THE SUBDIVISION OF PARTIAL GEO-GRID BASED ON GLOBAL GEO-GRID FRAME

TONG Xiaochong *, BEN Jin, JI Song, ZHANG Yongsheng

Institute of Surveying and Mapping , Zhengzhou, 450052, China - txchr@yahoo.com.cn

KEY WORDS: Global Geo-Grid System, Partial Geo-Grid, District Clipping, Hierarchical Data, Hexagonal Grid, ISEA

ABSTRACT:

Firstly, start with the application and requirement of Global Geo-Grid System, the paper analyzes the essentiality of generating partial grids. On the basis of them, the fundermental thought of generating partial grid which is under the global geo-grid frame are proposed, detailed edge problems of partial high-precision grid are analyzed. Start with the ubiety between a point and a spherical triangle, we discuss the ubiety between a point and a spherical random polygon, and then edge simplifying algorithm of spherical random polygon are studied, as well as co-relationship between points and spherical random polygon, grid district clipping based on span the developed surface, grid data hierarchical creating algorithm. In the end we prove exactness and efficiency of the algorithm through the experiments.

1. INTRODUCTION

With the enrichment of earth observing measures, the interesting field of traditional photogrammetry and remote sensing monitoring has expanded gradually to the whole world. In order to satisfy the need of scientific research and national defense, researchers have done much work of dynamic remote sensing monitoring these years. All the work is within the scope of the whole earth and its importance has aroused people's interest bit by bit. Under this background, traditional planar data structure no longer satisfies the requirements of global spatial information management. Constructing a continuous, hierarchical and dynamic sphere data structure has therefore become a significant research direction. How to design a global geo-grid frame that fits multi-scale presentation and management of global remote sensing data has thus to be solved promptly, while Global Geo-Grid System is just such an effective frame (A. Vince, 2006; Sahr, 2003; Goodchild, 2000; White, 1998; Zhao, 2003).

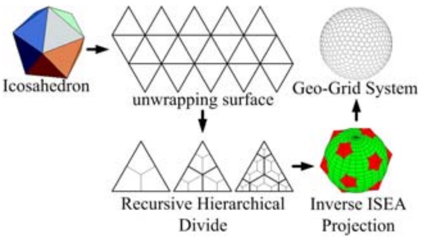


Figure 1. The subdivision methods of GGS

In the early researches, some researchers tried to construct global grid on geographic coordinate system (Tobler, 1986). Although they had obtained great results, they couldn't solve the problem of serious unit distortion and uneven distribution. Thus researchers try to find the other ways. One way is using regular polyhedron instead of sphere and, compared to the other methods, this method has its own advantages (White, 1998; Sahr, 2003). Author and others discussed the constructing ideas of Global Geo-Grid based on the icosahedron in the literature (Tong, 2006; Zhang, 2006) (Figure 1), but in practice the local area is more useful than the global scope. So how to construct a partial geo-grid in the global geo-grid frame is the concern of many scholars. Based on the global multi-resolution geo-grid the paper brings forward a method of constructing partial geo-grid which has the same attributes, and it can easily bring into the global geo-grid frame.

2. THE EDGE PROGRAM OF LOCAL AREA

The subject investigated of Global Geo-grid generation algorithm is the whole sphere itself, and it need not consider the edge of geo-grid, but toward the partial geo-grid, the program of edge become prominent especially. Spherical arbitrary irregular regions are all compose of major arcs which links end to end. How to find the units which are in the closed area is the first program need to solve. So during the process of design algorithm, we should consider the ubiety between a point and a spherical random polygon.

2.1 The Ubiety Between Points and Spherical Triangle Areas

It is difficult to judge whether a spherical point C is inside any of the spherical areas as $P_0 \ldots P_iP_j \ldots P_{N-1}$ or not. So, above all, we discuss the ubiety between an arbitrary point C and a spherical triangle. On a spherical triangular of an unfolded spherical icosahedron, we could connect two arbitrary points P_iP_j with the projection center A (gravity center) to build a spherical triangle, and then judge the ubiety between C and the spherical triangle $\triangle P_iP_jA$, as shown in Figure.2

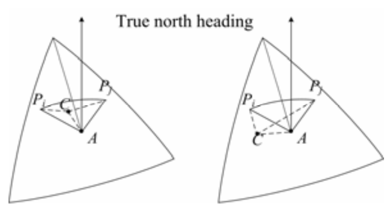


Figure 2. The ubiety between a point and a spherical triangle

According to literature (Snyder, 1992), if we choose Snyder polyhedral equal-area projection to build the ubiety between the plane and the sphere, though errors could be controlled within a grid unit by hierarchical grids, it is impossible that the edges of spherical great arcs on the unfolded sphere could be straight lines, so it may cause mistakes to directly judge the ascription question without considering this kind of distortion. Literature (Tong, 2006) analyzes the complexity of ascription calculating problem when the point and the edge are directly projected to the plane. This paper adopts inverse computation method to judge directly on the sphere. The following is a simple judging principle:

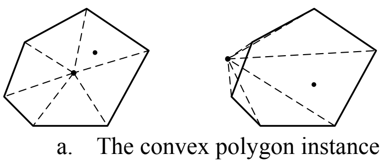
1. Connect CA, CP_i and CP_j on the sphere, then calculate the areas S, S_1, S_2 and S_3 of spherical triangles $\triangle P_iP_jA$, $\triangle P_iP_jC$, $\triangle P_iCA$ and $\triangle CP_jA$ according to spherical triangle functions.
2. When $S=S_1+S_2+S_3$, it can be judged that the point C is inside the spherical triangle $\triangle P_iP_jA$, and terminate; Whereas, C is outside the spherical triangle $\triangle P_iP_jA$, then turn to step3.
3. When $S+S_1=S_2+S_3$, it can be judged that $\overset{\frown}{AC}$ and $\overset{\frown}{P_iP_j}$ intersect, then record the values of i and j, terminate; whereas, terminate.

By this method, we could determine the ubiety between points and spherical triangles, building a foundation for the further work to judge whether a point is inside a spherical polygon area or not.

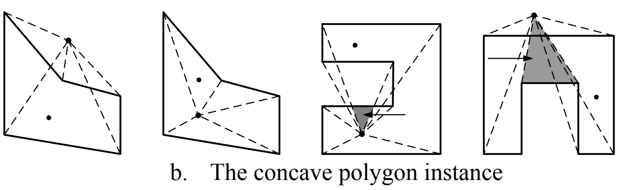
2.2 The Ubiety Between Points and Spherical Polygon Regions

In order to simplify the above problems, the edges of spherical areas are positioned in one of the icosahedron's triangle. So, now the problems are determining the ubiety between point C and spherical polygon $P_0 \cdots P_iP_j \cdots P_{N-1}$.

Analysis reveals that the position of projection center A and the type of spherical polygon are two important factors for the ascription of point C. Figure 3 shows some of the different ascription examples of point C, which are caused by factors referred above.



a. The convex polygon instance



b. The concave polygon instance

Figure 3. Different instance result in different ascription of point C

Because of the complexity of concave polygon, a rule to determine the ubiety between points and convex polygons are firstly given as follows.

1. Set initial value for unknown parameters, let i=0， NUM=0，j=1。

2. Connect the great arcs of AP_i and AP_j, and determine the ubiety between C and spherical $\triangle P_iP_jA$ by the methods proposed in section 2.1. If C falls in $\triangle P_iP_jA$, NUM= NUM+1. When NUM＞1, go to step 4, or else go to step 3.

3. i=i+1. If i＞N-1, go to step 4, while i＜N-1, j=i+1 and go to step 2. When i=N-1, j=0 and go to step 2.

4. If NUM=1, C falls in the convex polygon, or C falls out of the convex polygon.

Algorithms for concave polygon are comparatively more complex. Classical clipping algorithm for concave polygons proposed in paper (David, 2002) can clip a spherical concave polygon into several spherical convex polygons, which are then processed separately.

After the 4 steps, the aspiration of C can be determined most of the time. However, this algorithm has detrimental limitations as follows. 1. If a spherical hexagon has several edges, the aspiration of C should be calculated several times and, which directly influences the efficiency of generating high precision partial grids. 2. The partition of spherical concave polygons is not optimal, for the least convex polygons can not be generated. Additionally, for these overlapping spherical convex polygons, such as star-shaped spherical polygon, the algorithm shows its inability. Therefore, a new algorithm should have to be designed to ensure the efficiency and correctness of grid generation.

Here, a coarse-to-fine thought is proposed as follows. 1. Simplify complex edges hierarchically to form multi-scale simplified area. 2. Determine the aspiration of C to the simplified area and, if C belongs to the simplified area, then C is regarded as falling in the polygon area, or else determines the aspiration relationship between C and the previous simplified area layer. This process doesn't stop until the original area is reached or the ubiety of C and the original spherical polygon is determined.

To adopt the coarse-to-fine determination method, the edges' simplification mechanism has to be constructed firstly. Assume the original area is S_0, and the area after k times simplification is S_k (k=0,1,2, ⋯), where the requirement of $S_k \subset S_{k-1}$ has to be fulfilled. Then, the simplification strategy can be given as follows.

