article

A Space-Interconnection Algorithm for Satellite Constellation Based on Spatial Grid Model

Shuang Li ¹ , Kaihua Hou ², Chengqi Cheng ², Shizhong Li ³ and Bo Chen ^{2,4,*}

¹ Institute of Chinese Historical Geography, Fudan University, Shanghai 200043, China; li_shuang@fudan.edu.cn

² College of Engineering, Peking University, Beijing 100871, China; houkaihua@pku.edu.cn (K.H.); ccq@pku.edu.cn (C.C.)

³ TH Center of China, Beijing 100094, China; xalsz@163.com

⁴ School of Aeronautics and Astronautics, SunYat-Sen University, Guangzhou 510275, China

* Correspondence: pkuchenbo@pku.edu.cn; Tel.: +86-010-6275-5390

Received: 5 June 2020; Accepted: 1 July 2020; Published: 2 July 2020



Abstract: With the rapid development of large-scale satellite constellations and the increasing demand for rapid communication and emergency rescue using global satellite-based Internet, there have been new requirements for efficient algorithms for inter-communication between satellites. As the constellations of low-orbit satellites become larger, the complexities of real-time inter-satellite calculation and path planning are becoming more complicated and are increasing geometrically. To address the bottlenecks in large-scale node space computing, we introduced a global space grid. Based on this grid, an efficient calculation method of spatial inter-connection between satellite constellations is proposed, according to the concept of “storage for computing” and the high computational efficiency of the spatial grid model. This strategy includes the following parts: (1) the introduction of the GeoSOT-3D global grid model into aerospace and the construction of the aerospace grid indexing BigTable; (2) a set of algorithms for satellite visibility analysis according to the visible grid look-up table and the secondary grid index; and (3) planning inter-satellite routing by querying the grid’s inherent visibility. The idea at the basis of this method is to employ the “space for time” concept to convert the high-dimensional floating operations into one-dimensional matching operations by querying the inherent “visible” attribute of the grid. In our study, we simulated thousands of satellites, discretized their trajectories into grids, and pre-calculated the visibility between grid cells to plan the routing path for the ground data transmission. The theoretical analysis and experimental verification show that the algorithm is feasible and efficient, and it can significantly improve the computational efficiency of inter-satellite connection. We hope that the method can be used in emergency communications, disaster warning, and maritime rescue, and can contribute to the next generation of satellite internet and “satellite-ground” integrated networks.

Keywords: GeoSOT-3D; BigTable; satellite communication; satellite visibility; route planning

1. Introduction

With the rapid development of satellite communication technology, satellite networks have become an important part of the next generation of global communication systems as an important supplement to the ground network [1–3]. Today, many countries consider the construction of satellite Internet as part of their national strategies [4–6]. China has clearly included satellite Internet in the category of “new infrastructure” and positioned it as a space-based communications network infrastructure that complements global high-speed coverage, blind spot coverage in urban hotspots, outdoor applications, and high-mobility applications. Commercial institutions, governments, and the

military around the world have a strong demand for low-orbit satellite constellations. A series of plans with broad market prospects have therefore been developed. The plans are mainly based on low-orbit micro-satellite constellations, with great advantages in terms of functional density, development cost, launch difficulty, etc. [7–9]. These constellations can effectively solve the practical problem of networking in remote areas, such as oceans and deserts, and they have irreplaceable advantages in emergency communications, disaster warnings, and maritime rescue [10]. However, satellite networking includes not only communication satellites, but also navigation and remote sensing satellites (Table 1). The United States, the European Union, Russia, and China have already launched the positioning, navigation, and timing (PNT) satellite constellation systems, and established low-orbit satellite constellations such as SpaceX and OneWeb [6,9]. The existing PNT system needs to be improved in the areas of real-time observation of satellites, interconnection with communications, and navigation satellites. Considering the above issues and promoting the further integration of remote sensing satellites and PNT systems, the concept of PNTRC (positioning, navigation, timing, remote sensing, and communication) has been proposed [11]. The PNTRC is a real-time service system for space-based information, integrating communication, navigation, and remote sensing, and its primary goal is the interconnection of various space platforms. The increasing number of satellites in a constellation will cause the complexity of algorithms to increase geometrically. Therefore, it is of great significance, both theoretically and for the development of applications, to study the most efficient method of inter-connecting heterogeneous satellite constellations.

Table 1. Plans of Remote Sensing Satellite Constellations [7,9,12].

Plan of RS Satellite	Number of Satellites	Spatial Resolution	Time Resolution	Satellite Payload	Main Technical Characteristics
Planet-Dove	140+	3–5 m	/	Multi-spectral	3D observation, global coverage
SkySat	24 (14)	0.7–1 m	6–8 h	Video/Multi-spectral	The first sub-meter optical satellite
BlackSky	60	1 m	10–60 min	Video/Multi-spectral	Multi-orbit constellation
EarthNow	500	Unknown	/	Unknown	Provide images and videos anywhere on the earth

In the process of satellite constellation networking, the first step is to determine whether the satellites are visible before establishing a network link between the satellites. Traditional analysis algorithms are based on research on satellites that meet the visibility conditions in a short time and under accuracy constraints [13,14]. This requires multiple floating-point operations involving the orbital parameters in order to solve the orbital equations, which is inefficient and requires computing resources [4,11]. Therefore, researchers have conducted several studies on how to improve the efficiency of the inter-satellite visibility analysis algorithm. In addition, most algorithms are optimized for specific constellations, lacking a general optimization algorithm [15,16]. Discussion on visibility analysis has also taken place within the industry. The most widely used analysis method is the System Tool Kit (STK, of Analytical Graphics Inc.) software. The STK simulates data transmission between two objects, supports calculation based on access between satellites, can dynamically display the visibility between satellites in a map window, and can generate a visibility report [17].

After calculating whether the satellites are in sight, communication and data transmission between the visible satellites is one of the most important ways for the ground observation station and users to obtain target observation data that are outside of their receiving range. The biggest challenge in this process is that the satellites move very quickly, and their network topology (the topological structure of network which may be depicted physically or logically) is much more complex compared to the topologies of mid-orbit and high-orbit satellites [18]. In addition, the fast movement between the satellites will cause a Doppler frequency shift, which considerably affects communication quality. This is one reason for the importance of building a high-quality “space interconnection link” [19,20]. Therefore, choosing the appropriate routing path has become the key to effective data transmission from the satellite to the ground. Summarizing the existing research work, satellite routing algorithms can be divided into a centralized routing algorithm (CRA) and distributed routing algorithm (DRA) according to routing decisions [21–23]. In CRA, the satellite network control center periodically collects

the status of each link, calculates and plans the routing path, and sends the results to the nodes in the network topology model [24]. By dividing the time between status collection events into several intervals, the topology within each time interval is considered to be static, and the stored routing table and the path planning algorithm, such as the Dijkstra algorithm, are then used for route planning [25]. However, routing algorithms based on the virtual topology algorithm will use considerable storage resources to contain the topology, and when the satellite node fails, the algorithm requires a large amount of calculation and thus has poor adaptability. DRA is a connectionless routing algorithm that does not consider global nodes, and it reduces the need for large on-board storage. The main steps of DRA include: (1) dividing the earth surface into several areas, such as spatial grids, (2) constructing a virtual network node at the center of each area; and (3) selecting the next step of the node by calculating the neighboring spatial area [23]. This algorithm requires nodes to have high processing ability, but it may not be able to obtain the best path [26]. Currently, low-orbit satellites that do not build inter-satellite links have become a new source for real-time observation and communication with the ground station [4,27]. For low-orbit satellite constellations that do not have inter-satellite links, there is an urgent need to optimize more efficient inter-satellite calculation methods [4,21].

