

## MODELING AND EXPRESSION OF VECTOR DATA IN THE HEXAGONAL DISCRETE GLOBAL GRID SYSTEM

Xiaochong Tong<sup>a,b,\*</sup>, Jin Ben<sup>a</sup>, Yuanyuan Liu<sup>c</sup>, Yongsheng Zhang<sup>a</sup>

<sup>a</sup> Zhengzhou Institute of Surveying and Mapping, No. 66, Longhai Middle Road, Zhengzhou, Henan Province, P.R. China.

<sup>b</sup> Key Laboratory of Environment Change and Natural Disasters (Ministry of Education), Beijing Normal University, No. 19, Xijiekouwai Street, Haidian District, Beijing, P.R. China.

<sup>c</sup> 61512 Troops, Beijing, P.R. China.

### Commission

**KEY WORDS:** Discrete Global Grid System, Hexagon, Vector Data, Modeling, Expression

### ABSTRACT:

The Discrete Global Grid System (DGGS) is a new type of global spatial data model and is the extension of the plane grid on a sphere. The hexagon is usually used in the construction of DGGS for its advantageous geometric structure. The paper principally focuses on the issue of modeling and expression of vector data in the hexagon DGGS. The precision of vector data is the basis of data recording and data expression, and data with different precision fall into the grid cells of corresponding sizes, making the gridding data themselves contain the precision and scale information. The present method of data recording is reserved, as far as possible, in the data recording process, and only the geometric information of vectors is substituted by the one-dimension coding of grids. This approach is more simple and effective than the digital coordinate recording method. The gridding expression of vector data differs from the traditional technique, mainly due to the subdivision of the durative space by grids as well as the obedience of the subdivision special rules, among which the point expression should activate the corresponding grid cells in the light of the point coordinates. Linear expression should activate the corresponding grid cells of every coordinate as well as the connected grids between every two node cells, and area expression should express both the boundary and internal regions by virtue of grid cells. For spherical expression, vector data have to solve the cell filling problem, but also the extension from planum to sphere. This paper puts forward a reasonable sphere extension approach, in which the vector data expression on the spherical grids was accomplished by the dismantling of vector data on different extended areas and the multi-times transformation. Besides, the algorithm in connection with the vector data was verified through experiments for its effect and efficiency. Moreover, the distance and direction of vector data on the grids would change in the mapping process from planum to sphere grids, leading to an inaccurate spherical gridding expression. So, the effects on the rectilinear direction in grids of the hexagon from the planum-sphere mapping process was investigated, and accuracy control of the spherical expression was processed to make sure that the drawing error of the spherical grids for vector data should be limited within one cell.

### 1. INTRODUCTION

The Discrete Global Grid System (DGGS), a new type of global spatial data model, divides the Earth into uniform sized grids and multiple layers, forming a hierarchy of seamless, non-piled, multi-resolution grids (Goodchild 2000, Sahr 2005, Kidd 2005). It also adopts the address codes of grid cells to replace the traditional geographic coordinates previously used for data operations (Zhang et al. 2007, Vince and Zheng 2009, Tong et al. 2013). DGGS is the new extension of the plane grid model on spheres. In particular, the DGGS which is based on the polyhedron subdivision, developed in recent years, is of great interest to the field (Dutton 1999, Szalay et al. 2005, Sahr 2011b, Peterson 2011). Among three regular geometric graphics (triangle, quadrangle and hexagon) which can subdivide the plane space, the hexagonal grid is the most compact and isotropic. It is the hexagonal grids excellent spatial attributes which make it very suitable for the modeling and processing of spatial data, and causes it to receive an increasing amount of attention (Middleton and Sivaswamy 2005, Sahr 2005, Sahr 2011a). The study of this paper is, therefore, based on the

hexagon DGGS.

Modeling and expression are the essential issues of spatial data, with regard to processing and application, and any operation must be accomplished within some digital space. This research mainly studies the modeling and expression of vector data in the hexagon DGGS. Data models included two aspects of data recording and data expression, in which data recording referred to the way of the stored records in the computer, such as coordinates information, attribute information, topological information, etc.; and data expression referred to the display form of the output device of the computer. For instance, vector data recording is about discrete serial coordinates of points while vector data expression is about variable data entity of consecutive points, lines and areas, etc.

### 2. CORRESPONDING GRID LAYERS AND DATA RECORDING WAYS OF VECTOR DATA

Vector data is the form of a model in relation to a spatial entity; it abstracts the spatial entity in the real world into spatial targets

\*Corresponding author: He received the PhD degree from Zhengzhou Institute of Surveying and Mapping in 2010, and then became a postdoctor of Beijing Normal University at Beijing, P.R. China. His research interests include discrete global grid system, photogrammetry and geographic information system. Email: txchr@163.com.

such as points, lines and areas, and those targets have certain spatial relationships. Vector differs from raster data in respect to the presence of points without sizes and the lines without width that are abstracted from entity by vector and this kind of abstracting is convenient for computer data storing and processing. However, points without sizes or lines without width factually do not exist in the real world. Thus, some researches showed that the expression of raster data and vector data in the real world is consistent: vector data is for the unlimited thinning grids; the grid size is for the error range of vector data, and this error range does not affect the spatial relation reasoning (Zhou et al. 2009, Zhao et al. 2007). This paper was accomplished based upon the vector data precision for data recording and data expression, and different data with different precision in the experiment was expressed with grids of different sizes. Then data fell into the corresponding grid cells, making the gridding expressing data contain precision information and scale information.

Vector topographic map data is common among the spatial vector data. According to the present Chinese topographic map framing criterion, eight national series basic scale-topographic maps (1:5000, 1:10000, 1:25000, 1:50000, 1:100000, 1:250000, 1:500000, 1:1000000) (Wang et al. 2006) and some other non-basic scale topographic maps are authorized. And those standard vector data usually contain precision information. For example, the vector data of scale is 1:10000, the map scale precision (Tian 1995) is about  $10000 \times 0.0001 = 1\text{m}$ , while, the vector data of scale is 1:1000000, the precision of map scale is  $1000000 \times 0.0001 = 100\text{m}$ . These precision information exist in any spatial data, and vector data that is produced by standards must have the precision assessment and explanation. For other vector data that really do not have the precision information, the digits of effective numbers on the coordinate can offer some help. In addition, the precision information of vector data can contribute to find suitable grids with corresponding sizes.

Taking the hexagon DGGS of an ideal icosahedron as example (the subdivision mode of hexagon is aperture 4 hexagonal class I (A4HCI) subdivision) (Kidd 2005, Zhang et al. 2007, Tong 2010, Tong et al. 2013), the projection mode of DGGS is Snyder Equal-Area polyhedral projection (Snyder 1992), this projection is used most in the construction of hexagon DGGS (Sahr 2005, Vince and Zheng 2009, Zhang et al. 2007)). On the  $n^{\text{th}}$  layer of DGGS with the A4HCI subdivision, there are  $45 \times 2^{2n-3} + 2$  cells (formula (2)). In Table 1, it records the cells area (area on the sphere, equal-area projection) on the discrete grid whose subdivision layer is  $n$ , the cells average radius  $D_n$  (the radius of each cell's equal-area spherical cap (Zhang et al. 2007)) and the average curvature error  $h_n$  (the approximation error between plane and sphere caused by Earth curvature in one cell (Zhao 2007)). The radius of Earth is 6371007.22347m.

