A pole-oriented discrete global grid system: Quaternary quadrangle mesh

Mengyun Zhou, Jing Chen *, Jianya Gong

State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, 129 Luoyu Road, Wuhan, Hubei 430079, China

A R T I C L E   I N F O

Article history:
Received 17 April 2013
Received in revised form 3 August 2013
Accepted 29 August 2013
Available online 6 September 2013

Keywords:
Spherical partitioning
Polar semi-hexagon
Data indexing
Adjacent search
Geometrical distortion

A B S T R A C T

In addition to providing services for mid and low latitudes, global Geographic Information System (GIS) should provide services for high latitudes; these services include climate monitoring, energy exploitation in the polar regions. An improved Discrete Global Grid System (DGGS) could serve as a foundation for efficient indexing, visualization, and analysis of the ever-expanding global spatial data in the global GIS environment. However, existing DGGSs have problems with balancing polar and other regions, including serious cell area and shape distortion in the polar regions or precision loss when applying the systems to existing geographical data. Here, a new pole-oriented DGGS, the Quaternary Quadrangle Mesh (QQM), is proposed. This DGGS uses semi-hexagon (a type of quadrangle) grids in the polar regions and rectangular grids elsewhere. The semi-hexagonal partitioning in the polar regions reduces the redundancy of the polar data and avoids the polar singularities that frequently exist in DGGSs. A consistent encoding-decoding scheme and a uniform adjacent search algorithm were constructed by considering that polar cells and other cells form a coherent unity in the QQM, which has a hierarchical structure. The experimental results demonstrate that the QQM performs better than the recently proposed Degenerate Quadtree Grid (DQG) regarding geometrical distortion, and encoding-decoding and adjacent search efficiency. The QQM can satisfy the requirements for global data indexing and visualization, especially in the polar regions.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Schemes for tessellating the curved surface of the earth and for data indexing and visualization must improve to contend with the ever-expanding pool of global spatial data (Goodchild et al., 2012). An improved Discrete Global Grid System (DGGS) could serve as a foundation for efficiently indexing and visualizing the large pool of global spatial data in the global Geographic Information System (GIS) environment. In addition to providing services for mid and low latitudes, global GIS should provide services for high latitudes. These services include, for example, monitoring the impacts of climate change on polar glaciers and sea levels around the world, and exploiting the energy and ecology resource in polar regions. Thus, a DGGS with excellent data indexing and visualization properties in polar and other regions is needed.

DGGS researches strive to accurately express the global surface, efficiently index global data and smoothly visualize global data. However, currently, researchers have only achieved limited success when balancing polar and other regions. Most existing DGGSs are based on regular polyhedras or geographic coordinate systems. The latitude–longitude grid cell areas and shapes are severely distorted in the polar regions. Regular polyhedral DGGSs alleviate this distortion. However, most of their cell edges do not coincide with the meridians and parallels. Thus, polyhedral DGGSs are difficult to apply to some types of existing geographical data that are based on the geographic coordinate systems around the earth.

In this paper, a pole-oriented discrete global grid DGGS, referred to as the Quaternary Quadrangle Mesh (QQM), is proposed to address existing DGGS deficiencies. The QQM reduces cell distortion and data redundancy in the polar regions. All of the QQM grid cells are quadrangles, semi-hexagons (in the polar regions) or rectangles (elsewhere). The QQM is seamless and the polar grid edges match the non-polar grid edges. Quadrangular subdivisions, which can be represented as squares or rectangles on a map projection, are easier to manage with computers than other geometric shapes (Ottoson and Hauska, 2002). Data can be stored at different levels of resolution and all nodes can be manipulated by the same programs and algorithms in the QQM’s regular hierarchical structure (Tobler and Chen, 1986). Because multi-dimensional codes (level, row, column) degrade the system performance (Mokbel et al., 2003), they are mapped as 1-Dimensional codes in the QQM. The conversion between a QQM address code and its geographic coordinates is...
simple, which is advantageous for using existing spatial data. In addition, the QQM has simple adjacencies, which are the basis for efficient spatial processing (including indexing and range searching) (Gold and Mostafavi, 2000).