The real-time calculation of the relative position between satellites is therefore required both in the satellite visibility and path planning algorithms, especially in large constellation networks, such as Star Link and OneWeb (including hundreds or even thousands of plans). It is necessary to develop a more efficient satellite inter-communication method for the real-time computing needs of large constellations. The grid-code-calculating method based on the spatial grid model is an efficient position calculation system that extends the range of the global spatial grid to the aerospace region [28–30]. More importantly, by dividing the spatial range into discrete grids, visibility between the grids will turn into an inherent property, which can improve the calculation efficiency. In our study, GeoSOT-3D was selected as the discrete grid framework and extended to the aerospace region [31]. A general method for inter-satellite grid computing is then proposed.

The remainder of the paper is organized as follows: in Section 2, the grid inter-connection calculation method for satellite constellations based on GeoSOT-3D is proposed, and the key algorithms including satellite grid visibility and grid network topology are presented. Section 3 presents the results and analysis. The advantages and disadvantages of the work are discussed in Section 4. Finally, Section 5 concludes the paper.

2. Methods

This study proposes a grid interconnection calculation method for satellite constellations based on the GeoSOT-3D grid model. Previous studies have shown that GeoSOT-3D has good results in the fields of multi-source spatial data integration and efficient spatial database indexing, and is particularly helpful for improving the efficiency of spatial calculation [32–34]. A general grid-based calculation method for inter-satellite is proposed, taking the need for the user to obtain remote sensing satellite data in the future as an application scenario. The technical method includes: (1) GeoSOT-3D BigTable, (2) grid modeling, (3) satellite visibility calculation, and (4) satellite network routing planning.

The main steps comprise: (1) divide the entire aerospace space into discrete grids of different scales, identify each grid cell uniquely, and then create an aerospace grid indexing BigTable with the grid code as the primary key, serving as the link to spatial information (the concept of BigTable comes from Google BigTable, and in this article, it refers to the organization and storage architecture that uses grid coding as “Row Key” and associates spatial information) [35]. In “satellite-to-satellite” and “satellite-to-ground user” transmission, the satellite trajectories and satellite resources are mapped to the grid to unify the organization and calculation of trajectories and resources, respectively. (2) Construct a secondary index mechanism, and replace the traditional and complicated visibility calculation method by querying the BigTable and GeoSOT-3D secondary spatial index. (3) Establish a grid virtual node, create the virtual topology, and then plan an inter-satellite transmission path that meets the coverage

requirements by querying the visibility in the aerospace grid indexing BigTable. The research process is shown in Figure 1.

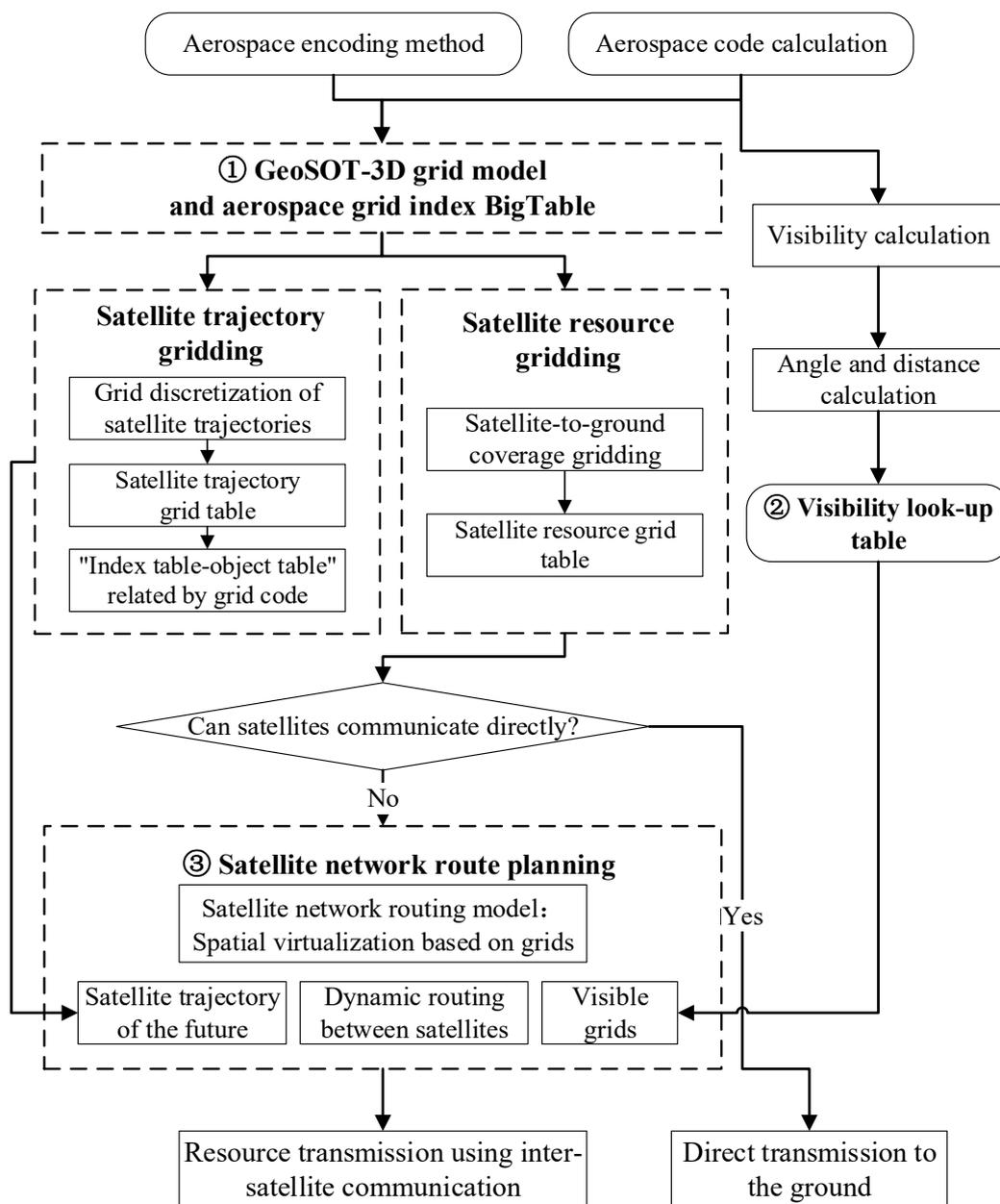


Figure 1. Research process.

2.1. GeoSOT-3D and Aerospace Grid BigTable

2.1.1. GeoSOT-3D Grid Model and Coding Method

The geographic coordinate subdivision grid with one-dimensional integer coding on a 2^n -tree (GeoSOT) model is a subdivision and coding method that subdivides the Earth’s surface into grids [31]. The subdivision method is as follows: via a three-time virtual expansion of the earth (the real space on the earth surface is extended to $512^\circ \times 512^\circ$, 1° extended to $64'$, $1'$ extended to $64''$), the global subdivision model can maintain grid code integral degrees, minutes and seconds, when a quad-tree subdivides the latitude and longitude coordinate space [32]. Finally, a multi-scale grid system is formed up to the global level (level 0) and down to the centimeter level (level 32) [33].

The GeoSOT-3D model expands the height dimension of the GeoSOT grid model, and divides the three dimensions of longitude, latitude, and elevation continuously through octree division, forming a multilevel 3-D grid system [34]. Finally, we obtain a multilevel grid model of integer degrees, integer minutes, and integer seconds, and form an earth grid 3-D model with a space of 50,000 km up to the outer periphery of the earth and down to the center of the earth, with 32 levels. The grids of the same level have no gaps or overlaps, and the grids of adjacent levels are nested within each other. Each grid is then assigned a code, which is used to uniquely identify the spatial area. Codes of grids at the same level are not random, but are sorted according to the Z-order space-filling curve (a spatial indexing method), which will result in neighboring grids having codes with close values [36]. Eventually, the high-dimensional spatial information is converted into a one-dimensional aerospace coordinate code, which is convenient for the identification, storage, and calculation of spatial information (Figure 2).