Suppose that the scale precision of the vector data point is  $\rho$ , (vector point is regarded as a circle of uncertainty, point precision is the diameter and the corresponding area is  $S_\rho$ ):

$$\begin{cases} \frac{S_\rho}{H_i} \geq \frac{1}{2} \\ \frac{S_\rho}{H_{i-1}} < \frac{1}{2} \end{cases} \Rightarrow \frac{1}{2} H_i \leq S_\rho < \frac{1}{2} H_{i-1} \Rightarrow \sqrt{2} D_i \leq \rho < \sqrt{2} D_{i-1} \quad (1)$$

Corresponding data subdivision layer  $n=i$ . Where: the average area and average radius of the cell whose subdivision layer is  $i$  is considered from Table 1.

E.g. For a vector topographic map of 1:500000, the scale precision  $\rho \approx 500000 \times 0.0001 = 50\text{m}$ ,  $31.87138222837214 = \sqrt{2} D_{18} \leq \rho < \sqrt{2} D_{17} = 63.742764456249574$ . Thus, the subdivision layer should be the 18<sup>th</sup> layer of DGGS.

Layer	Average Area(km <sup>2</sup> )	Average Radius(km)	Average Curvature Error(m)
2	5544191.6145	1460.8081	169735.3625
3	1409021.0733	736.4323	42705.668
4	353720.9629	368.9814	10693.8838
5	88522.3236	184.5867	2674.5702
6	22136.3436	92.3054	668.7114
7	5534.4462	46.1542	167.1821
8	1383.6341	23.0773	41.7958
9	345.9099	11.5387	10.449
10	86.4776	5.7693	2.6122
11	21.6194	2.8847	0.6531
12	5.4048	1.4423	0.1633
13	1.3512	0.7212	0.0408
14	0.3378	0.3606	0.0102
15	0.0845	0.1803	0.0026
Layer	Average Area(m <sup>2</sup> )	Average Radius(m)	Average Curvature Error(m)
16	21112.6949	90.1459	0.000638
17	5278.1737	45.0729	0.000159
18	1319.5434	22.5365	0.000040
19	329.8859	11.2682	0.000010
20	82.4715	5.6341	0.000002
21	20.6179	2.8171	0
22	5.1545	1.4085	0
23	1.2886	0.7043	0
24	0.3222	0.3521	0
25	0.0805	0.1761	0
26	0.0201	0.088	0
27	0.005	0.044	0
28	0.0013	0.022	0
29	0.0003	0.011	0

Table 1. The average area, average radius and average curvature error of cells on A4HCI hexagon DGGS.

Discussion about data models is divided into two aspects: data recording and data expression. Firstly, data recording is considered. Through the research and experiment, it can be seen that the point coordinate can be expressed by the cell coding of the corresponding grid subdivision layer under the spherical discrete grid framework. Analysis of the research showed that the grid coding HQBS (Hexagonal Quaternary Balanced Structure) method (single code element is {0, 1, 2, 3} (2Bit)) (Tong et al. 2013) is much smaller than the traditional method of geographic coordinate system pattern (x, y). For instance:

- (1) Coding: 03-2010302010102  
 Layer: 13 Cell radius  $\approx 700\text{m}$   
 Coding size: 4.25Bytes  
 Geographic coordinate size (32bits double type): 8Bytes
- (2) Coding: 03-2010302010102020  
 Layer: 16 Cell radius  $\approx 90\text{m}$   
 Coding size: 5Bytes  
 Geographic coordinate size (32bits double type): 8Bytes
- (3) Coding: 03-20103020101020203020111  
 Layer: 23 Cell radius  $\approx 0.1\text{m}$   
 Coding size: 6.75Bytes  
 Geographic coordinate size (64bits double type): 16Bytes
- (4) Coding: 03-201030201010203020111002203  
 Layer: 29 Cell radius  $\approx 0.01\text{m}$   
 Coding size: 8.25Bytes  
 Geographic coordinate size (64bits double type): 16Bytes

From the above, it can be seen that the grid coding record and expression for the point coordinates is much more effective and simple than the digital coordinate pattern. In the concept model, one point coding not only offers the compressed format of a coordinate store for the mass data, which can realize the multi-resolution operation and nearby operation of the data, but also achieves the distance measurement in the grids. Other systems definitely record the spherical coordinates of points while the grid coding system hides the point position into the grid coding. And the point-entity grid record is similar both in plane and sphere.

There are three types of information in data record: geometric information, attribute information and topological information. The geometric information is the basic and essential spatial information for all the vector information, and it consists of point coordinates. The basic procedures of data record are: 1) substitution of coordinate points in the geometric information by cell coding serial; 2) attribute information reservation; 3) topological information reservation.

Vector data record patterns on the discrete grid differ from the traditional one in the aspects of coordinate record only, and other aspects are all equivalent. The exception is the data expression, as grids subdivide the continuous space and in this way the data expression have to obey the spatial principles of subdivision. The following two sections discuss the data expression of the point, the line and the area models on the hexagon grids, and then expand it to the whole sphere.

### 3. VECTOR EXPRESSIONS ON PLANAR GRIDS

The point entity is defined as the simplest data type among the vector data; it consists of a single coordinate or latitude and longitude. In the discrete grid framework, the point expression is just the cell expression, and what should be done is to activate the corresponding cell according to the point coordinate precision.

#### 3.1 Linear vector expressions on the planar grid

The line entity is described as a kind of chain entity consisting of a series of ordered coordinate values or latitude and longitude. The corresponding grid cell of every coordinate point is activated in the expression process, and every point should be connected with straight lines due to the serial attribute of the line entity. Therefore, for the line entity expression in the discrete grid space, the linkage cells of every two node cells should be activated by the linear filling method, forming continuous lines. This process is similar with the linear filling in the Raster Scanning Graphics (David 2002b).

One cell is set as the origin of coordinate  $O$ , the Tilted 120° Coordinate System  $O-IJ$  is established ( $OI$  axis is to the level-right, counter-clockwise for 120° for the  $OJ$  axis) (Middleton and Sivaswamy, 2005). Every two adjacent node cells coding of line entity is designed as  $G_0$  and  $G_S$ , and the codes are converted into Titled 120° Coordinates  $G_0=(I_0, J_0)$ ,  $G_S=(I_S, J_S)$  according to the algorithm in bibliography (Tong 2010, Tong et al. 2013). Table 2 gives the linear vector filling algorithm on the planar hexagon grid.

Step 1: Input variables:  $I_0, J_0, I_S, J_S$ ; initializing variable:  $\Delta I = \text{abs}(I_S - I_0)$  and  $\Delta J = \text{abs}(J_S - J_0)$ ,  $S_j = \text{Sign}(I_S - I_0)$ ,  $S_j = \text{Sign}(J_S - J_0)$ ; initializing loop variable:  $k = 0$ , the present activated cell coordinate  $i = I_0, j = J_0$ ;  
 Step 2: if  $S_j = S_j$ : go to Step 3; else: {initializing variable  $d = -\Delta I - \Delta J$ ,

go to Step 6};  
 Step 3: if  $\Delta I \geq \Delta J$ : {initializing variable  $d = -\Delta I$ , go to Step 4}; else: {initializing variable  $d = -\Delta J$ , go to Step 5};  
 Step 4: if  $k < \Delta I$ : {  $d = d + 2\Delta J$ , if  $d < 0$ : { $i = i + S_j$ }; else: { $i = i + S_j, j = j + S_j, d = d - 2\Delta I$ },  $k = k + 1$ , activate cell  $(i, j)$ , go to Step 4;}; else: process is finished;  
 Step 5: if  $k < \Delta J$ : {  $d = d + 2\Delta I$ , if  $d < 0$ : { $j = j + S_j$ }; else: { $i = i + S_j, j = j + S_j, d = d - 2\Delta J$ },  $k = k + 1$ , activate cell  $(i, j)$ , go to Step 5;}; else: process is finished;  
 Step 6: if  $k > -\Delta I - \Delta J$ : { if  $d < 0$ : { $i = i + S_j, d = d + 2\Delta J$ }; else: { $i = i + S_j, j = j + S_j, d = d - 2\Delta I$ },  $k = k - 1$ , activate cell  $(i + j, j)$ , go to Step 6;}; else: process is finished.