The partitioning method, the encoding and decoding scheme, and the adjacent search algorithm are presented in Section 3. The performance of the QQM is compared with the performance of the DQG in Section 4. In addition, the application of the QQM for organizing and visualizing global spatial data is provided in Section 5. Overall, the experimental results demonstrated the following: (1) the cell shapes and areas of the QQM are analogous; (2) the efficiencies of encoding, decoding, and adjacent searches in the QQM are greater than in the DQG; and (3) the visualization of global data based on the QQM is smooth.

2. Related work

Regular polyhedral DGGSs have been studied by several researchers. For example, the Spherical Quaternary Triangle (SQT) was proposed by Fekete and Treinish (1990); the Quaternary Triangular Mesh (QTM) was proposed by Dutton (1997); a hierarchical tessellation mesh for a WGS-84 ellipsoidal surface that was based on quaternary triangular meshes was proposed by Bai et al. (2011); recursive diamond subdivision grids were proposed by White (2000); and hexagonal DGGSs were used by several authors (Sahr, 2008; Tong et al., 2010; Vince, 2006; White et al., 1992). These regular polyhedral DGGSs have excellent properties, including a hierarchical structure, similar cell areas and shapes, and global addressability. However, mapping from a polyhedra to a sphere is a complex process because most cell edges do not coincide with meridians or parallels in the polyhedral DGGS. Thus, it is difficult to apply polyhedral DGGSs to existing spatial data that are based on the geographic coordinate system without a costly conversion process. However, this conversion process results in a loss of precision. Therefore, polyhedral DGGSs cannot satisfy the organization and updating requirements of global spatial data.

The traditional latitude–longitude grid system, includes a relationship between the cell edges and the geographic coordinate system and simple data storage and processing (Gregory et al., 2008). Thus, this DGGS has been widely used by researchers (Albergel et al., 2010; Fekete and Treinish, 1990; Lindstrom and Pascucci, 2001; Samet and Sivan, 1992) and agencies (such as the US Geological Survey, which provided the GTOPO30 data). However, the latitude–longitude grid cell areas become smaller towards the poles. In addition, the shape of the cells becomes increasingly distorted and turn into triangles at the poles. The North and South poles are points on the surface of the earth, but map to lines on the latitude–longitude plane (Sahr et al., 2003). Each of these polar singularities has an unique latitude but an uncertain longitude. The irregular cell areas and shapes decrease the manipulation accuracy and result in data redundancy. For example, the elevation of the poles is repeated 43,200 times in GTOPO30 (U.S. Geological Survey, 1999).

Attempts have been made to create adjusted latitude–longitude grid systems to obtain more consistent cell sizes. For example, by considering the curvature of the earth, Ottoson and Hauska (2002) proposed Ellipsoidal quadrilaterals (EQT) with latitudinal intervals that increase in polar regions and in the converging areas of the eastern and the western hemispheres.

Most of the adjusted latitude–longitude grid systems have complex initial division rules. Two separate index mechanisms must be implemented, one for the base level and one for lower levels (Ottoson and Hauska, 2002). No unified equation exists for converting between the latitude–longitude and address code, which complicates encoding and decoding because several different types of conversions must be implemented for the grids at different locations. These adjusted latitude–longitude grid systems are difficult to use for organizing and visualizing global multi-resolution data.

To address some of these difficulties, Sun et al. (2009) proposed the Degenerate Quadtree Grid (DQG) with longitudinal intervals that increase regularly from the equator to the poles and latitudinal intervals that remain the same. The longitudinal interval of one row in the DQG is half or twice as large as its adjacent row. The DQG has attractive properties, including a hierarchical structure, convergent geometrical distortion, and simple cell adjacencies. The DQG has solved the problem (to some extent) of large