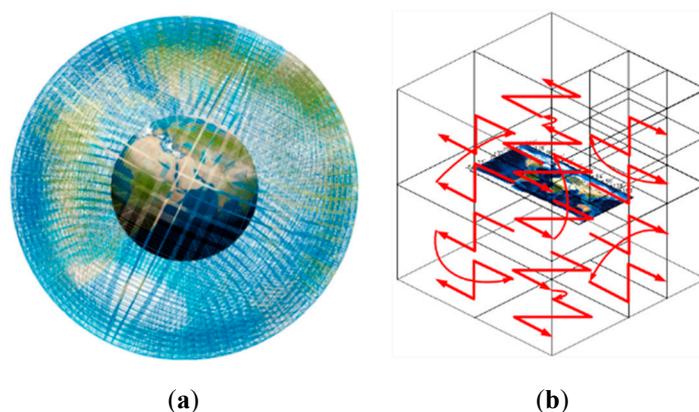


Figure 2. GeoSOT-3D model: (a) Global grid model; (b) Z-order filling curve.

2.1.2. Aerospace Grid Indexing BigTable

The grid coordinate code is globally unique and contains spatial information. It can be associated with the spatial information of aerospace and multi-sourced data (Figure 3a). Subsequently, the one-dimensional grid code is used as the primary key in the aerospace grid indexing BigTable, and the k-dimensional search is then converted to a one-dimensional search to reduce the complexity of the algorithm. Considering that the code is continuous, organizing spatial data according to the coding order can ensure efficient access to its spatial neighborhood. Generally speaking, spatial data that are adjacent in space are more likely to be accessed simultaneously. The structure of the aerospace grid indexing BigTable is summarized in Table 2.

Table 2. Structure of Aerospace grid indexing BigTable.

Id	Name	Type	Description
1	id	STRING/INT	GeoSOT-3D grid code
2	visible	STRING/INT	Visible grid codes
3	DataExist	INT	Whether there are data in the grid (true or false)
4	Data	JSON	Data description
	Time	TIME	Time, including the time to enter the grid, the time to leave the grid
	DataSource	JSON	Data source description
	Metadata	JSON	Metadata information

In the designed BigTable (Table 2), id is an integer or string type, supporting 2-D or 3-D grid code and various spatial operations based on code algebra. Visible is stored in an array format, that is, the visible grids of *id* can be stored as an array here.

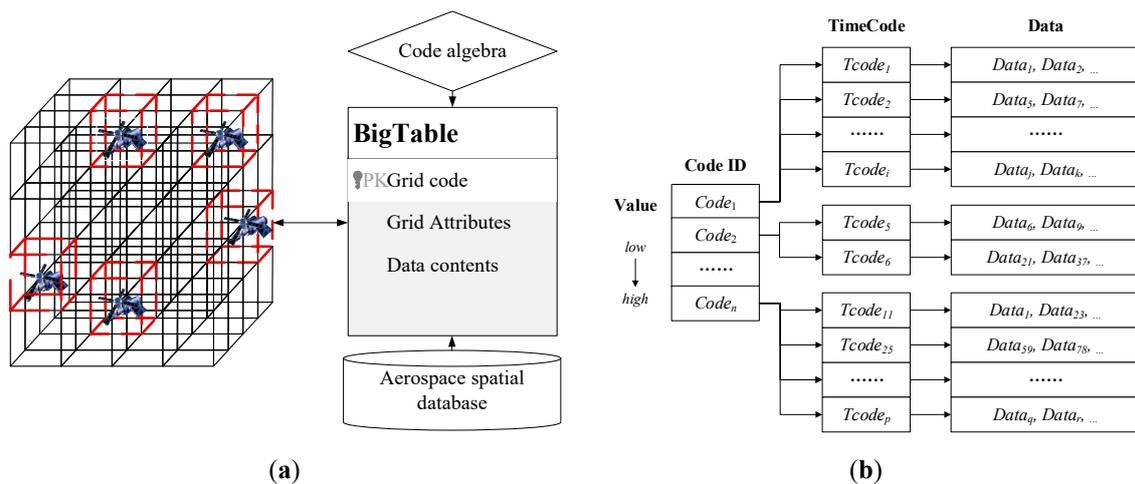


Figure 3. Aerospace storing model and indexing method: (a) aerospace storage model; (b) spatio-temporal secondary indexing method.

The most important aspect of BigTable is to build an index implementation mechanism based on grid code to improve the query efficiency in applications. In our study, a spatio-temporal secondary indexing model is established; that is, the spatial code is first sorted and then the time code in each grid is sorted (Figure 3b).

In the indexing model (Figure 3b), $Code_i$ is the i th grid code in the aerospace, $Tcode_i$ is the i th time code (using the coding method in [37]), and $Data_i$ is the spatial data or entity.

2.2. Satellite Grid Visibility Algorithm

The basic requirement for communication between satellites is that they are visible to each other. The main factors influencing satellite visibility include geometric visibility, antenna visibility, and the influence of distance between satellites [14,15]. Geometric visibility means that the two satellites are not blocked by the earth and the atmosphere, whereas antenna visibility implies that the angle range of the two satellites is within the angle range of each other’s antenna. The influence of distance refers to satellite communication being affected by power and other aerospace factors, resulting in the signal not being transmitted.

2.2.1. Traditional Satellite Visibility Analysis

Owing to the rapid movement of satellites, the prerequisite for the discussion in this section is to consider the satellite to be stationary at a time t . At time t , the distance between satellites A and B is l_{AB} , the orbital height is d_A and d_B (from the center of the earth), the radius of the earth and the thickness of the atmosphere are R_{earth} and h_{atm} , respectively, and the angle θ between the inter-satellite link and the satellite-earth link (the center of the earth) should satisfy the following Equation (1) [16]:

$$\begin{cases} \theta_A = \arccos\left[\frac{(d_A^2 + l_{AB}^2 - d_B^2)}{2d_A l_{AB}}\right] \\ \theta_B = \arccos\left[\frac{(d_B^2 + l_{AB}^2 - d_A^2)}{2d_B l_{AB}}\right] \end{cases} \quad (1)$$

When the link between the two satellites l_{AB} is tangential to the atmosphere, the two satellites are geometrically visible to each other, and this is a critical condition for calculating geometric visibility. Therefore, to judge whether the two satellites are visible, θ should be satisfied (2):

$$\begin{cases} \theta_A > \arcsin[(R_{earth} + h_{atm})/d_A] \\ \theta_B > \arcsin[(R_{earth} + h_{atm})/d_B] \end{cases} \quad (2)$$

When the distance between the two satellites is used to describe the visibility between the satellites, the following Equation (3) should be satisfied:

$$\begin{cases} l_{AB} < L_{AB} = L_{Amax} + L_{Bmax} \\ L_{Amax} = \sqrt{d_A^2 - (R_{earth} + h_{atm})^2} \\ L_{Bmax} = \sqrt{d_B^2 - (R_{earth} + h_{atm})^2} \end{cases} \quad (3)$$

where L_{AB} is the distance between satellites A and B under critical visible conditions; L_{Amax} is the distance from the tangent point to satellite A when the communication link between the two satellites is tangent to the atmosphere; L_{Bmax} is the distance between the tangent point and satellite B (Figure 4).