Table 2. Linear vector filling algorithm on the planar hexagon grids.

The linear generation algorithm between any two hexagon cells in Table 2 is the basis of linear entity expression algorithm in the discrete grid, and this algorithm can calculate the straight line filling-cell between any two cells; in this way, any linear entity can be ascertained by the connection of both sides of the fold lines. Through the analysis of this algorithm, it can be seen that every straight line generation of the linear entity is not related with one another, and it is a typical parallelization algorithm (Wilkinson and Allen 2005). Based upon the above discussion, any linear entity in the form of fold lines can be ascertained by the parallelized entity-filling algorithm, and cells of straight lines that are formed by any two nodes can be parallel processed.

#### 3.2 Area vector expressions in the planar grid

Area entity is founded on the basis of linear entity; it is a kind of geometric polygon that consists of a serial of sealed-internal boundary lines. Area entity as well as the boundary and internal regions is expressed in the discrete grid, and this expression method has the same precision and higher superiority on the spatial relation reasoning than that of the traditional method that merely consists of simple boundary lines of a serial of vertexes (Zhao et al. 2007).

This research principally modified some problems in the arbitrary polygon and level boundary, and further expanded to the hexagon grid based upon the rectangle grid polygon-region edge flag filling algorithm (David 2002b), as well as designed the area vector filling algorithm of the planar hexagon grid.

Step 1: All the nodes of the edges of the polygon were input according to the edges sequence  $(I_n, J_n), n \in [0, N - 1], n \in Z$ , the range of the bounding box of the polygon was ascertained:  
 $I_{min} = \min\{I_n\}, I_{max} = \max\{I_n\}, J_{min} = \min\{J_n\}, J_{max} = \max\{J_n\}$ ,  
 initializing all the elements of the variable array:  
 $B_{I_{max}-I_{min}+1, J_{max}-J_{min}+1} = 0$ , initializing loop variable  $n = 0$ ;  
 Step 2: (1) Nodes confirmation. if  $n \leq N - 1$ : {initialize  $F = \text{false}$ , go to Step 3}; else: { $i = 0$ , go to Step 5};  
 Step 3:  $k = 1$ , whiel  $J_n = J_{n-k}$ : { $k = k + 1, F = \text{true}$ }, if  $J_n > J_{n-k}$ :  
 $Logic_L = \text{false}$ ; else:  $Logic_L = \text{true}$ ;  $k = 1$ , whiel  $J_n = J_{n+k}$ :  
 $\{k = k + 1, F = \text{true}\}$ , if  $J_n > J_{n+k}$ :  $Logic_R = \text{false}$ ; else:  
 $Logic_R = \text{true}$ ;  
 Step 4: if  $Logic_L = Logic_R$ : { $B_{n+I_{min}-1, J_n+J_{min}-1} = 2, n = n + 1$ }; else:  
 {if  $F$ : { $B_{n+I_{min}-1, J_n+J_{min}-1} = 2, B_{n+I_{min}-1, J_n+1+J_{min}-1} = 1,$   
 $n = n + 2$ }; else: { $B_{n+I_{min}-1, J_n+J_{min}-1} = 1, n = n + 1$ }};  
 Step 5: (2) Contour line drawing. if  $n \leq N - 1$ : {invoke the algorithm in Table 2 (algorithm Step 4,5,6, internal regions if and else were all added with: if  $j = \Delta Y$ : break; else:  $B_{i+x_{min}-1, j+y_{min}-1} = B_{i+x_{min}-1, j+y_{min}-1} + 1$ . Remove the “activate cell  $(i, j)$ ”); else:  
 $\{n = 0, \text{go to Step 6}\}$ ;  
 Step 6: (3) Grid filling. if  $n < \Delta J$ : {initializing variable  $Logic = \text{false}$ ,  
 $j = 0$ , whiel  $j < \Delta I$ : {if  $Logic$ : activate cell  $(j, i)$ ; if  
 $B_{j+I_{min}-1, i+J_{min}-1} \neq 0$ : { $B_{j+I_{min}-1, i+J_{min}-1} = B_{j+I_{min}-1, i+J_{min}-1} -$



Figure 2(a) showed that the straight lines connected from any point in 00 area to other areas would not pass through the fractured zone, which was distinct from the unfolding pattern in Figure 1. And in this case, planum grid filling patterns can be considered directly for the vector expression. The key issue is the transformation between the record pattern of the triangle facet coordinate and the record pattern of the uniform coordinate, and operations of translation and rotation were involved in the transformation process. Table 4 gave the translation parameters  $(\Delta I, \Delta J)_{n,P}$  and rotation parameters  $\theta_P$ . For the case in which subdivision layer  $n > 2$ , translation parameter was changed into  $(\Delta I, \Delta J)_{n,P} = 2^{n-2} \cdot (\Delta I, \Delta J)_{2,P}$ . However, the rotation parameter did not change.

Facet No.	00	01	02	03	04
Translation	(0,0)	(1,2)	(1,-1)	(0,-3)	(3,0)
Rotation	0°	0°	0°	0°	0°
Facet No.	05	06	07	08	09
Translation	(3,3)	(4,-1)	(1,-4)	(6,0)	(4,5)
Rotation	60°	0°	-60°	60°	120°
Facet No.	10	11	12	13	14
Translation	(7,-1)	(0,-6)	(-6,0)	(1,5)	(-5,-1)
Rotation	0°	-120°	0°	-120°	0°
Facet No.	15	16	17	18	19
Translation	(-5,-4)	(-3,0)	(0,3)	(-2,-1)	(-3,-3)
Rotation	60°	0°	-60°	0°	0°

Table 4. Subdivision layer  $n=2$ , 00 area was taken as the center for the icosahedron unfolding, the translation parameters  $(\Delta I, \Delta J)_{n,P}$  and rotation parameters  $\theta_P$  from the local coordinate system on the triangle facet to the global coordinate system on the icosahedron unfolding area.

Transformation from the local coordinate system  $(P, I, J)$  on the triangle facet to the global coordinate system  $(I, J)$  on the icosahedron unfolding area in which 00 area was taken to be the center as following:

$$\begin{bmatrix} I \\ J \end{bmatrix} = \begin{bmatrix} \cos \theta_P & -\sin \theta_P \\ \sin \theta_P & \cos \theta_P \end{bmatrix} \cdot \begin{bmatrix} I \\ J \end{bmatrix}_P + \begin{bmatrix} \Delta I \\ \Delta J \end{bmatrix}_{n,P} \quad (3)$$

Inverse transformation:

$$\begin{bmatrix} I \\ J \end{bmatrix}_P = \begin{bmatrix} \cos \theta_P & \sin \theta_P \\ -\sin \theta_P & \cos \theta_P \end{bmatrix} \cdot \left( \begin{bmatrix} I \\ J \end{bmatrix} - \begin{bmatrix} \Delta I \\ \Delta J \end{bmatrix}_{n,P} \right) \quad (4)$$

The end-result relationship of the cells in the global coordinate system and the triangle facet should be judged for a proper triangle facet  $P$ , and was used in formula (4).