---

2. Related work

Regular polyhedral DGGSs have been studied by several researchers. For example, the Spherical Quaternary Triangle (SQT) was proposed by Fekète and Treinish (1990); the Quaternary Triangular Mesh (QTM) was proposed by Dutton (1997); a hierarchical tessellation mesh for a WGS-84 ellipsoidal surface that was based on quaternary triangular meshes was proposed by Bai et al. (2011); recursive diamond subdivision grids were proposed by White (2000); and hexagonal DGGSs were used by several authors (Sahr, 2008; Tong et al., 2010; Vince, 2006; White et al., 1992). These regular polyhedral DGGSs have excellent properties, including a hierarchical structure, similar cell areas and shapes, and global addressability. However, mapping from a polyhedra to a sphere is a complex process because most cell edges do not coincide with meridians or parallels in the polyhedral DGGS. Thus, it is difficult to apply polyhedral DGGSs to existing spatial data that are based on the geographic coordinate system without a costly conversion process. However, this conversion process results in a loss of precision. Therefore, polyhedral DGGSs cannot satisfy the organization and updating requirements of global spatial data.

The traditional latitude–longitude grid system, includes a relationship between the cell edges and the geographic coordinate system and simple data storage and processing (Gregory et al., 2008). Thus, this DGGS has been widely used by researchers (Albergel et al., 2010; Fekète and Treinish, 1990; Lindstrom and Pascucci, 2001; Samet and Sivan, 1992) and agencies (such as the US Geological Survey, which provided the GTOPO30 data). However, the latitude–longitude grid cell areas become smaller towards the poles. In addition, the shape of the cells becomes increasingly distorted and turn into triangles at the poles. The North and South poles are points on the surface of the earth, but map to lines on the latitude–longitude plane (Sahr et al., 2003). Each of these polar singularities has an unique latitude but an uncertain longitude. The irregular cell areas and shapes decrease the manipulation accuracy and result in data redundancy. For example, the elevation of the poles is repeated 43,200 times in GTOPO30 (U.S. Geological Survey, 1999).

Attempts have been made to create adjusted latitude–longitude grid systems to obtain more consistent cell sizes. For example, by considering the curvature of the earth, Ottoson and Hauska (2002) proposed Ellipsoidal quadrilaterals (EQT) with latitudinal intervals that increase in polar regions and in the converging areas of the eastern and the western hemispheres.

Most of the adjusted latitude–longitude grid systems have complex initial division rules. Two separate index mechanisms must be implemented, one for the base level and one for lower levels (Ottoson and Hauska, 2002). No unified equation exists for converting between the latitude–longitude and address code, which complicates encoding and decoding because several different types of conversions must be implemented for the grids at different locations. These adjusted latitude–longitude grid systems are difficult to use for organizing and visualizing global multi-resolution data.

To address some of these difficulties, Sun et al. (2009) proposed the Degenerate Quadtree Grid (DQG) with longitudinal intervals that increase regularly from the equator to the poles and latitudinal intervals that remain the same. The longitudinal interval of one row in the DQG is half or twice as large as its adjacent row. The DQG has attractive properties, including a hierarchical structure, convergent geometrical distortion, and simple cell adjacencies. The DQG has solved the problem (to some extent) of large
grid area and shape distortions at high-latitudes. However, the polar grids are degenerated from quadrangles to triangles, as shown by the spherical triangle ABC in Fig. 1(a) and ADE in (b). These results indicate that polar singularities occur that negatively affect the organization of polar data. Sun et al. (2009) classified the DQG cells into seven categories based on their locations in an adjacent search. In this case, different cases remained within each category and no explanation was provided to assist in choosing a DQG case.

The regular latitude–longitude grid system and the existing adjusted latitude–longitude grid systems cannot satisfy the indexing and visualization requirements of polar regions. Most existing adjusted latitude–longitude grid systems have irregular cell region shapes or complex cell adjacencies. This paper provides a solution for the difficulties that are encountered in adjusted latitude–longitude DGGSSs.