1. The edge points of the original area is anticlockwise numbered as $P_0 \cdots P_iP_j \cdots P_{N-1}$ to construct list array E_k on different scale. The edge points' number is stored in the list array. Let the initial value of k is 1, and E_0 is such a list array when the original edge points are not simplified.

2. Conduct the simplification process for k times. Assume that the edge points in E_{k-1} is expressed by PE_j, where $j \in [0,N_{k-1}-1]$ and N_{k-1} is the point number on the (k-1)th layer. Search all the concave points in E_{k-1} list by the convex polygon partition method (David, 2002). These points' subscript is noted

3. form the original j to T_i, where $i\in[0, U]$ whose initial value is 0 and U is the number of concave points on this layer.

4. Start from the concave point PE_{T_i}, whose number is T_i, anticlockwise search the next simplification point. If the next point PE_{T_i+1} is a concave point, then it is preserved and goes to step 4. However, if the next point PE_{T_i+1} is a convex point, then the searching process goes on, $i=i+1$, let $M=T_i+1$ and go to step 5。

5. PE_{T_i+1} is preserved and added to the simplified edge list E_k, then $i=i+1$ and go to step 3.

6. If $M>N_{k-1}-4$, go to step 6, or determine the convexity and concavity of PE_{M+1}, PE_{M+2} and PE_{M+3}. If they are all convex points, add PE_{M+3} to the list E_k and let $M=M+3$ and go to step 5. If PE_{M+m} (m=1,2,3) are concave points, then $i=i+1$ and add PE_{T_i} to list E_k.

7. $k=k+1$ and when $k>$the maximum simplification layer, terminate the process and record the simplification results $E_1, E_2, \cdots$, otherwise, go to step 2.

Figure4 shows the simplification process of Chinese mainland borderline, where three simplification process is conducted. The mainland borderline is a quite irregular spherical polygon and holds such geometrical phenomenon as convex, concave and overlapping. The partition of concave polygon can not be applied to every situation. However, after some edge simplification process, special geometrical phenomenon as overlapping will disappear and the edge points of the area will gradually decrease to be appropriate for the partition method of concave polygon and the speed can be ensured.

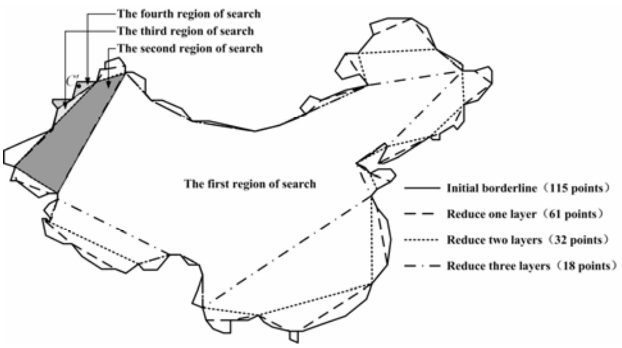


Figure 4. The simplification result of Chinese mainland borderline

In the following, an algorithm to determine the ubiety between spherical points and spherical random polygons is presented.

1. Given the simplification layer K, simplify the spherical polygon by the edge simplification algorithm and record the list array E_k ($k\in[0,K]$) for the simplified edges.

2. Partition the simplified layer E_K by the spherical random polygon's partition algorithm to obtain different spherical convex polygons (David, 2002). Determine the ubiety between edge points and the convex polygons by spherical polygon determining algorithm. And if a point belongs to one of the convex polygons, then the point is assumed to be in the random polygon, otherwise, let $k=K$ and go to step 3.

3. Record the intersection points P_i and P_j between the great arc P_iP_j and AC. Search points i and j in the simplified edge lists and record the number p, q (no more than two) between i and j in E_{k-1}. If p, q do not exist, then C is not in the given spherical polygon. Whereas, search continuously in the convex polygon $P_iP_pP_qP_j$ or $P_iP_pP_j$, and determine the ubiety between C and the spherical convex polygon by simplified determination rule. If C belongs to one of the convex polygon, then C is in the given polygon, otherwise, $k=k-1$ and go to step 3. When $k<0$, terminate the whole process.

Figure 4 shows the searching process of point C, and the searching process is accomplished in 4 regions. The searching regions decrease gradually by applying spherical polygon partition algorithm once. From the second time, the determination is completed in the spherical convex polygon whose edges are no more than 4. Therefore, the algorithm designed above can not only ensure correct subordination relationships, but also greatly save the searching time and provide a necessary basis for the generation of dense, precise partial geo-gird globally.