The attribute of the Titled Coordinate System could ascertain the triangle facet in which the cell lied by virtue of the relative relationship between the cell and three boundaries. Figure 2(a) indicated that the effective straight lines within the triangle facet were 13 with 3 patterns. They are:

$$\begin{cases} I = a_n: I = -2^n, I = -2^{n-2}, I = 2^{n-1}, I = 5 \cdot 2^{n-2} \\ J = b_n: J = -5 \cdot 2^{n-2}, J = -2^{n-1}, J = 2^{n-2}, J = 2^n \\ I - J = c_n: I - J = -5 \cdot 2^{n-2}, I - J = -2^{n-1}, \\ I - J = 2^{n-2}, I - J = 2^n, I - J = 7 \cdot 2^{n-2} \end{cases} \quad (5)$$

The range line in formula (5) can effectively illustrate the relationship between the triangle facet and cells on the extended facet of icosahedron. Besides, the open and close intervals were also taken into consideration. Formula (6) gave the grid (shown in Figure 2(a)) of any layer  $n$  and the judgment of the end-result for cell  $(I, J)$  in the triangle facet, and value of the open and close attribute parameters was shown in Table 5.

Facet No.	00	01	02	03	04	05	06	07	08	09
$\alpha$	0	1	1	0	0	0	1	0	0	1
$\beta$	0	-1	-1	0	0	1	-1	-1	0	0
$\gamma$	-1	0	0	-1	-1	-1	0	0	-1	1
Facet No.	10	11	12	13	14	15	16	17	18	19
$\alpha$	1	-1	0	0	1	1	0	-1	1	0
$\beta$	-1	0	0	-1	-1	0	0	0	-1	0
$\gamma$	0	0	-1	1	0	0	-1	-1	0	-1

Table 5. 00 area was taken as the center for the icosahedron unfolding, the value of the open and close attribute parameters  $\alpha, \beta, \gamma$

$$(I, J) \text{ is in the down triangle facet} \Leftrightarrow \begin{cases} I > a_n + \alpha \\ J < b_n + \beta \\ J > I - c_n + \gamma \end{cases} \quad (6)$$

$$(I, J) \text{ is in the up triangle facet} \Leftrightarrow \begin{cases} I < a_n' + \alpha \\ J > b_n' + \beta \\ J < I - c_n' + \gamma \end{cases}$$

Cells that do not belong to any triangle facet are North Pole and South Pole cells. And coordinates of the Titled Coordinate System of North Pole and South Pole are  $(2^{n-1}, 2^n)$ ,  $(-2^{n-2}, -5 \cdot 2^{n-2})$  respectively when the grid layer  $n \geq 2$ .

An arrangement of the 13 effective straight lines can reduce the calculation complexity in the judgment process in cases where the cell was in the triangle facet. This can avoid checking the 20 triangle facets one by one.

Next the drawing of straight lines between node cells  $(P_i, I_i, J_i)$  and  $(P_{i+1}, I_{i+1}, J_{i+1})$  was considered. When  $P_i = 00$ , the technique outlined above satisfied the requirements; when  $P_i \neq 00$  but still was the down triangle facet, it needed to consider the extension pattern in Figure 2(a) about the relationship between other triangle facets and the center facets  $P_i$ . The adjacent facet number in the three directions on arbitrary triangle facet of icosahedron extended surface was given as Table 6.

Facet No.	00	01	02	03	04	05	06	07	08	09
1	01	00	03	02	05	09	07	11	09	05
2	18	05	04	19	02	01	08	06	06	08
3	02	17	00	07	06	04	04	03	10	13
Facet No.	10	11	12	13	14	15	16	17	18	19
1	11	07	13	17	15	11	17	13	19	18
2	12	15	10	09	16	19	14	16	00	15
3	08	10	14	12	12	14	18	01	16	03

Table 6. 00 area was taken as the center for the icosahedron unfolding, the adjacent facet number in the three directions on arbitrary triangle facet

Where: For down triangle facet: vector  $\overline{(00,0,0)(01,0,0)}$  was 1 direction,  $\overline{(00,0,0)(18,0,0)}$  was 2 directions,  $\overline{(00,0,0)(02,0,0)}$  was 3 directions; for up triangle facet:  $\overline{(02,0,0)(03,0,0)}$  was 1 direction,  $\overline{(02,0,0)(04,0,0)}$  was 2 directions,  $\overline{(02,0,0)(00,0,0)}$  was 3 directions.

The shortest distance between arbitrary two triangle facets can be found by virtue of the adjacent facet information in the Table 6. Tong (2010) gave the algorithm of the shortest distance between arbitrary two triangle facet  $P_a$  and  $P_b$  of icosahedron based on the Dijkstra algorithm (Kenneth 2003). Table 7 was deduced by the algorithm and it showed the shortest distance from 00 facet to all the triangle facets expressed by directions 1, 2, 3.

Facet No.	01	02	03	04	05	06	07	08	09
Minimum route	1	3	3	3	1	3	3	3	1
			1	2	2		2	1	2
						3	3	3	1
								2	

Facet No.	10	11	12	13	14	15	16	17	18
Minimum route	3	3	2	1	2	2	2	1	2
	2	1	3	3	3	1	3	3	
	3	3	2	1	2	2			
	2	1	3						
	3								

Table 7. The shortest distance from 00 facet to all the triangle facets.

The shortest distance of arbitrary down triangle facets to other triangle facets can be matched with proper routes in Table 7, and in this way,  $P_b$  can match along with certain kind of triangle facet on the unfolding 00 facet. And then, all the coordinate transformations from  $(P_i, I_i, J_i)$  to  $(P_{i+1}, I_{i+1}, J_{i+1})$  can be accomplished by the algorithm in formula (3) and (4).

Table 8 showed the basic procedures of the algorithm generation for arbitrary linear entity  $P_0G_0, P_1G_1, \dots, P_NG_N$  on the icosahedron surface, among which  $P_iG_i \in$  the  $n^{\text{th}}$  spherical grid  $\mathbb{P}G_n$ .

- Step 1: Loop variable  $i = 0 \rightarrow N - 1$ , the straight lines generation from  $(P_i, I_i, J_i)$  to  $(P_{i+1}, I_{i+1}, J_{i+1})$ , go to Step 2;
- Step 2: if  $P_i = P_{i+1}$ : {invoke algorithm in Table 2}; else: {the shortest path  $(P_i, P_{i+1}, \dots, P_{i,n-1}, P_{i+1})$  of  $P_i \rightarrow P_{i+1}$  was calculated by virtue of minimum path algorithm based on the adjacent triangle facet relationship in Table 6, and match along with routes in Table 7,  $P_i \leftrightarrow 00, P_{i,1} \leftrightarrow P_{00,1}, \dots, P_{i,n-1} \leftrightarrow P_{00,n-1}, P_{i+1} \leftrightarrow P_{i+1}'$  was got based upon the adjacent relationship in Table 6, go to Step 3};
- Step 3: Convert  $(P_{i+1}', I_{i+1}, J_{i+1})$  to the global coordinate system (formula (3)) where the center was  $P_i$ , and generate straight lines by the algorithm in Table 2, end-result situations of points on the straight lines were considered according to formula (6), transform points on the global coordinate system to the partial coordinate system in the light of end-result relationship by formula (4), go to Step 4;
- Step 4: Convert  $(P_{00,j}, I, J) \rightarrow (P_{i,j}, I, J)$ ,  $j = 0 \rightarrow n$  by the corresponding relationship of triangle facets in Step 2.

Table 8. The algorithm generation for arbitrary linear entity on the icosahedron surface.

Floating-point arithmetic was not involved in the whole process but the integer arithmetic, and the arithmetic efficiency can be ensured. Similarly, the line entity generation on the icosahedron was likewise the typical parallel algorithm, among which the straight line generation between arbitrary two nodes  $P_i$  and

$P_{i+1}$  was completely independent with high efficiency by parallel algorithms.