3. Methodology

3.1. The partitioning method

Existing adjusted latitude–longitude grid systems have drawbacks, including polar singularities, irregular cell shapes and complex adjacencies. Adjusted latitude–longitude grid systems, whose latitudinal intervals increase from the equator to the poles while the longitudinal intervals remain the same, have highly irregular cell shapes. Adjusted latitude–longitude grid systems, whose latitudinal and longitudinal intervals change simultaneously, require complex border calculations. To achieve analogous cell areas and shapes, we developed the QQM based on constant latitude intervals ($\Delta\phi$) and varying longitudinal intervals ($\Delta\lambda$).

3.1.1. Determination of the first latitude border

To obtain comparable areas and shapes and to keep the cell adjacencies simple, we determined the first latitude border based on the following two criteria: (1) the arc length of the cell parallels above the border are approximately equal to that of the equator cells; and (2) no new vertex is generated at the border when the longitudinal intervals increase, Fig. 2(b).

A border is defined as a parallel where the longitudinal interval of cell changes. A bank is defined as a region between two borders, as shown in Fig. 3. Cells in the same bank have the same longitudinal intervals. Banks in the southern and northern hemisphere with the same cell longitudinal intervals are defined to share the same bank numbers. In addition, larger bank number correspond with smaller cell longitudinal intervals.

From criteria (1), the quantitative relationship between the base side length of the border cells ($S$) and the base side length of the equator cells ($SR$) is $S = SR$. From criteria (2) the number of cells below the border is an integral multiple of the number of cells above the border. To avoid an abrupt change at the border, the minimum integral multiple was chosen. Thus, the quantitative relationship between the longitudinal intervals of the grids on the border ($\Delta\lambda$) and the longitudinal intervals of the grids under the border ($\Delta\lambda_R$) is $\Delta\lambda = 2\Delta\lambda_R$. The equations for calculating $S$ and $SR$ are $S = R \cos \varphi \Delta\lambda$ and $SR = R \Delta\lambda_R$, where $\varphi$ is the latitude of the first border and $R$ is the radius of the equator. From these equations, $\cos \varphi = 0.5$ and $\varphi = \pm 60^\circ$. Therefore, the latitude of the first border is $7^\circ 60'$. The equation parameters in this section are diagramed in Fig. 4.

3.1.2. The subdivision method

The subdivision of the global surface in the QQM is detailed as follows. The latitudinal differences between the first border and the pole and equator were set to 30$^\circ$ and 60$^\circ$, respectively. The

![Fig. 2.](image1.png) **Fig. 2.** (a) Most existing adjusted latitude–longitude grid systems introduce new nodes, such as points D and E, when the longitudinal intervals change, which complicates the cell adjacencies. (b) No new node is introduced when the longitudinal interval changes in the QQM.

![Fig. 3.](image2.png) **Fig. 3.** Definition of borders and banks in the QQM.
Latitudinal interval was set at 30° to divide the earth’s surface into six latitude zones. Next, four latitude zones of \([ -60°, -30° ]\), \([ -30°, 0° ]\), \([ 0°, 30° ]\) and \([ 30°, 60° ]\) were divided into 24 quadrilateral grids (with a grid size of \(30° \times 60°\)) by the 0°, 60°, 120°, ±180°, −120°, and −60° meridians. In addition, the two spherical caps at latitudes of \([ -90°, -60° ]\) and \([ 60°, 90° ]\) were
divided into two regions. The spherical caps were mapped into their inscribed hexagons, with vertices at the intersections of the \( \pm 60^\circ \) parallels and the \( 0^\circ, 60^\circ, 120^\circ, \pm 180^\circ, -120^\circ \), and \(-60^\circ \) meridians. Next, two semi-hexagons, which belonged to the quadrangles, were formed (quadrangles \( A_0A_6A_{120}A_{180} \) and \( A_0A_{-60}A_{-120}A_{-180} \) in Fig. 5). The 28 quadrangles that were generated in the initial division were regarded as level 0 (Fig. 5). The grids at the equator were defined as equatorial, and the most polarward grids were defined as polar. The recursive subdivisions were conducted based on level 0.