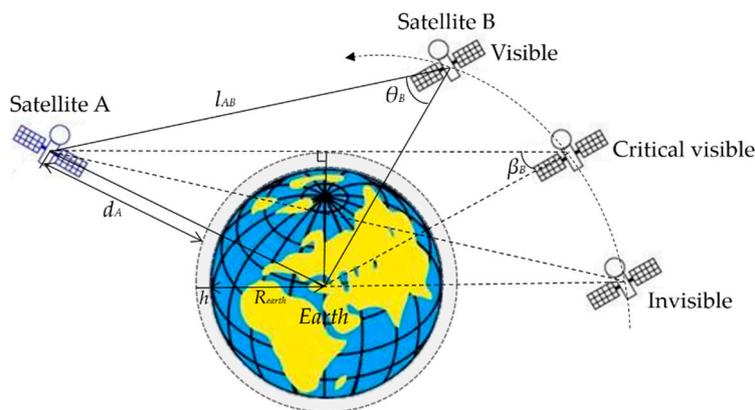


Figure 4. Geometric visibility between satellites.

2.2.2. Grid Visibility Analysis

In this section, we construct a grid visibility analysis method based on the aerospace grid indexing BigTable, including the satellite visibility grid lookup table and grid calculation method.

(i) Satellite visibility grid lookup table

The structure definition of the aerospace grid indexing BigTable is presented in Section 2.1.2, which contains the column “visible”. The type of this column is an array, which can store one or more codes of visible grids. Here we will provide samples of two grids in BigTable, one with data and the other without data. The column “data” can add or remove attributes.

As can be seen in Figure 5, the grid code “visible” is an inherent property of the grid. Whether the grid has data or not, there will be a corresponding “visible” grid (or set). When querying the “visible” grid or set of a grid, the corresponding query statement is entered. After creating a satellite visibility grid look-up table, querying the “visible” field shows that a grid may have many geometrically visible grids. The ground station does not need to consider the storage cost and can store all grids. However, the storage resources on the satellite are limited; hence, it is necessary to optimize the satellite visibility grid look-up table. Because satellite movement has periodicity, we can store all the visibility grids of a certain satellite at different times on the satellite when performing storage optimization, and we can match the geometric visibility grids according to the periodic running time. If the impact of satellite orbit perturbation has to be considered, we can periodically go to the background to calculate and update the view grid.

(ii) Grid Visibility Calculation

This section will then discuss how to calculate the visibility of the two grids. The principle of grid geometric visibility calculation is to calculate the grid code set of links between satellites, and then

judge whether the height code is blocked by the earth or the atmosphere, in order to realize the geometric visibility calculation between satellites.

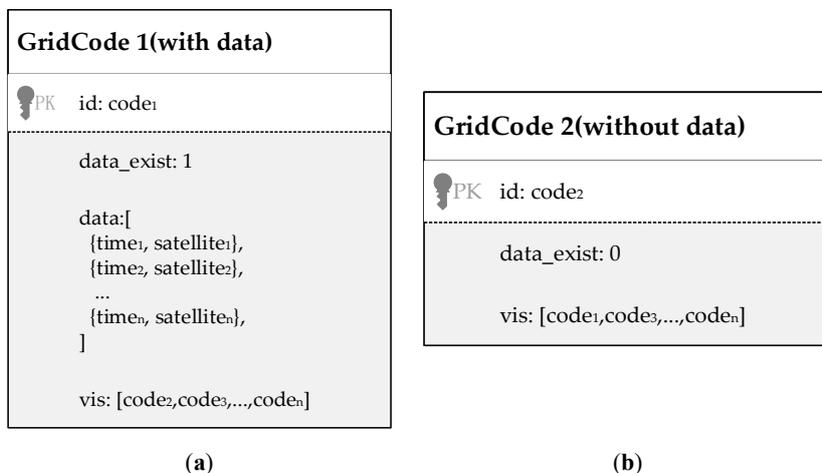


Figure 5. Documents of grid codes: (a) with data; (b) without data.

Step 1: Calculate the inter-satellite grid code set. Based on the 3D Bresenham algorithm, the path between two target grids is filled with grids with adjacent edges or corners [38].

Step 2: Compare height codes. Compare the minimum elevation code in the grid code set with the height code for the occlusion height (Equation (4)):

$$\min (H_{codes}) < H_{code (Rearth+hatm)} \tag{4}$$

where H_{codes} is the inter-satellite grid code set, and $H_{code (Rearth+hatm)}$ is the height code of the Earth and atmosphere.

The advantage of the algorithm is that it does not need to determine whether satellites are in the same orbit (calculations of different orbits are complicated), and the high-dimensional floating operations are converted into comparisons between one-dimensional height codes. After all visible grids of the grid are calculated in advance, they can be stored in the grid visibility look-up table and become an inherent property of the grid. When one satellite moves into the visible grid of the grid in which the other satellite is located, the two satellites are geometrically visible.

In addition, based on geometric visibility analysis, combined with the code azimuth calculation and distance measurement algorithms, we can further analyze the antenna visibility and impact of distance, and then achieve a more accurate visibility calculation between satellites.

2.3. Grid Network Topology Model and Routing Algorithm

A low-orbit satellite constellation is composed of hundreds of satellites. If only a time-sliced routing algorithm is used in the system, this centralized routing algorithm will result in great pressure on the storage and calculation abilities of the satellite. At the same time, in the event of an emergency, the efficiency and reliability are not high enough. For situations such as emergency communications, disaster warning, and maritime rescue, a set of low-cost and high-efficiency on-board route calculation methods is urgently needed to achieve efficient communication [9]. Therefore, it is necessary to deeply consider the characteristics of low-orbit satellite constellations and design a set of models and routing algorithms to improve the efficiency of satellite network topology. This new proposed method for network modeling and routing is clever and innovative, by mapping the complex dynamic topological relationships between satellites into a series of static grid topological models.

2.3.1. Grid Network Topology Model

First, a basic model of space virtualization routing is established for low-orbit satellite constellation. The main idea is to divide the surface of the earth into different areas, namely the grid division of the aerospace grid model. Each area is assigned a virtual node (VN), which is marked as the grid coordinate code. Thereafter, OneWeb is taken as an example to allocate virtual nodes (Figure 6a).

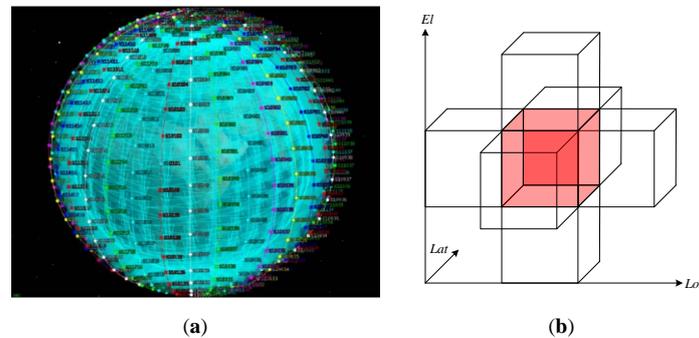


Figure 6. Grid network topology model: (a) LEO (low earth orbit) satellite constellation; (b) Six cell neighborhood.

In the grid network topology model, when the satellite is in a grid cell, the network node related to the cell is held by the satellite. When the satellite leaves the cell, it also loses its holding relationship with the cell. When the satellite enters the area from a neighboring area, it enters from a neighboring cell, and then leaves the cell to go to another neighboring cell. When entering a certain grid routing, the virtual node will identify the six directions in relation to the grid cell (up, down, left, right, front, and back), and the global coordinate system uses the six directions, comprising top, bottom, west, east, north, and south, which correspond to the six neighbors of the cell (Figure 6b). After selecting the appropriate neighboring cell, directional enhancement of the grid routing will be performed. As soon as the satellite enters the cell, it matches the grid code to obtain the corresponding routing table and grid-related information. Combined with the aerospace grid indexing BigTable in Section 2.1.2, it can be seen that if the satellite enters a grid cell, the information associated with the BigTable can be found.