### 4.3 Cross-facet issue of area vector data

Linear data expression is the fundamental issue for the area vector data expression on the sphere grids. Nodes confirmation and contour line drawing of the area-filling algorithm, shown in Table 3, had no significant distinction no matter whether in the planum grid or on the icosahedron surface; the only difference was the last procedure of the line grid filling process. The core of the cross-facet issue is that the planum grid can extend without limitation while the icosahedron surface is closed, and the unfolding of icosahedron is limited regions with fractures, which would bring about some difficulties with regards to filling. Theoretically, the less the fractured zones are, the higher the area-filling efficiency is. Thus, the unfolding pattern of icosahedron was involved. This research adopted the unfolding pattern in Figure 2, which can reduce the fracture calculation time in the filling process, shown an example in Figure 3.

There are two filling patterns for the polygons of the icosahedron surface:

1) Partition method. Deleave the polygon regions to every triangle facet. The scanning filling algorithm can process smoothly because the triangle facet of icosahedron can be unfolded into planum. However, the nodes that need to be calculated are excessive.

2) Filling method. Fill the fractured zones of the icosahedron unfolding surface into closed zones as well as complementary hexagon cells for the direct polygon filling. And finally remove the fictitious cells of the fractured zones in the polygon regions, which form the polygon filling regions that only exist on the icosahedron surface, shown in Figure 4.

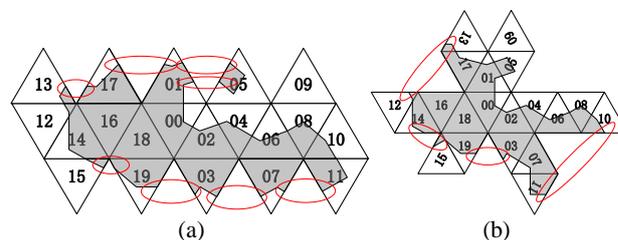
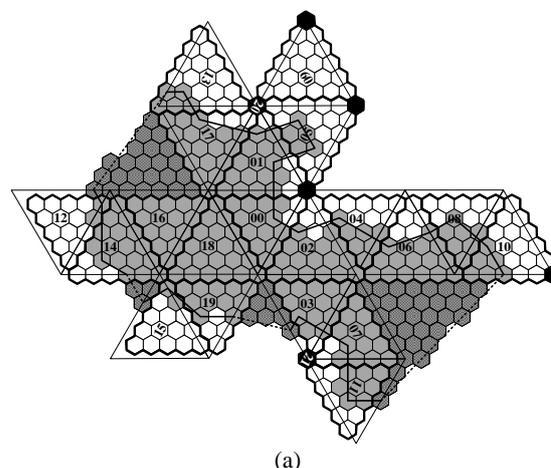


Figure 3. Fracture times comparison of the same area object between two different unfolding patterns. (a) Fracture processing operation for 8 times; (b) Fracture processing operation for 4 times.



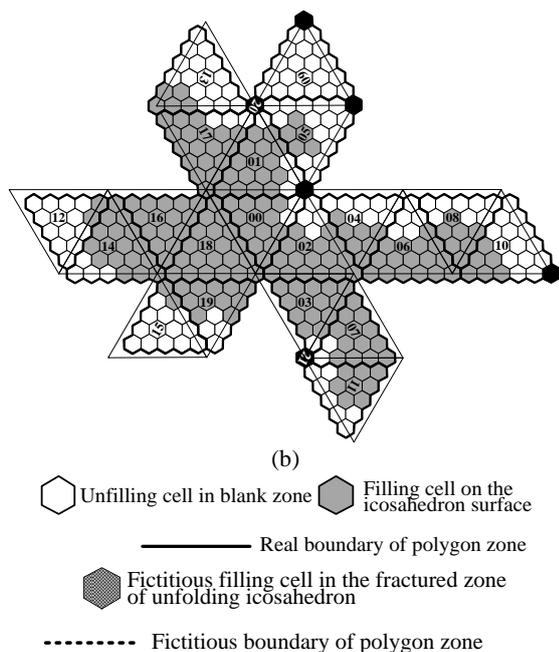


Figure 4. Polygon filling effect of the icosahedron surface. (a) Polygon filling by filling method; (b) Final filling project without fictitious cells.

Filling method can save the cross-facet calculation time as much as is possible, yet it doesn't mean that the cross-facet calculation is unnecessary. Take Figure 4(a) for example, on four occasions the cross-facet calculation is still required.

In the following we can see the solution to the cross-facet issue. Suppose that 00 facet is the center of the polygon region (Figure 2(a)),  $a$  direction of the  $P_i$  facet is the facet  $P_{i+1}$ , and  $b$  direction of the  $P_{i+1}$  facet is the facet  $P_i$ :

- (1)  $P_i$  and  $P_{i+1}$  are connected together with no fracture, if  $a = b$
  - (2)  $P_i$  and  $P_{i+1}$  are divided with fracture, if  $a \neq b$
- (7)

Cross-facet issues would be omitted when the first case in the above was satisfied and it would be processed when the second case was satisfied. The crossing point of the line segment  $P_i P_{i+1}$  and triangle boundary was calculated. One manner of confirmation is to compute the crossing point coordinate of  $P_i P_{i+1}$  and the borderlines of triangle facet, and then ascertain the grid cell coding by this coordinate, inserting the new cell into  $P_i G_i$  and  $P_{i+1} G_{i+1}$  as a new node prior to the calculation.

The basic procedures for the generation algorithm of arbitrary area entity  $S = P_0 G_0, P_1 G_1, \dots, P_N G_N, P_0 G_0$  of the icosahedron surface are given in the Table 9.

- 
- Step1: Calculate the triangle facet  $P_{mid}$  in which the gravity of the area entity  $S$  lied based on its vector boundary, and unfold the icosahedron in which the  $P_{mid}$  was the center;
- Step2: Transform all nodes  $P_i G_i$  into the global coordinate system where  $P_{mid}$  was the center (the same as the algorithm in Table 8), judge whether there is fracture between  $P_i$  and  $P_{i+1}$  according to formula (7). If so, go to Step3; if not, go to step 4;
- Step3: Calculate the crossing point of  $P_i P_{i+1}$  and fractured lines: transform points on  $P_{i+1}$  to the global coordinate system in which the  $P_i$  facet was the center, then calculate the crossing point of  $P_i P_{i+1}$  and lines in formula(5), then transform to the global coordinate system in which  $P_{mid}$  was the centre, and insert those two points (one point is in the  $P_i$  facet, another is in the  $P_{i+1}$  facet) into  $P_i$  and  $P_{i+1}$ . go to
- 

Step 4;

Step4: Fill regions by the method of Table 3, and transform the global coordinate system into area partial coordinate by formula(4), and the process of deleting the fictitious cells is in the transform process from global coordinate transformation to partial coordinate transformation.

---

Table 9. Generation algorithm of the area entity of the icosahedron surface

#### 4.4 The Accuracy Control in the Process of Vector Data Drawing in the Hexagon DGGS

The generation of DGGS based on a polyhedron involves the method of the polyhedron replacing the sphere, and there are relative errors when the plane replaces the curve. The grid on the polyhedron surface is accurate on direction and distance, but it will not be equal everywhere in the process of polyhedron mapping to a sphere. This leads to the direction and length of the spherical grid changing, and the greater the distance, the wider the gap.

Therefore, in the process of the grid expressing vector line data, the distance of any two adjacent nodes should not be too large. It needs to be confined to a certain range, which can guarantee the accuracy of data expression. For the large-spanned adjacent nodes in spherical vector data, some control nodes need to be interpolated to subdivide spherical line segments into many local lines, meeting grid expressing accuracy, in order to achieve an accurate vector data expression and provide a reliable basis for data measurement on a grid.