1. The grids between the \(-60^\circ\) and \(60^\circ\) parallels in level 0 were recursively and equally divided into 4 smaller child grids by degrees at each lower level. For level \( N \), the latitudinal intervals of these grids were \( \Delta \phi = 30 / 2^N \), and the longitudinal intervals were \( \Delta \lambda = 180 / 2^N \), where \( N \) is a nonnegative integer. The relationship between the coordinates and grids in each level are provided in Section 3.2.

2. In level 1, the two semi-hexagons in level 0 with latitudes of \([60^\circ, 90^\circ]\) were divided into four child semi-hexagons with latitudinal intervals of \(15^\circ\) (Fig. 6(a)). The latitudes of \([60^\circ, 75^\circ]\) were equally divided into three grids with longitudinal intervals of \(60^\circ\). The latitudes of \([75^\circ, 90^\circ]\) with a longitudinal interval of \(180^\circ\) in the QQM became another child grid. Starting from level 2, the two polar semi-hexagons with latitudes of \([90^\circ, 30^\circ / 2^{N-1}, 90^\circ]\) were divided into 4 child semi-hexagons. In addition, other grids were equally and recursively divided into 4 child grids (Fig. 6(b)).

The unique and exact grid codes. In contrast with many existing adjusted grid systems, the encoding and decoding scheme in the QQM is uniform. In addition, the conversion between the address code and the latitude–longitude is simple.

3.2. The encoding and decoding scheme

The surface of the virtual earth was recursively partitioned into hierarchical quadrangles. These quadrangles could serve as a basis for indexing and addressing spatial data. A corresponding encoding and decoding scheme for the QQM was needed to efficiently organize and handle the spatial data.

For global GIS visualization, the cells that are accessed at the same time are always in continuous columns and rows at the same level. Because the data access efficiency can be improved by intensive storing and indexing the data that accessed at the same time, the 3D codes (level, row, column) were mapped as 64 bit integers to obtain exact and unique grid codes. In contrast with many existing adjusted grid systems, the encoding and decoding scheme in the QQM is uniform. In addition, the conversion between the address code and the latitude–longitude is simple.

### 3.2.1. The encoding scheme

The following steps were conducted to calculate the address code of the latitude–longitude \( (\lambda, \phi) \) at level \( N \). The unique identifiers of the level, row, column, and band in the QQM are all nonnegative integers.

1. Calculate the latitudinal interval \( \Delta \phi \) (in deg) by using the level number \( N \) in the following equation:

\[
\Delta \phi = \frac{180}{6 \times 2^N} 
\]

2. Calculate the row number \( I \) by using the latitude \( \phi \) and the latitudinal interval \( \Delta \phi \) as follows:

\[
I = \text{int} \left( \frac{\phi + 90}{\Delta \phi} \right)
\]

where the symbol ‘int’ indicates that the value is rounded down.

3. Calculate the band number \( k \) using the level number \( N \) and the row number \( I \) as follows:

\[
k = \begin{cases} 
\text{int} \left( \log_2 \left( I + \frac{1}{2} \right) \right) + 1, & I < I_{\text{max}}/2 \\
\text{int} \left( \log_2 \left( I_{\text{max}} - I + \frac{1}{2} \right) \right) + 1, & I > I_{\text{max}}/2 
\end{cases}
\]

when \( k > N + 1 \), then \( k = N + 1 \).

4. Calculate the longitudinal interval \( \Delta \lambda \) (in degrees) by using the band number \( k \) as follows:

\[
\Delta \lambda = \frac{180}{\text{int}(3 \times 2^{k-1})}
\]

5. Calculate the column number \( J \) by using the longitude \( \lambda \) and the longitudinal interval \( \Delta \lambda \) as follows:

\[
J = \text{int} \left( \frac{\lambda + 180}{\Delta \lambda} \right)
\]

6. Combine the valid-bit, the binary number of level, row and column into a 64 bit integer to obtain an address code for each

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Valid-bit</td>
</tr>
<tr>
<td>1 bit</td>
</tr>
</tbody>
</table>