3. REGION CUTTING AND PARTIAL GRID GENERATION ALGORITHM

The above spherical polygon problems are all considered on one of icosahedrons' planes. Although the determination of point's ascription and the simplification of spherical polygon can be applied globally, but the segmentation algorithm of spherical polygon includes Snyder projection computation and has to be conducted in one plane. We unwrap spherical icosahedrons by the method shown in figure 1, so, the problem of determining whether a point falls in the spherical polygon should be transformed and processed in different planes of icosahedrons. Therefore, two problems have to be considered: 1. Surface-striding cutting of spherical polygon region; 2. Rapid generation of partial grids.

3.1 Surface-striding Cutting of Spherical Polygon Region

Since the grid data is generated in the unwrapped surface of spherical icosahedrons, the cutting of partial high-precision grid edges involves unavoidably the problem of surface-striding. The basic operation process of surface-striding is given as follows.

1. Compute the serial number S_i of triangular plane which containing all the spherical points by Snyder projection (Snyder, 1992). Select one point as the start point. What should be satisfied is that the start point and its right points are not in the same triangle plane. Number anticlockwise these edge points P_i of spherical polygon into $i\in[1,N]$, and build edge list array E_k in different triangle planes. Add P_i into E_1 ,and let $i=1$, $k=S_1$.

2. If $i<N$, let $i=i+1$. And if $S_i=S_{i-1}$, add P_i into E_k, go to step 2, while if $S_i \neq S_{i-1}$, compute the intersecting point P^k between the great arc $\overset{\frown}{P_{i-1}P_i}$ and the spherical triangles S_{i-1} and S_i , add P^k into E_k ,go to step 3; when $i \geq N$, go to step 4.

3. $k=S_i$, add P^{k-1} and S_i into list E_k , then go to step 2.

4. Check the entire edge list E_k. If the triangle plane which contains the previous point of the second point $E_{k,2}$ has the same serial number as the triangle plane which contains the next point of $E_{k,end-1}$ (the point before the last point), the spherical polygon on the triangle planes are closed. Otherwise, add all the vertexes of icosahedrons which encircled by triangle planes whose number is S_i into chain E_k, and spherical polygons of this triangle plane are closed.

The paper above gives a surface-striding cutting algorithm for spherical polygon regions. This algorithm divides a spherical polygon region into several polygon regions which are in different spherical triangle surface. In order to guarantee the efficiency of the algorithm, the segmentation step can be put after step 1 of the ubiety exterminating algorithm between spherical points and spherical random polygons to cut the simplified region directly (chapter 2.2). Since the ascription property for spherical edge points have been determined by Snyder projection, the relationship between points and spherical polygons can be determined on a single spherical triangle plane, and therefore, the amount of computation is massively reduced. In the following section, we put forward the algorithm of the partial high-precision grid in detail, before which, the edge ascription property has to be determined.

3.2 Generating of the Partial Grid Data

There are mainly two questions to be discussed here. 1. Determining the approximate region of the generated partial geo-grid; 2. Generation algorithms for hierarchical and high-precision partial geo-grid.

Although this paper has discussed the ubiety between points and spherical random polygons, in generating the partial high-precision grids with boundaries, judging whether the points, or more importantly, the hexagonal grid units are in the region or not is the actual question that needs to be solved. Figure 5 illustrates some ubiety between the edge and the hexagonal grid unit, we prescribe that when instance of (a)(b)(c) is met, the hexagonal grid unit belongs to the selected region, while the instance of (d)(e) do not belong to the selected region, and judging the ascription relationships of hexagonal grid units equals to judging the ascription relationships of the hexagonal grid's center. There are no doubt about the ascription of (a)(b), since the unit's area in the selected region is larger than that outside of the selected region, and so does (d)(e). However, the instance of (c) needs to be explained, because the multi-resolution hexagonal grids are symmetric to their center, which means the low-resolution unit's center is the center of a higher grid level. Although, on a given grid layer, the unit's area in the selected region is smaller than the area outside the selected region, the center-symmetric units on higher grid layers may not meet this condition (Figure 5(c)).

Since partial grids have more data volume and higher accuracy compared with global grids, the grid-cluster generating algorithm designed in literature (Ben, 2006) can not meet the requirements of high efficiency and massive data. We have designed an algorithm for generating hierarchical partial grids.