Consider the situation on one triangular facet of the spherical ideal icosahedron, the subdivision structures of A4HCI are represented respectively by  $a$ ,  $b$  and  $c$  the three line directions according to the order of grid arrangement (Figure 5). The arbitrary discrete line displayed by the filling mode on the hexagon grid consists of lines in these three directions. Thus it is the important, in terms of the line distortion on a spherical grid, to analyse the line distortion in three directions.

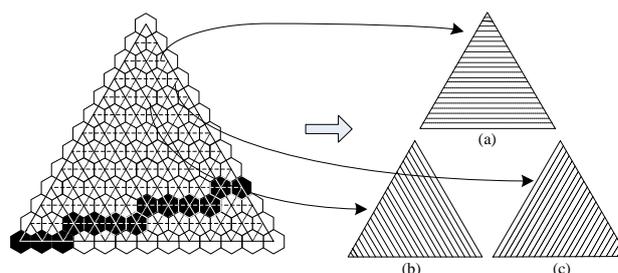


Figure 5. Three directions of lines arranged by hexagon grids on icosahedron triangular facet.

The following is to evaluate the line distortion by using numerical interpolation. The specific process is as follows:

- 1) Select separately isometric  $x_{i,j}y_{i,j}$  lines at  $a$ ,  $b$  and  $c$ , the three directions of plane triangle, which  $i = 1, 2, \dots, j = a, b, c$ ; The endpoints of the line segments all fall at the edges of the triangle;
- 2) Interpolate a series of control points  $P_1, P_2, \dots, P_k$  in the line segment  $x_{i,j}y_{i,j}$ , then convert  $P_1 \sim P_k$  into spherical points  $P^s_1, P^s_2, \dots, P^s_k$  through projection transformation;
- 3) Transform the endpoints of the line segment into spherical points  $X_{ij}$  and  $Y_{ij}$  by projection transformation, forming the spherical great circle arc  $\widehat{X_{i,j}Y_{i,j}}$ ;

4) Calculate the spherical distance  $\Delta h_{m,i,j}$  between each spherical point  $P_m^S$ ,  $m=1\sim k$  and great circle arc  $X_{i,j}Y_{i,j}$ , calculate respectively  $\Delta h_{\max,i,j}$  and  $\overline{\Delta h_{i,j}}$  by using the formula(8).

$$\Delta h_{\max,i,j} = \max_{m=1\sim k}(\Delta h_{m,i,j}), \overline{\Delta h_{i,j}} = \frac{1}{k} \sum_{m=1}^k (\Delta h_{m,i,j}) \quad (8)$$

$\Delta h_{\max,i,j}$  and  $\overline{\Delta h_{i,j}}$  can evaluate objectively the distortion situation that is caused by a projection system to a plane line. The two values can indicate the size of the line distortion caused by projection. Since the two values can also evaluate the line expression on the global grid,  $\overline{\Delta h_{i,j}}$  describing the average departure degree between line  $x_{i,j}y_{i,j}$  and spherical great circle arc; and  $\Delta h_{\max,i,j}$  describes the maximum departure degree. For the DGGs of any layer  $n$ , if the average radius  $D_n$  of the cell is superior to the maximum departure degree  $\Delta h_{\max,i,j}$  in all three directions, there will not exist any error (or the error is less than one cell) to generate the line due to the plane grid mode on the discrete global grid (in one triangular plane) of this layer.

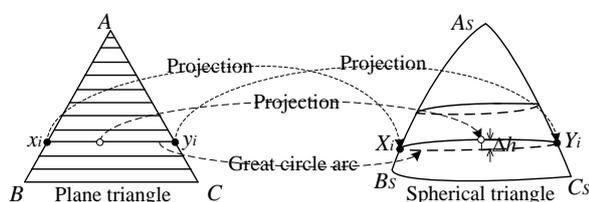


Figure 6. The mode of calculating the spherical line distortion by using projection method

The  $\Delta h_{\max,i,j}$  and  $\overline{\Delta h_{i,j}}$  in Figure 7 are produced through Snyder projection. It's possible to consider only the situation of direction  $a$ , as the Snyder projection is symmetric to the three medial axes of the triangular facet.

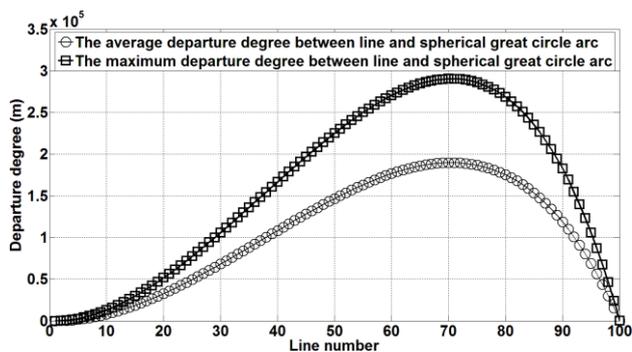


Figure 7. The average value  $\overline{\Delta h_{1,a}}$  and maximum value  $\Delta h_{\max,i,a}$  of the distance between projection points at each line and spherical great circle arc by using Snyder projection.

From the Snyder projection transformation in Figure 7, the maximum distortion error of the lines produced in one triangular plane is 290.529km, and the average distortion error is 189.801km. Compared with the average radius of cells in Table 1, the conservative approach is that the vector data directly generates the line without interpolation in the 4<sup>th</sup> layer of the discrete grid, and the eclectic approach is that the vector data directly generates the line without interpolation in the 5<sup>th</sup> layer of the discrete grid.

In one triangular facet of the  $n=4$  layer grid, it can meet the requirement of direct line generation. In this layer of the grid, the longest line is not more than  $3 \times 2^{4-2}=12$  cells (A4HCI grid is arranged closely on the icosahedron, with the maximum cell number in one direction on each triangular facet (Zhang *et al.* 2007, Tong 2010, Tong *et al.* 2013)). In one triangular facet of the  $n=5$  layer grid, the longest arranged cell is  $3 \times 2^{5-2}=24$ . Since the line projection distortion will never change, the aperture of the A4HCI grid structure is 4, and the average radius is half of the 4<sup>th</sup> layer. The line that is expressed accurately in the 5<sup>th</sup> layer will be no more than  $24/2=12$  cells (using one of the arranging modes at  $a$ ,  $b$  and  $c$  the three directions, Figure 5 is the result of the line arranging 13 cells in the  $a$  direction). In the same way, for the grid at the  $n^{\text{th}}$  layer, there are  $3 \times 2^{n-2}$  arranging cells whose cell radius average is the  $1/2^{n-4}=2^{4-n}$  of the 4<sup>th</sup> layer in one triangular facet, then the longest line than can be expressed in the  $n^{\text{th}}$  layer grid is not more than  $3 \times 2^{n-2} \times 2^{4-n}=12$  cells (use one of the arranging modes in the  $a$ ,  $b$  and  $c$  the three directions).

The above strategy refers to the most conservative line generation method under the constraint of the maximum projection distortion. In fact, the grid layer to which the average error of line projection corresponds can generate an accurate grid line. The average distortion error will be not more than the 5<sup>th</sup> layer if the Snyder projection transformation is adopted. According to the above analysis, it can be assumed that the line can be expressed accurately by the cells which are not more than  $3 \times 2^{5-2}=24$  on the  $n^{\text{th}}$  grid.