Fig. 6. Recursive subdivision of the grids between \( 60^\circ \) to \( 90^\circ \) latitude. (a) Level 1, (b) Level 2.
3.2.2. The decoding scheme

The steps to calculate the latitudinal and longitudinal ranges of a grid with a given address code are as follows:

1. Separate the level number $N$, the row number $i$, and the column number $j$ from the address code.
2. Calculate the latitudinal interval $\Delta \varphi$ by using Eq. (1).
3. Calculate the minimum latitude $\varphi_{\text{min}}$ and the maximum latitude $\varphi_{\text{max}}$ by using the latitudinal interval $\Delta \varphi$ and $I$ when $\varphi_{\text{min}} = -90^\circ + \Delta \varphi \times I$ and $\varphi_{\text{max}} = -90^\circ + \Delta \varphi \times (I+1)$.
4. Calculate the longitudinal interval $\Delta \lambda$ following steps (3) and (4), as shown in Section 3.1.
5. Calculate the minimum longitude $\lambda_{\text{min}}$ and the maximum longitude $\lambda_{\text{max}}$ by using the longitudinal interval $\Delta \lambda$ and $J$ when $\lambda_{\text{min}} = -180^\circ + \Delta \lambda \times J$ and $\lambda_{\text{max}} = -180^\circ + \Delta \lambda \times (J+1)$.

The hierarchical relationships between the levels and the row-column numbers at levels 0 and 1 are shown in Fig. 7.

3.3. The grid relationships

3.3.1. The parent–child relationship

The hierarchical structure is shown in Fig. 7. The grids at level $N$ were subdivided into 4 child grids at level $N+1$. The parent grid at level $N-1$ and child grids at level $N+1$ of grid cell $(I,J)$ at level $N$ are shown in Table 2. The parameter $i$ was used to determine if a grid is located in the first or last row. This parameter has a value of 0 or 1 and is calculated by using the following equation:

$$i = \text{int}(I/l_{\text{max}}).$$

3.3.2. The adjacency relationship

In the QQM, neighbor cells that share edges are called edge-neighbor cells. Cells that only share common vertices are called vertex-neighbor cells. Based on their relative positions, the four edge-neighbor cells of a cell are defined as follows: Top-Edge-neighbor, Bottom-Edge-neighbor, Left-Edge-neighbor, and Right-Edge-neighbor. The four vertex-neighbor cells are defined as follows: Left-Top-Vertex-neighbor, Right-Top-Vertex-neighbor, Left-Bottom-Vertex-neighbor, and Right-Bottom-Vertex-neighbor.

Due to the uniform encoding and decoding scheme in the QQM, the row and column numbers of the neighbor cells can be easily calculated. The adjacent search algorithm in the QQM can be used to address the special conditions found in the polar grids and is simpler and more uniform than other adjusted latitude–longitude grid systems. The steps of the adjacent search algorithm are listed as follows:

1. Separate the level number $N$, the row number $i$ and the column number $j$ from the address codes.
2. Calculate the latitudinal interval $\Delta \varphi$ of level $N$ and the longitudinal interval $\Delta \lambda_i$ of row $i$ at level $N$ using Eqs. (1) and (4).
3. Calculate the latitudinal interval of the upper grid $\Delta \lambda_{i+1}$ and lower grid $\Delta \lambda_{i-1}$ by using Eq. (4). Next, calculate the ratios between $\Delta \lambda_i$ and $\Delta \lambda_{i+1}$ and between $\Delta \lambda_i$ and $\Delta \lambda_{i-1}$ by using the following equations $m_{i+1} = \text{ceil}(\Delta \lambda_i/\Delta \lambda_{i+1})$ and $m_{i-1} = \text{ceil}(\Delta \lambda_i/\Delta \lambda_{i-1})$, respectively, where the symbol ‘ceil’ means round up. If $\Delta \lambda_{i+1}$ or $\Delta \lambda_{i-1}$ are non-existent, the
(4) Calculate the neighbor cells of the grid \((I,J)\) at level \(N\) based on Table 3, where \(m\) is a positive integer and \(T, L, B, R, E\) and \(V\) indicate the Top, Left, Bottom, Right, Edge-neighbor and Vertex-neighbor, respectively. When a grid cell adjoins the border of a band, several LE or RE neighbor cells potentially occur when \(m_{i+1} > 1\) or \(m_{i-1} < 1\). Furthermore, LTV, RTV, LBV or RBV neighbor cells may not exist when the \(m_{i+1}\) or \(m_{i-1}\) values are infinite. However, this scenario does not result in mis-identified neighbor cells because the cells that are obtained are the same as the TE or BE neighbor cells.