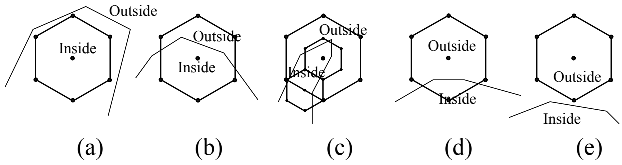


Figure 5. The ubiety between the edge and the hexagonal grid unit

The algorithm makes the recalculation of redundant data unnecessary and improves efficiency. This generating idea also demonstrates the instance of Figure 5(c): if the unit on a given layer isn't recorded in the region, then the unit on the next layer can't be calculated. According to our analysis, when the partition layer goes to infinite, the preserving style of figure 5(c) can ensure the edge of the grid infinitely goes closely to any specified boundary (illustrated in Figure 6).

Now we will give the principle of determining the approximate region of the generated partial grids. Because we have used range partitioning to store the hexagonal grid data (Tong, 2006), and for the convenience of computer processing, the approximate region generated by partial grids is determined similarly. The process is as follows: project the spherical polygon's boundary points to the Snyder plane, record the triangular plane they belong to, give the initial partition layer N_0, acquire the discrete grid coordinates of these points according to literature (Ben, 2006), calculate the maximum and minimum coordinate I_{min}, I_{max}, J_{min}, J_{max} in both direction I and J and the same parallelogram plane. Then, in this parallelogram, the approximate region of the generated local grids is $I \in [I_{min}, I_{max}]$, $J \in [J_{min}, J_{max}]$, as shown in Figure 7.

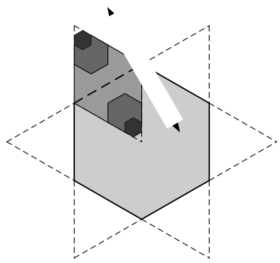


Figure 6

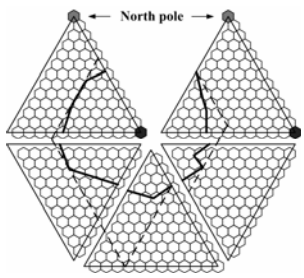


Figure 7

Figure 6. The ultimate situation of plotting an initial unit illimitably

Figure 7. The rough area of generating partial grid(dashed)

In this region, use the type I grid-cluster generation algorithm proposed in literature (Ben, 2006) to generate the desired grids of N_0 layers, and then estimate the ascription of units' centers by the algorithm proposed in section 2. If a unit belongs to the spherical polygon, preserve it; otherwise, delete it.

There are two ideas to generate high-resolution grid: one is to generate grid units on a single layer, which does not use grid data of adjacent layers; another one is generating hierarchical grids, and if the Nth layer grids are to be generated, grid data of the (N-1)th or the (N+1)th layer has to be used. The former idea requires small computation amount in the generating of single layer grid data and is easy to be applied, while the latter outshines in the generating of multiple-layer grid data as well as partial high-precision grid data. In the following, the generating methods for hierarchical grids are discussed in detail. There are two instances in generating hierarchical grid, one is generating the Nth layer grids by the (N-1)th layer grids, and another one is generating the Nth layer grids by the (N+1)th layer grids (shown in Figure 8).

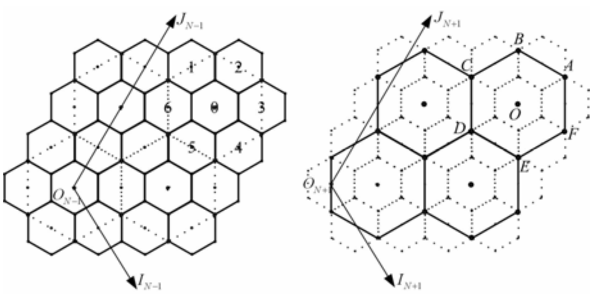


Figure 8. Two direction of generating hierarchical grids

First of all, we consider the first instance. As the unit data is stored by discrete grid coordinates in terms of the row order (order I), the newly generated units are stored in the same way. Suppose the discrete grid coordinates of unit data in the initial layer is (i^0, j^0) (note: the initial units may not be continuous arranged), the coordinates(planar coordinates) of the hexagonal units are expressed by $(x^M_{i^0,j^0}, y^M_{i^0,j^0})$, where M=0,1,2,3,4,5,6 and represents the centers' coordinates of the newly generated seven units. The order of the units are allocated according to Figure 8(a), where M= O, A,B,C,D,E,F means the coordinates of the units' centers and angular points. The angular points' orders are allocated according to Figure 8(b), then the formula that fits the first situation for calculating the central points of the newly generated grid units is:

$$\begin{cases} x^0_{i^0,j^0} = x^O_{i^0,j^0}, & y^0_{i^0,j^0} = y^O_{i^0,j^0}, & \text{Discrete grid coordinates}\left(2\cdot i_0+1, 2\cdot j_0+1\right) \\ x^1_{i^0,j^0} = \dfrac{x^B_{i^0,j^0}+x^C_{i^0,j^0}}{2}, & y^1_{i^0,j^0} = \dfrac{y^B_{i^0,j^0}+y^C_{i^0,j^0}}{2}, & \text{Discrete grid coordinates}\left(2\cdot i_0, 2\cdot j_0+1\right) \\ x^2_{i^0,j^0} = \dfrac{x^A_{i^0,j^0}+x^B_{i^0,j^0}}{2}, & y^2_{i^0,j^0} = \dfrac{y^A_{i^0,j^0}+y^B_{i^0,j^0}}{2}, & \text{Discrete grid coordinates}\left(2\cdot i_0+1, 2\cdot (j_0+1)\right) \\ x^3_{i^0,j^0} = \dfrac{x^A_{i^0,j^0}+x^F_{i^0,j^0}}{2}, & y^3_{i^0,j^0} = \dfrac{y^A_{i^0,j^0}+y^F_{i^0,j^0}}{2}, & \text{Discrete grid coordinates}\left(2\cdot (i_0+1), 2\cdot (j_0+1)\right) \\ x^4_{i^0,j^0} = \dfrac{x^E_{i^0,j^0}+x^F_{i^0,j^0}}{2}, & y^4_{i^0,j^0} = \dfrac{y^E_{i^0,j^0}+y^F_{i^0,j^0}}{2}, & \text{Discrete grid coordinates}\left(2\cdot (i_0+1), 2\cdot j_0+1\right) \\ x^5_{i^0,j^0} = \dfrac{x^D_{i^0,j^0}+x^E_{i^0,j^0}}{2}, & y^5_{i^0,j^0} = \dfrac{y^D_{i^0,j^0}+y^E_{i^0,j^0}}{2}, & \text{Discrete grid coordinates}\left(2\cdot i_0+1, 2\cdot j_0\right) \\ x^6_{i^0,j^0} = \dfrac{x^C_{i^0,j^0}+x^D_{i^0,j^0}}{2}, & y^6_{i^0,j^0} = \dfrac{y^C_{i^0,j^0}+y^D_{i^0,j^0}}{2}, & \text{Discrete grid coordinates}\left(2\cdot i_0, 2\cdot j_0\right) \end{cases} \tag{1}$$

Units' boundary can be obtained by applying and zooming formula (1) presented in literature (Ben, 2006). The zoom parameter is $1/2^N$, and the new generated units' coordinates on the Nth discrete grid coordinate system layer can also be obtained.

The second instance is generating grids on the Nth layer by grids on the N+1th layer. Actually, this case equals to sub-sampling, which means generating low-resolution grid units by high-resolution grid units. In fact, the grids' centers and boundaries are hidden in high-resolution grids, and what we have to do is connecting the gird units' order and number as shown formula 2. However, the sub-sampling is based on the premise that the central units' neighboring units must exist, otherwise, only the coordinates of the central units on the Nth layer can be obtained and the units' boundaries should be calculated by applying and zooming the formula (1) in literature (Ben, 2006) as well.

$$\begin{cases} x^O_{i,j} = x^O_{i^0,j^0}, & y^O_{i^0,j^0} = y_{i^0,j^0}, \\ x^A_{i,j} = x^F_{i^0,j^0+1}, & y^A_{i,j} = y^F_{i^0,j^0+1} \\ x^B_{i,j} = x^A_{i^0-1,j^0}, & y^B_{i,j} = y^A_{i^0-1,j^0} \\ x^C_{i,j} = x^B_{i^0-1,j^0-1}, & y^C_{i,j} = y^B_{i^0-1,j^0-1}; \quad (i,j) = \left(\dfrac{i_0-1}{2}, \dfrac{j_0-1}{2}\right) \\ x^D_{i,j} = x^C_{i^0,j^0-1}, & y^D_{i,j} = y^C_{i^0,j^0-1} \\ x^E_{i,j} = x^D_{i^0+1,j^0}, & y^E_{i,j} = y^D_{i^0+1,j^0} \\ x^F_{i,j} = x^E_{i^0+1,j^0+1}, & y^F_{i,j} = y^E_{i^0+1,j^0+1} \end{cases} \tag{2}$$