From the results in Table 1, when the subdivision layer  $n$  is more than 10, it can be thought that the subdivision of the grid will not be generated by projection but through the plane cell direct subdivision for the average curvature error. This is taken from the Earth's spherical curvature, which is less than 2.6122m. Therefore, when the two nodes of vector data both fall in the same cell of a discrete grid, the error taken from the Earth's curvature will not be considered when the grid line is generated directly. To sum up, if the spherical distance of any two points  $S_1, S_2$  in one triangular facet of spherical icosahedrons is  $d_s$ , the grid line generation between two points will obey the following rules:

- 1) When the grid subdivision layer is  $n \leq 5$ , it should directly generate grid lines regarding the triangular facet as a plane;
- 2) When grid subdivision layer  $n > 5$  and the distance between two endpoints of a line is  $d_s > 2 \times D_{10} = 11.5386\text{km}$ , the judging rules and methods of node interpolation are such that: when  $d_s/2D_n > 24$ ,  $d_s = d_s/2$ ; divide the spherical great circle arc  $S_1S_2$  into two equal parts, calculate the midpoint and interpolate new nodes; repeat this step until  $d_s/2D_n \leq 24$ , then end the interpolation nodes;
- 3) When the distance  $d_s$  between two endpoints of a line is  $d_s \leq 2 \times D_{10} = 11.5386\text{km}$ , the line is in the  $n^{\text{th}}$  ( $n > 10$ ) layer of one cell; as a plane grid, it can directly conduct grid line generation.

According to the above rules, only if  $d_s/2 \times D_n > 24$  in condition 2, is grid interpolation required in the process of line generation to ensure that the error of the vector line is limited in one cell of the current layer. The vector data used are usually formed by broken lines among several nodes. Not all the distances of any two nodes can satisfy the interpolation requirements of condition 2, so it is unnecessary to interpolate one by one.

## 5. EXPERIMENT AND ANALYSIS

The vector data filling effect and its efficiency with respect to the global discrete grid was processed by virtue of the following

spatial data: vector boundary data of the global mainland, 1875 area entities (633485 point coordinates), data amount of 9.90MB; Chinese county level administrative regions vector data, 3408 area entities (1123947 point coordinates), data amount of 17.3MB; Zhengzhou region vector data, include 5229 area entities, 806 line entities and 224 point entities, data amount of 16.2MB.

The grid layer  $n=9\sim 15$ , and two different kinds of vector data were loaded respectively. Line filling and area filling were processed with the same vector entity in the experiment and the calculation time was got in Table 10, among which the line-filling cell number refer to the activated cell numbers in the filling process; the area-filling cell number refers to the activated cell number in the filling process. Figure 8 showed a comparison between the activated cell number and activated efficiency in the filing process with vector data of different layers and types. Figure 9 showed the effects of partial layers and types. Analysis of Figure 10 showed the following:

- 1) The needed activated cell number increased with a stable rate when the grid layers increased, no matter whether the area-filling or the line-filing was adopted;
- 2) Seen from vector data, with the same coordinate points, the efficiency of area-filing is higher than that of line-filing. The reason being that every activated cell needed to adopt the line scanning algorithm for a calculation in the line entity filling process; while activated cells adopted row scanning algorithm, only the endpoints of every row were scanned, and cells in the middle were filled directly;
- 3) The experiments found that the bigger the expression range is, the less the filling numbers in unit time, and the lower the efficiency. That is because the cross-facet calculation time increased when the range became bigger, so the efficiency became lower too. Besides, only 5 triangle facets were crossed in the vector data of China, while 20 triangles facets were crossed in the global range.

Experimental Environment: ThinkPad T61, CPU Intel(R) Core(TM)2 Duo, 0.98GB Memory, Windows XP Operating System, Visual C++ 2008.

Vector boundary in the global mainland						
layer	Liner filing cell No.	Time(ms)		Area filling cell No.	Time(ms)	
		Total time	cells/ms		Total time	cells/ms
9	96806	56	1779	411793	160	2573
10	221443	125	1770	1651882	646	2575
11	510575	289	1763	6569058	2546	2568
12	1184196	682	1736	26535265	10348	2562
13	2689847	1546	1739	105483896	41109	2563
14	6076418	3489	1741	423526019	165067	2561
15	13815339	7947	1738	1708956559	665994	2560

Vector boundary in county-level administrative regions						
layer	Liner filing cell No.	Time(ms)		Area filling cell No.	Time(ms)	
		Total time	cells/ms		Total time	cells/ms
9	9301	2	4650	26173	4	6543
10	21337	5	3879	104991	19	5526
11	49077	13	3775	417520	77	5422
12	113048	31	3647	1686544	314	5371
13	260548	72	3644	6704409	1249	5368
14	600656	155	3640	26918723	5013	5370
15	1384895	380	3644	108618897	20231	5369

Table 10. Efficiency comparisons of line filling and area filling of vector data

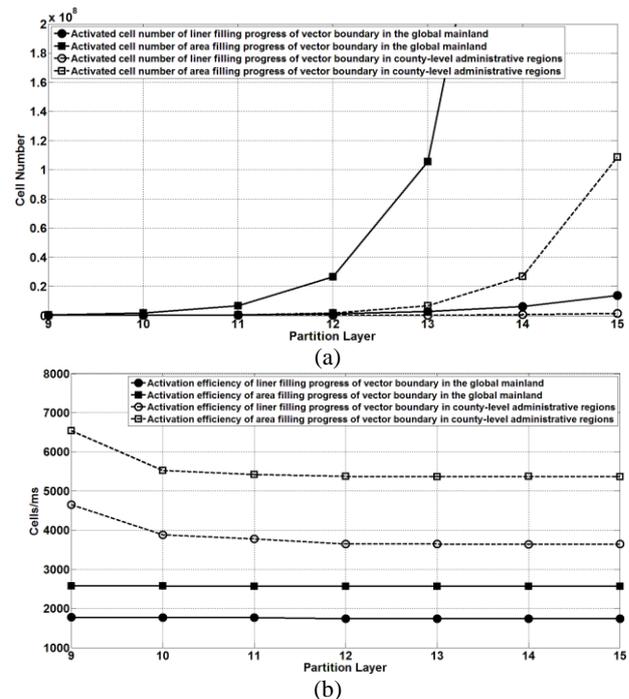
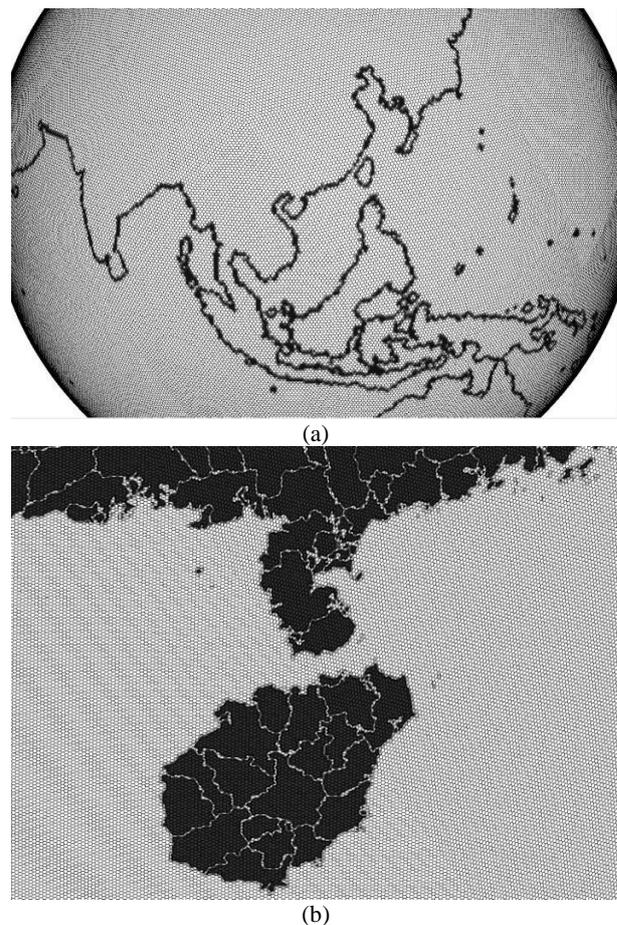
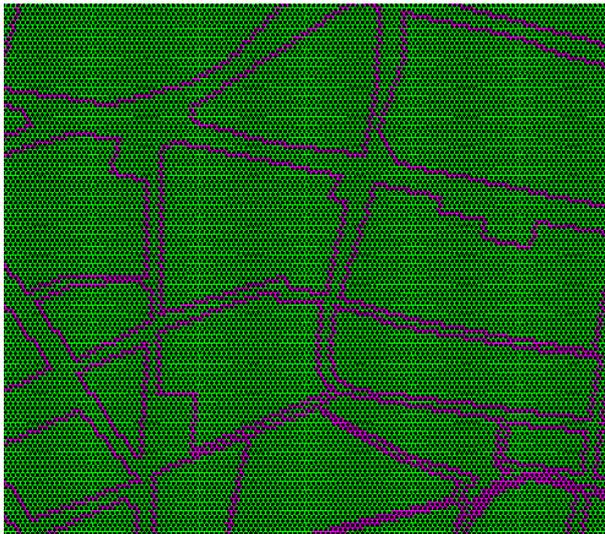