Furthermore, the satellite grid virtual nodes can be divided into sub-grids (smaller and larger scale grids) according to the requirements of applications, as shown in Figure 7. The grids in figure represent both the location range and the virtual nodes. The main idea of the model is to convert the satellite’s highly variable topology into a connection graph between grids. The smaller grid at the center of each grid virtual node will indicate which satellite the virtual node is currently occupying. The stations or users on the ground can also be mapped as nodes in fixed grids. Because GeoSOT-3D is extended by GeoSOT in the height dimension, the ground (2D) and satellite grid nodes (3D) can be integrated and easily converted.

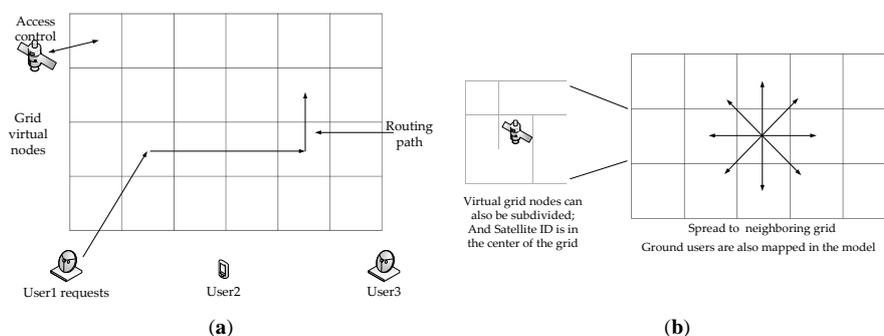


Figure 7. Grid network topology model: (a) virtual nodes (using the grid as a virtual node to connect the network in this area); (b) multi-scale virtual nodes and neighboring grids.

2.3.2. Dynamic Routing Planning Algorithm

The low-orbit satellite constellation system has an improvement in on-board processing capability and will help realize communication between satellites. However, choosing the shortest path in a highly time-varying situation involves a dynamic planning algorithm. This is often used to solve the problem of optimization in a multi-stage, multi-task, and multi-objective decision-making process [1,20]. Therefore, satellite routing planning is an NP-hard (Non-deterministic Polynomial hard) problem, and the satellite route planning algorithm for a complete data transmission process will be given (Figure 8a).

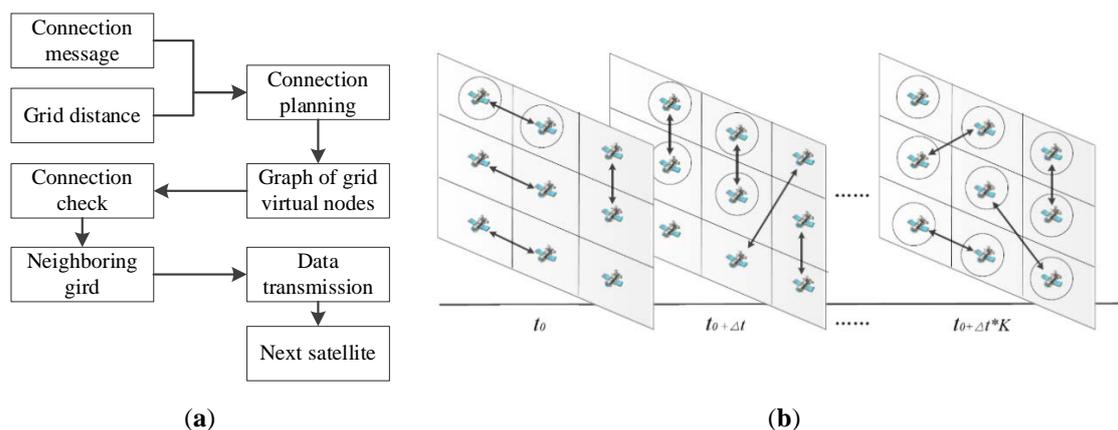


Figure 8. Dynamic routing planning method: (a) routing method; (b) topology of time-slice based on grids.

Based on the idea of spatial grid virtualization, the satellite moving period T in the study period is divided into K sub-periods. When the time ($\Delta t = T/K$) is sufficiently small, the satellite topology can be considered static, and equivalently regarded as the topology of the grids where the satellite is located (Figure 8b). Therefore, a correlating matrix of topology containing satellites and ground targets can be established according to the situation where the cells are occupied in the particular time slice. The size of the entire correlating matrix is $(M + N) \times (M + N)$, where M and N represent the number of satellites and ground targets established, respectively. The main correlating matrix of the satellite topology exists on-board and is updated regularly [26]. The shortest path algorithm is developed based on this matrix, and its time complexity and space complexity are $O((M + N)^2)$. The number of satellites and ground targets are large, and the cost of calculation and storage is too high, not meeting the requirements of on-board algorithms and storage. Considering that visibility is a necessary condition for communication, each time a connection graph is constructed, a step of querying the visibility table is added. This can filter out satellites that are inaccessible and less likely to communicate, and thus reduce the cost of the algorithm and the pressure of on-board storage. Finally, the path is selected based on different decisions, such as minimum steps and shortest distance. Because there must be unconnected satellites in the calculation process, the size of the newly established correlating matrix must be smaller than $(M + N) \times (M + N)$. This algorithm removes the satellites that are not involved in the communication in each time period, and reduces the dimension of the main correlation matrix. The path planning time of the shortest path planning algorithm is shortened, and the time and space complexity of the shortest path planning algorithm is greatly reduced.

3. Experiments and Results

3.1. Experiment Design

According to the proposed methods, the following three sets of experiments were designed: (1) aerospace grid indexing BigTable, (2) satellite visibility analysis, and (3) satellite routing planning. Considering the large number of satellites in the low-orbit satellite constellation, the experiments used

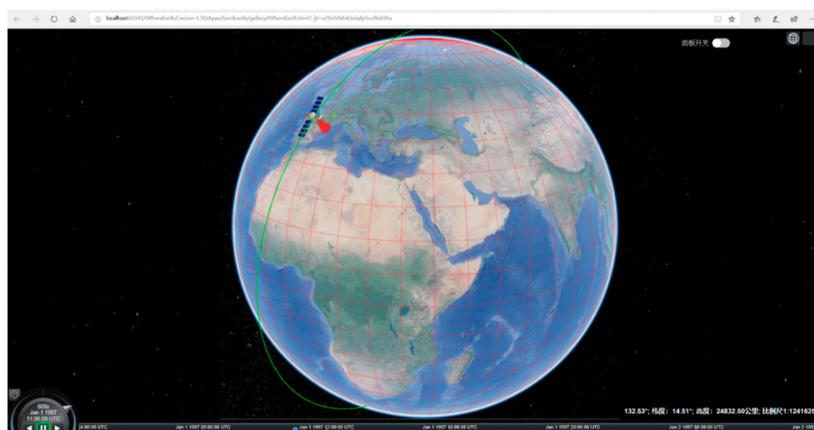
the simulated Walker constellation, with a configuration of 12000/200/2, that is, 200 orbitals, each with 60 satellites. The $satellite_{ij}$ is the j th satellite in the i th orbit of the constellation. The simulation ran from 04:00 26 February 2020 UTC to 04:00 27 February 2020 UTC (Coordinated Universal Time). The STK software was used to generate the simulation constellation, and the J2000 coordinate system was selected in STK.