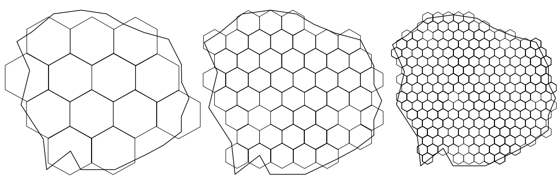


Figure 9. A sample of generating partial grids

In this paper, partial grid generation algorithms are investigated only, and they equal to the first hierachical grid generation instance. When operated practically, the subordination relationship of the six new gird units (apart form the central units) on the Nth layer are determined to estimate the spherical polygon the new units belong to. If it belongs to the spherical polygon, the unit is preserved and used for grid generation of higher layers, or delete the unit. The process doesn't stop until the desired partition layer is reached and Figure 9 is such an example (with the increasing of partition layers, grids' edge approximate gradually to the given region's edge). Finally, the units are projected to spheres by inverse Snyder equal-area projection (Snyder, 1992).

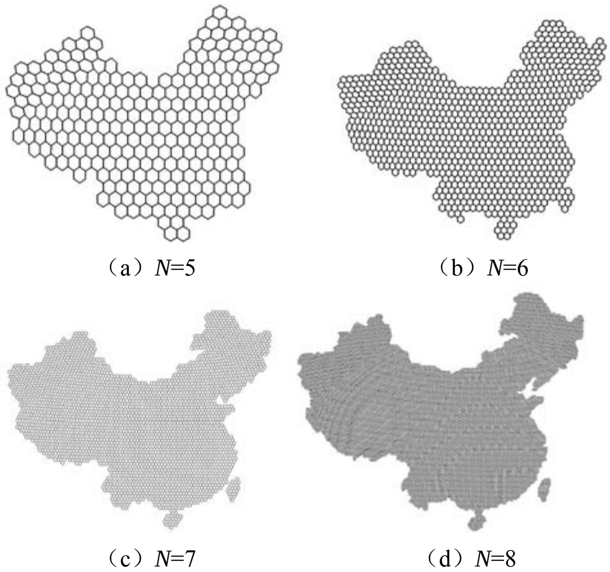
4. EXPERIMENTAL RESULTS AND ANALYSIS

Based on the partial geo-grid generation algorithm mentioned above, we make applications which can generate and display partial grids. Some constants can be declared as follow: sphere radius was arranged as the equal-area radius of WGS-84 reference ellipsoid, $R \approx 6371007.22347m$, and accordingly, the side length of an icosahedron is $L \approx 7674457.99928m$ (Ben, 2005). The experiment applies two kinds of data:

Data1. The vector boundaries of Chinese Mainland (588 discrete points altogether).

Data2. The vector boundaries in Taiwan Island (55 discrete points altogether).

As to data 1, the partition layer of partial grid is chosen as N=5,6,7,8, and final result shows in Figure 10. Figure 11 shows process of generation algorithm using partial grid data of Taiwan, from N=8 to N=10. Figure 12 illustrates the result of applying local grid data to the whole earth (Global Geo-Grid, partition layer N=7; the data of China, partition stage N=9; the data of Taiwan, partition stage N=14). Figure 13 shows hierarchical structure of the partial grids.



（a）N=5 （b）N=6

（c）N=7 （d）N=8

Figure 10. The different hierarchical partial grid of Chinese ground

The following experiments compare the efficiency of homogeneous hexagonal grid data generated by both hierarchical and unhierarchial algorithm. In these experiments, the boundary data of Chinese mainland is used, and the partition layer is N=5,6,7,8,9. Detailed computation efficiency of the experiments is listed in Table 1, and the initial value of N_0 (the number of partition layer) is 5. Figure 14 shows the efficiency curve of the two algorithms. Since the hierarchical algorithm is used here, only the newly generated units' centers have to be computed. In the process of multi-resolution grid data generation, grids on the previous $N_0 \rightarrow N$ layer have been generated. As can be seen from the boxed time in Table 1, the computation amount is decreased magnificently. The configuration of computers used in the experiments is 1.8GHZ CPU, 1024MB memory and Win2000+SP4 operating system.