Figure 8. Activated cell number and activation efficiency of vector data of different layers and types in the display process of DGGs. (a) Comparison of activated cell number; (b) Comparison of activated cell efficiency.





(c)

Figure 9. Display effects of different types of vector data in the hexagonal DGGs.

## 6. CONCLUSION

The Discrete Global Grid System (DGGs), a new type of global spatial data model, divides the Earth into uniform sized grids and multiple layers, forming a hierarchy of seamless, non-piled, multi-resolution grids. In the modeling and expression of DGGs data, the vector data is difficult, because there are vital differences. One difference is that the grid is discretely formed in space but the vector is continuous. Thus the relation of essential differentiation leads to the difficulty of the overlapping display of vector and grid; and the ability of grid isomorphism using spatial data cannot be exerted.

The paper principally focus on the issue of modeling and expression of vector data in the hexagon DGGs. The precision of vector data is the basis of data recording and data expression, and different data with different precision are illustrated by grids of different sizes of expression. Data with different precision fall into the corresponding grid cells, making the gridding data themselves contain the precision and scale information. The present method of data recording is reserved, as far as possible, in the data recording process, and only the geometric information of vectors is substituted by the one-dimension coding of grids. This approach is more simple and effective than the digital coordinate recording method.

The gridding expression of vector data differs from the traditional technique, mainly due to the subdivision of the durative space by grids as well as the obedience of the subdivision special rules, among which the point expression should activate the corresponding grid cells in the light of the point coordinates. Linear expression should activate the corresponding grid cells of every coordinate as well as the connected grids between every two node cells, and area expression should express both the boundary and internal regions by virtue of grid cells. For spherical expression, vector data have to solve the cell filling problem, but also the extension from planum to sphere. As a result, this paper puts forward a reasonable sphere extension approach, in which the vector data expression on the spherical grids was accomplished by the dismantling of vector data on different extended areas and the multi-times transformation. Besides, the algorithm in connection with the vector data was verified through experiments for its effect and efficiency.

Moreover, the distance and direction of vector data on the grids would change in the mapping process from planum to sphere grids, leading to an inaccurate spherical gridding expression. So, the effects on the rectilinear direction in grids of the hexagon from the planum-sphere mapping process was investigated, and accuracy control of the spherical expression was processed to make sure that the drawing error of the spherical grids for vector data should be limited within one cell. And finally, this research may pave the way for establishing a theoretical basis on the high-accuracy demonstration of gridding vector data and the establishment of grid space measurement.

## ACKNOWLEDGEMENTS

This work is supported by the National Science Foundation of China (NSFC) under cooperative 41201392, 41271391 and 40930104.

## REFERENCES

- David, A., Todd, D., and Ross, P., 2002a. Climate modeling with spherical geodesic grids. *Computing In Science & Engineering*, Sep/Oct, pp. 32-41.
- David, F., R., 2002b. *Procedural Elements for Computer Graphics (Second Edition)*, McGraw-Hill, Inc.
- Dutton, G., 1999. *A hierarchical coordinate system for geoprocessing and cartography*. Berlin: Springer-Verlag.
- Goodchild, M.F., 2000. Discrete global grids for digital earth. *In: International Conference on Discrete Global Grids*, California: Santa Barbara.
- Kenneth, H. Rosen, 2003. *Discrete Mathematics and Its Applications (Fifth Edition)*, McGraw-Hill, Inc.
- Kidd, R., 2005. NWP Discrete Global Grid Systems. *In: Vienna University of Technology, ed. ASCAT Soil Moisture Report Series*, No. 4, Austria: I.P.F TU Wien.
- Middleton, L. and Sivaswamy, J., 2005. *Hexagonal image processing*. London: Springer.
- Peterson, P., 2011. *Closed-Packed Uniformly Adjacent, Multi-resolution Overlapping Spatial Data Ordering*. United States Patent #0,8018,458, awarded 2011-09-13.
- Sahr, K., 2005. *Discrete Global Grid Systems: A New Class of geospatial data structures*. Thesis (PhD). University of Oregon.
- Sahr, K., 2011a. Hexagonal discrete global grid system for geospatial computing. *Archives of Photogrammetry, Cartography and Remote Sensing*, 22, pp. 363-376.
- Sahr, K., 2011b. *Icosahedral modified generalized balanced ternary and aperture 3 hexagon tree*, United States Patent #07,876,967, awarded 2011-01-25.
- Snyder, J.P., 1992. An equal-area map projection for polyhedral globes. *Cartographica*. 29(1), pp. 10-21.
- Szalay, A.S., et al., 2005. Indexing the sphere with the hierarchical triangular mesh. *Microsoft Technical Report*, MSR-TR-2005-123, August 2005.
- Tian, Q., 1995. *An Introduction to Cartography*, Wuhan: China University of Geosciences Press.
- Tong, X., 2010. *The Principles and Methods of Discrete Global Grid Systems for Geospatial Information Subdivision Organization*. Thesis (PhD). Zhengzhou Institute of Surveying and Mapping.
- Tong, X., et al., 2013. Efficient encoding and spatial operation scheme for aperture 4 hexagonal discrete global grid system, *International Journal of Geographical Information Science*, 27(5), pp. 898-921.

- Vince, A. and Zheng, X., 2009. Arithmetic and Fourier transform for the PYXIS multi-resolution digital Earth model, *International Journal of Digital Earth*, 2(1), pp. 59-79.
- Wang, J., Sun, Q., and Wang, G., 2006. *Principles and Methods of the Cartography*, Beijing: Science Press.
- Wilkinson, B., and Allen, M., 2005. *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers (Second Edition)*, Prentice Hall, Inc.
- Zhang, Y., Ben, J., and Tong, X., 2007. *Discrete Global Grids for Geospatial Information: Principles, Methods and its Applications*, Beijing: Science Press.
- Zhao, X., Hou, M. and Bai, J., 2007. *Spatial Digital Modeling of the Global Discrete Grids*, Beijing: Surveying and Mapping Press.
- Zhou, C., Ou, Y. and Ma, T., 2009. Progresses of Geographical Grid Systems Researches, *Progress in Geography*, 28(5), pp. 657-662.