3.2. Aerospace Grid Indexing BigTable

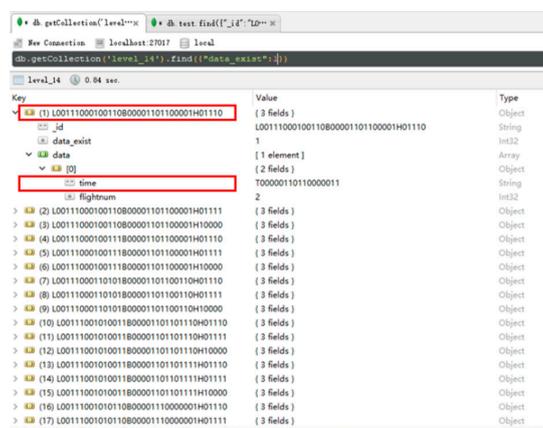
The experiment of grid indexing BigTable included four steps: (1) grid encoding, (2) data storing (in database), (3) creating index, and (4) grid querying. Based on the GeoSOT-3D model and its method, the trajectories of the 12,000 satellites of the simulated constellation during this period were mapped to the grid and stored in the table.

The process was as follows:

Step 1: Grid encoding: The trajectories of 12,000 satellites were mapped to the grids and grid codes were assigned. According to the TLE (Two-Line Element, which is a data format encoding a list of orbital elements of an Earth-orbiting object for a given point in time [39]), the spatial position information of satellites at each time and their corresponding grid codes were calculated. Then, the line object grid modeling and 3-dimensional path-filling algorithm were used to achieve grid mapping of the entire satellite trajectory during the simulation period. The Walker simulation constellation and GeoSOT-3D grid were visualized in CesiumJS (an open source JavaScript library, shown in Figure 9a).



(a)



(b)

Figure 9. System interfaces: (a) satellite with codes in CesiumJS; (b) data stored in MongoDB.

Step 2: Data storing (in database): the calculated data results were generated in the JSON format, imported into the database in batches, and associated with the primary key of the aerospace grid indexing BigTable with codes. Taking the MongoDB database as an example, the `_id` column is the primary key, which is unique and stores the grid code; the data column stores the data attributes; the time column is the time code (Figure 9b).

Step 3: Creating indices: a spatio-temporal secondary index was created on the S-Code and T-Code columns, and a joint ascending index was created for latitude, longitude, and elevation.

Step 4: Grid querying: queries were carried out with the conditions of “grid code” and “latitude, longitude and height” representing the same area, and the number and costing time of returned items was recorded.

In our experiment, when the ninth grid level was chosen, there were 345,720,000 data in database, and for the 15th level, there were 518,400,000 data. Subsequently, we created indices on the code column and the three columns of latitude, longitude, and height separately. After creating the indices, the 15th level was selected for querying. The query results are listed in Table 3. It can be seen that the efficiency of the code method is significantly better than the combined query method of longitude, latitude, and height, with an average increase of 8.82 times.

Table 3. Query results (ms).

Query Method	Satellite Trajectory			Satellite Coverage Area		
	Aera 1	Aera 2	Aera 3	Aera 4	Aera 5	Aera 6
Lon, Lat, Height	504.158	983.956	817.712	347.369	585.005	477.232
GeoSOT-3D code	53.394	95.743	87.416	38.03	85.18	61.47

3.3. Satellite Visibility Analysis

In the simulated constellation, we randomly selected 500, 1000, 2000, 4000, 6000, 8000, 10,000 and 12,000 satellites and calculated the grid codes of each latitude, longitude, and height area at the level of 15th. We then compared the efficiency of the grid method after testing in the BigTable, including a time comparison with the grid geometric visibility algorithm and the traditional geometric visibility algorithm (the satellite visibility means whether the two satellites can see through the earth and atmosphere). The calculation time of the methods above is shown in Table 4 and a comparison of different methods is shown in Figure 10.

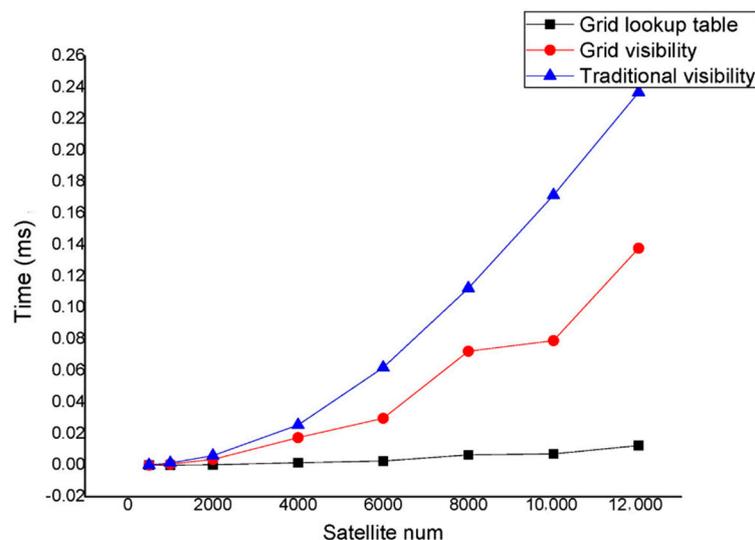


Figure 10. Comparison of visibility calculation efficiency.

Table 4. Visibility calculation results (ms).

Number of Satellites	500	1000	2000	4000	6000	8000	10,000	12,000
Grid lookup table	0.00001	0.00007	0.00024	0.00081	0.00171	0.00559	0.0062	0.01148
Grid visibility	0.00010	0.00072	0.00370	0.01757	0.02984	0.07247	0.07921	0.13709
Traditional Visibility	0.00014	0.00145	0.00612	0.02559	0.06208	0.11249	0.17162	0.23701

It can be seen from Figure 10 that the calculation time using traditional algorithms is related to the number of satellites, and increases exponentially with the number. However, when using the grid code method, only one of the height codes in the two-point connection grid set needs to be negative. This method avoids complex spatial operations, and the calculation efficiency is 1.81 times higher than that of the traditional algorithm.

What is more, the position of the grid relative to the center of the earth is fixed, so the visibility relationship between the cell and other cells can be calculated in advance. When the satellite enters several cells with a clear visibility relationship, it is only necessary to query the grid look-up table calculated in advance. Through the pre-calculated grid visibility look-up table, the visible cells of the grid can be directly queried, and the efficiency is 23.62 times that of the traditional algorithm.

3.4. Satellite Routing Planning

The premise of inter-satellite routing planning is based on spatial grid modeling. The simulation time is divided into several equal time periods, and each time period is considered static. According to the visibility between the cells themselves (look-up table) to determine the visibility between the satellites falling in different cells, a satellite routing association matrix is generated. The ground point is set to a certain point (taking Shenzhen (114° E, 22° N, 50 m) as an example), and a connection graph is established between the satellites according to the visible topological relationship (where 1 is visible; -1 is invisible; 0 indicates that the calculation object is the satellite node itself and no further calculation is required). Taking 10 satellites as an example (a total of 11 network nodes including ground points), the visible connection graph is shown as Equation (5):

$$S = \begin{bmatrix} 0 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 0 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 \\ -1 & -1 & 0 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & 0 & 1 & -1 & -1 & 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 & -1 & 0 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 1 & -1 & -1 & -1 & 0 & -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 & 1 & -1 & -1 & 0 & -1 & 1 & 1 \\ -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 0 & -1 & 1 \\ -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 0 & 1 \\ -1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (5)$$

Combining the selection strategies of the shortest path and the smallest number of steps, the shortest path is calculated according to the routing path algorithm proposed in Figure 8a. The path to the ground point for both strategies is the same for the matrix above, owing to the small size of the selected constellation. When the constellation size increases, such as selecting 100 satellites, there may be different alternate paths using the two different strategies; for example, from 100 to 43, by the shortest distance is (100, 12, 24, 43) and the minimum number of steps is (100, 20, 43), using the Dijkstra algorithm [25].