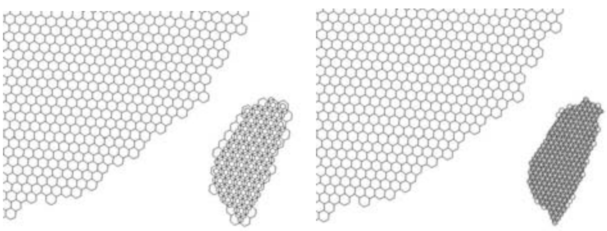


Figure 11. The course of generating partial grids

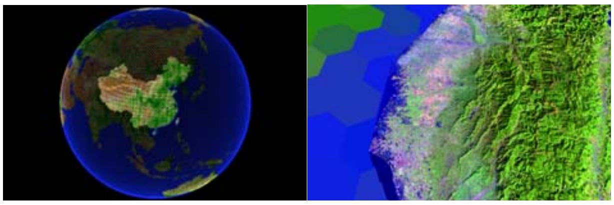


Figure 12. The demo of partial grids being in global grids

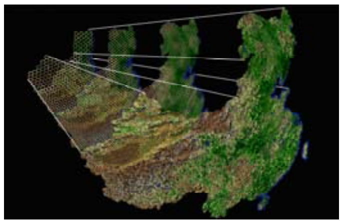


Figure 13. The hierarchical structure of partial grids

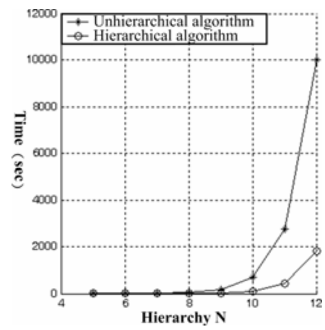


Figure 14. The efficiency curves of hierarchical and unhierarchical algorithm

Partition Layer *N*	**5**	**6**	**7**	**8**	**9**	**SUM**
The Number of Units	247	875	3277	12617	49016	66032
Unhierarchical Algorithm（Sec）	0.845	3.228	12.102	46.596	181.026	243.797
Hierarchical algorithm（Sec）	0.845	1.228	2.613	7.800	28.335	$\boxed{28.335}$

Table 1 Efficiency compare

5. CONCLUTION

The combination of Global Geo-Grid Systems and massive geo-information management is a fresh and evolving researching content. In this paper, the complex boundary problems of partial grids are analyzed in detail, the boundary simplification algorithms for spherical random polygons as well as the determination algorithms for the ubiety between points and spherical random polygons are also proposed. Additionally, the grid region cutting and the hierarchical generating algorithm for grid data are also proposed after considering the surface-striding problem. However, more research contents, including the construction of multi-resolution global geo-gird index system, the construction of girds' digital space and the application of geo-grids, remain to be investigated in the future.

ACKNOWLEDGEMENTS

The research was supported by the Projects of National Natural Science Foundation in China (40671163) and the Foundation of LREIS, CAS (A0612) and the Foundation of Institute of Surveying and Mapping (Y0702).

REFERENCES

A. Vince, 2006. Indexing the aperture 3 hexagonal discrete global grid. *Journal of Visual Communication and Image Representation*, 17, pp. 1227–1236.

Ben Jin, 2005. A Study of the Theory and Algorithms of Discrete Global Grid Data Model for Geospatial Information Management. *Ph D Thesis*, Zhenzhou: Information Engineering University, China.

Ben Jin, Tong Xiaochong, Zhang Yongsheng, 2006. A Generation Algorithm of Spherical Equal-Area Hexagonal Discrete Gird and Analysis of Its Deformation. *Geography and Geo-Information Science*, 22(1), pp. 7-11.

David F.Rogers, 2002. *Procedural Elements for Computer Graphics* (2^{th} ed.), China Machine Press, pp. 168-172.

Goodchild M. , 2000. Discrete global grids for digital earth. In: *International Conference on Discrete Global Grids*, California, Santa Barbara.

Sahr K, White D, and Kimerling A J, 2003. Geodesic Discrete Global Grid Systems. *Cartography and Geographic Information Science*, 30(2), pp. 121-134.

Snyder J P., 1992. An equal-area map projection for polyhedral globes. *Cartographica*, 29(1), pp. 10-21.

Tobler W R, and Chen Z, 1986. A quadtree for global information storage. *Geographical Analysis*, 18(4), pp. 360-71.

Tong Xiaochong, 2006. The Construct of Digital Space of Global Mul-resolution Grid System and the Study of Index Mechanism. *Ms D Thesis*, Zhenzhou: Information Engineering University, China.

White D, Kimerling A J, Sahr K, and Song L, 1998. Comparing area and shape distortion on polyhedralbased recursive partitions of the sphere. *International Journal of Geographical Information Science*, 12, pp. 805-827.

Zhang Yongsheng, Ben Jin, Tong Xiaochong, 2006. Geospatial Information Processing Method Based on Spherical Hexagon Grid System. *Journal of Zhengzhou Institute of Surveying and Mapping*, 23(2), pp. 110-114.

Zhao Xuesheng, Chen Jun, 2003. Fast Translating Algorithm between QTM Code and Longitude/Latitude Coordination. *Acta Geodaetica et Cartographica Sinica*, 32(3), pp. 272-277.