The experimental results of the two strategies are shown in Table 5 and a comparison is shown in Figure 11. Overall, the efficiency of the grid method is significantly better than that of traditional algorithms. In path planning based on the shortest distance strategy, the efficiency is improved by an

average of almost 29.74 times; and for the minimum number of steps, the improvement is approximately 23.31 times (mainly related to the visibility between satellites, so the efficiency improvement is close).

Table 5. Routing planning results (ms).

	500	1000	2000	4000	6000	8000	10,000	12,000
Grid distance	0.00005	0.0001	0.00028	0.00056	0.00153	0.00385	0.00835	0.0145
Grid step	0.00011	0.00018	0.00035	0.00051	0.00101	0.00191	0.00289	0.00547
Formal distance	0.0012	0.00241	0.01205	0.02209	0.04818	0.08937	0.23128	0.46256
Formal step	0.00208	0.00277	0.00554	0.01109	0.02218	0.03913	0.07392	0.13306

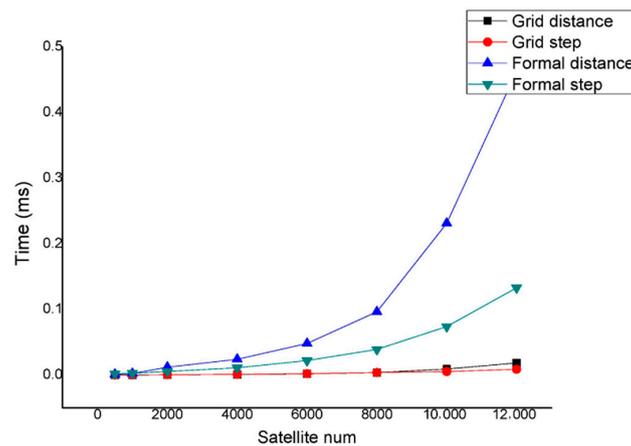


Figure 11. Comparison of route planning.

4. Discussion

First, we analyzed the time complexity of the satellite interconnection algorithms proposed in this study. In the traditional inter-satellite visibility analysis algorithm, we assumed that there are N satellites in the aerospace industry. For each satellite, it is necessary to measure the distance to other satellites to determine whether its angle is within the visible range and whether the elevation is obstructed by the earth and the atmosphere. The time complexity is $O(N)$, and it increases linearly with the number of satellites. However, when querying in the BigTable based on the grid's visibility, the 3-dimensional space coordinates of the satellite are stored in the database in code format. The idea at the basis of this method is to employ the "space for time" concept to convert the high-dimensional floating operations into one-dimensional matching operations by querying the inherent "visible" attribute of the grid. The time complexity of the code query is $O(\log N)$. Especially, when the number of satellites increases, the time complexity of algorithm is greatly reduced compared to $O(N)$. In addition, in the grid network topology model and routing algorithm, we established an association matrix containing the topology information of satellites and ground targets. The size of the matrix is $(M + N) \times (M + N)$, where M and N represent the number of satellites and ground targets established, respectively. The matrix is then transformed into the topological matrix of the corresponding cells, and their visibilities are set to be the necessary conditions for communication to reduce the size of the correlation matrix and further reduce the cost of the algorithm and storage.

According to the analysis of the experiment results, we created an aerospace grid indexing BigTable that verifies the feasibility of gridding satellite trajectory data, the efficiency of inter-satellite visibility analysis, the feasibility of grid network topology, and the high efficiency of path planning. In the inter-satellite visibility analysis, by calculating the visible grid lookup table in advance, we only need to query the visible cells of the grid, which is 23.62 times faster than traditional algorithms. In route planning calculations, path planning is based on visibility analysis, which is 29.74 times faster than traditional algorithms. In particular, after the construction of BigTable is completed, the large increase

in the number of satellites should have a much smaller impact on the calculation of the grid algorithm than traditional methods. This “space-for-time” method has efficiency advantages in inter-satellite interconnection calculations, and is an efficient satellite interconnection calculation method.

In addition, it should be noted that in practical applications, the complexity of the connection situation, and other factors such as delay and interference should also be considered. The existing experimental results were completed without considering the input/output (io) overhead of database and other factors, and theoretically improved more. Further study should be combined with a variety of possible influencing factors in practical applications for a more accurate analysis.

5. Conclusions

This research is based on the background of space-based big data and satellite Internet, combined with low-orbit satellite constellations and a complex network communication environment, in particular the construction of the PNTRC system. To address the bottlenecks in large-scale node space computing, we introduced a global space grid, GeoSOT-3D. Based on this grid, an efficient calculation method of spatial inter-connection between satellite constellations is proposed, according to the concept of “storage for computing” and the high computational efficiency of the spatial grid model. The foundation of this method is to establish an aerospace grid indexing BigTable with grid code as the primary key correlating spatial information. The satellite trajectories and resources are grid modeled separately. When distant satellite resources cannot be directly transmitted to ground users, inter-satellite data transmission is required. This process involves establishing a grid virtual node model, using the virtual topology, querying the grid through the BigTable, and iterating to meet the coverage requirements of the inter-satellite transmission path. The experiment started with the establishment of the aerospace grid indexing BigTable, and verified the feasibility and efficiency of the inter-satellite computing and application process of organization, storage, calculation, and planning. This “space-for-time” approach has efficiency advantages in inter-satellite interconnected computing. After considering the actual physical expenses, the efficiency of visibility analysis was increased by 23.62 times and the efficiency of route planning increased by 29.74 times.

The application and development of the grid inter-satellite connection and calculation algorithm proposed in this study will be conducive to the construction of a location-based integrated network communication between satellites and earth as well as the realization of more efficient space-based information intelligent services. However, since this is the first study on application of the grid system to the interconnection algorithm of low-orbit satellite constellations, there are still many issues worthy of in-depth study. These include: index optimization of the aerospace grid indexing BigTable, storage optimization of the on-board algorithm, optimization of the multi-task and multi-object algorithm of the dynamic planning of inter-satellite routing, amongst others. The spatial interconnection algorithm is not only applicable to remote sensing constellations but is also applicable to other constellations that require interconnection, such as communication and navigation. We hope that the algorithm can be used in emergency communications, disaster warning, and maritime rescue, and can contribute to the next generation of satellite Internet and “satellite-ground” integrated networks.

Author Contributions: S.L. (Shuang Li) conceived, designed, and performed the experiments and wrote the manuscript; K.H. helped perform the experiments and offered helpful suggestions for experiments; C.C. supervised the study; S.L. (Shizhong Li) offered helpful suggestions and provided some data; B.C. offered helpful suggestions and reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Plan (Grant No. 2018YFB0505300).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sanctis, M.D.; Cianca, E.; Araniti, G.; Bisio, I.; Prasad, R. Satellite communications supporting internet of remote things. *IEEE Internet Things J.* **2016**, *3*, 113–123. [[CrossRef](#)]

2. Liu, F.; Li, X.; Wang, J.; Zhang, J. An adaptive UWB/MEMS-IMU complementary Kalman filter for indoor location in NLOS environment. *Remote Sens.* **2019**, *11*, 2628. [[CrossRef](#)]
3. Wei, J.; Han, J.; Cao, S. Satellite IoT edge intelligent computing: A research on architecture. *Electronics* **2019**, *8*, 1247. [[CrossRef](#)]
4. Li, D.; Shen, X.; Gong, J.; Zhang, J.; Lu, J. On construction of China's space information network. *Geomat. Inf. Sci. Wuhan Univ.* **2015**, *40*, 711–715.
5. SpaceX to Launch Astronauts—And a New Era of Private Human Spaceflight. Available online: <https://www.nature.com/articles/d41586-020-01554-8> (accessed on 29 May 2020).
6. Mann, A. Crewed launch deepens ties between NASA and SpaceX. *Science* **2020**, *368*, 811–812. [[CrossRef](#)] [[PubMed](#)]
7. Kodheli, O.; Guidotti, A.; Vanelli-Coralli, A. Integration of satellites in 5G through LEO constellations. In Proceedings of the 2017 IEEE Global Communications Conference (GLOBECOM 2017), Singapore, 4–8 December 2017; pp. 1–6.
8. Chan, C.C.; Al-Hourani, A.; Choi, J.; Gomez, K.M.; Kandeepan, S. Performance modeling framework for IoT-over-satellite using shared radio spectrum. *Remote Sens.* **2020**, *12*, 1666. [[CrossRef](#)]
9. Graydon, M.; Parks, L. 'Connecting the unconnected': A critical assessment of US satellite Internet services. *Media Cult. Soc.* **2020**, *42*, 260–276. [[CrossRef](#)]
10. Liu, M.; Li, K.; Song, H.; Chen, Y.; Gao, X.; Gong, F. Using heterogeneous satellites for passive detection of moving aerial target. *Remote Sens.* **2020**, *12*, 1150. [[CrossRef](#)]
11. Li, D.; Shen, X.; Li, D.; Li, S. On civil-military integrated space-based real-time information service system. *Geomat. Inf. Sci. Wuhan Univ.* **2017**, *42*, 1501–1505.
12. Shimizu, K.; Ota, T.; Mizoue, N.; Saito, H. Comparison of multi-temporal PlanetScope data with Landsat 8 and Sentinel-2 data for estimating Airborne LiDAR Derived Canopy Height in Temperate forests. *Remote Sens.* **2020**, *12*, 1876. [[CrossRef](#)]
13. Braten, L.E. Satellite visibility in Northern Europe based on digital maps. *Int. J. Satell. Comm.* **2000**, *18*, 47–62. [[CrossRef](#)]
14. Han, S.H.; Gui, Q.M.; Li, J.W. Analysis of the connectivity and robustness of inter-satellite links in a constellation. *Sci. China Phys. Mech.* **2011**, *54*, 991–995. [[CrossRef](#)]
15. Sharaf, M.A. Satellite to satellite visibility. *Open Astron.* **2012**, *5*, 26–40. [[CrossRef](#)]
16. Wang, X.; Han, C.; Yang, P.; Sun, X. Onboard satellite visibility prediction using metamodeling based framework. *Aerosp. Sci. Technol.* **2019**, *94*, 1–9. [[CrossRef](#)]
17. STK Access Tool. Available online: <http://help.agi.com/stk/index.htm#stk/tools-01.htm> (accessed on 23 May 2020).
18. Tilwari, V.; Dimyati, K.; Hindia, M.N.; Mohamed Noor Izam, T.F.B.T.; Amiri, I.S. EMBLR: A high-performance optimal routing approach for D2D communications in large-scale IoT 5G network. *Symmetry* **2020**, *12*, 438. [[CrossRef](#)]
19. Xia, X.; Yang, B.; Liu, Z.; An, K.; Guo, K. Performance evaluation of a full-duplex relaying-enabled satellite sensor network. *Sensors* **2019**, *19*, 5453. [[CrossRef](#)]
20. Lysogor, I.; Voskov, L.; Rolich, A.; Efremov, S. Study of data transfer in a heterogeneous LoRa-Satellite network for the Internet of Remote Things. *Sensors* **2019**, *19*, 3384. [[CrossRef](#)]
21. Li, X.; Zhang, X.; Ren, X.; Fritsche, M.; Wickert, J.; Schuh, H. Precise positioning with current multi-constellation Global Navigation Satellite Systems: GPS, GLONASS, Galileo and BeiDou. *Sci. Rep.* **2015**, *5*, 8328. [[CrossRef](#)]
22. Sun, L.; Wang, Y.; Huang, W.; Yang, J.; Zhou, Y.; Yang, D. Inter-satellite communication and ranging link assignment for navigation satellite systems. *GPS Solut.* **2018**, *22*, 38–51. [[CrossRef](#)]
23. Liu, Z.; Zhu, J.; Zhang, J.; Liu, Q. Routing algorithm design of satellite network architecture based on SDN and ICN. *Int. J. Satell. Comm. Netw.* **2020**, *38*, 1–15. [[CrossRef](#)]
24. Li, J.; Ke, L.; Ye, G.; Zhang, T. Ant colony optimisation for the routing problem in the constellation network with node satellite constraint. *Int. J. Bio-Inspired Comput.* **2017**, *10*, 267–274. [[CrossRef](#)]
25. Li, B.; Ge, H.; Ge, M.; Nie, L.; Shen, Y.; Schuh, H. LEO enhanced global navigation satellite system (LeGNSS) for real-time precise positioning services. *Adv. Space Res.* **2019**, *63*, 73–93. [[CrossRef](#)]
26. Madni, M.A.A.; Raad, R.; Iranmanesh, S. DTN and Non-DTN routing protocols for inter-cubesat communications: A comprehensive survey. *Electronics* **2020**, *9*, 482. [[CrossRef](#)]
27. Li, X.; Jiang, Z.; Ma, F.; Lv, H.; Yuan, Y.; Li, X. LEO precise orbit determination with inter-satellite links. *Remote Sens.* **2019**, *11*, 2117. [[CrossRef](#)]

28. Discrete Global Grid. Available online: https://en.wikipedia.org/wiki/Discrete_global_grid (accessed on 31 May 2020).
29. Tong, X.; Ben, J.; Wang, Y.; Zhang, Y. Efficient encoding and spatial operation scheme for aperture 4 hexagonal discrete global grid system. *Int. J. Geogr. Inf. Sci.* **2013**, *27*, 898–921. [[CrossRef](#)]
30. Zhou, L.C.; Lian, W.J.; Zhang, Y.D.; Lin, B.X. A topology preserving gridding method for vector features in Discrete Global Grid Systems. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 168. [[CrossRef](#)]
31. Cheng, C.; Tong, X.; Chen, B.; Zhai, W. A subdivision method to unify the existing latitude and longitude grids. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 161. [[CrossRef](#)]
32. Li, S.; Pu, G.; Cheng, C.; Chen, B. Method for managing and querying geo-spatial data using a grid-code-array spatial index. *Earth Sci. Inform.* **2018**, *12*, 173–181. [[CrossRef](#)]
33. Li, S.; Cheng, C.; Pu, G.; Chen, B. QRA-Grid: Quantitative risk analysis and grid-based pre-warning model for urban natural gas pipeline. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 122. [[CrossRef](#)]
34. Miao, X.; Cheng, C.; Zhai, W.; Ren, F.; Zhang, B.; Li, S.; Zhang, J.; Zhang, H. A low-altitude flight conflict detection algorithm based on a multilevel grid spatiotemporal index. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 289. [[CrossRef](#)]
35. Overview of Cloud Bigtable. Available online: <https://cloud.google.com/bigtable/docs/overview> (accessed on 21 June 2020).
36. Park, S.; Ko, D.; Song, S. Parallel insertion and indexing method for large amount of spatiotemporal data using dynamic multilevel grid technique. *Appl. Sci.* **2019**, *9*, 4261. [[CrossRef](#)]
37. Tong, X.; Cheng, C.; Wang, R.; Ding, L.; Zhang, Y.; Lai, G.; Wang, L.; Chen, B. An efficient integer coding index algorithm for multi-scale time information management. *Data Knowl. Eng.* **2019**, *119*, 123–138. [[CrossRef](#)]
38. The Beauty of Bresenham’s Algorithm. Available online: <http://members.chello.at/easyfilter/bresenham.html> (accessed on 19 May 2020).
39. Two-Line Element Set. Available online: https://en.wikipedia.org/wiki/Two-line_element_set (accessed on 19 June 2020).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).