(5) Calculate the maximum column numbers of the three involved rows by using the following equation: \(J_{\text{max}} = \frac{360}{\Delta \phi} - 1\). Next, check and adjust the row-column numbers of the obtained neighbor cells as follows: (1) the grid is nonexistent if \(i < 0\) or \(i > J_{\text{max}}\); (2) make \(j = J_{\text{max}}\) if \(j < 0\); and (3) make \(j = 0\) if \(j > J_{\text{max}}\).

Two examples of adjacent searches are given below.

The latitudinal and longitudinal intervals of grid \((1, 0)\) at level 1 are \(\Delta \phi = 15^{\circ}\) and \(\Delta \lambda = 60^{\circ}\). The longitudinal intervals of the upper and lower grids are \(\Delta \lambda_2 = 30^{\circ}\) and \(\Delta \lambda_0 = 180^{\circ}\), respectively. The ratios are \(m_{i+1} = \text{ceil}(\Delta \lambda_2 / \Delta \lambda_0) = \text{ceil}(60^{\circ} / 30^{\circ}) = 2\) and \(m_{i-1} = \text{ceil}(\Delta \lambda_2 / \Delta \lambda_2) = \text{ceil}(60^{\circ} / 180^{\circ}) = 1\). According to Table 3, the TEs are grids \((2, 0)\) and \((2, 1)\), the BE is grid \((0, 0)\), the LE is grid \((1, -1)\), the RE is grid \((1, 1)\), the LTV is grid \((2, -1)\), the RTV is grid \((2, 2)\), the LBV is grid \((0, -1)\), and the RBV is grid \((0, 2)\). The maximum column numbers for rows 0, 1, and 2 are \(J_{\text{max}} = 360^{\circ} / 180^{\circ} - 1 = 1\), \(J_{\text{max}1} = 360^{\circ} / 60^{\circ} - 1 = 5\), and \(J_{\text{max}2} = 360^{\circ} / 30^{\circ} - 1 = 11\), respectively. Check the row-column numbers of the neighbor cells and change the LE values from \((1, -1)\) to \((1, 1)\), the LTV values from \((2, -1)\) to \((2, 11)\), the LBV values from \((0, -1)\) to \((0, 1)\), and the RBV values from \((0, 2)\) to \((0, 1)\). The final adjacent search results of grid \((1, 0)\) at level 1 are grids \((0, 0)\), \((0, 1)\), \((1, 5)\), \((1, 10)\), \((2, 0)\), \((2, 1)\) and \((2, 2)\) (Fig. 8(a)).

The same procedure can be easily adapted to obtain the neighbor cells of grid \((0, 1)\) at level 1. According to Table 3, the TEs are grids \((1, 3)\), \((1, 4)\) and \((1, 5)\), the LE is grid \((0, 0)\), the RE is grid \((0, 2)\), the LTV is grid \((1, 2)\), the RTV is grid \((1, 6)\), and the BE, LBV and RBV are nonexistent. The row-column numbers of the neighbor cells were checked by changing the RE values from \((0, 2)\) to \((0, 0)\) and by changing the RTV values from \((1, 6)\) to \((1, 0)\). The final adjacent search results of grid \((0, 1)\) at level 1 are grids \((0, 0)\), \((0, 1)\), \((1, 3)\), \((1, 4)\), \((1, 5)\) and \((1, 0)\) (Fig. 8(b)). Although the relative locations of some neighbors vary, their row-column numbers are correct. This result indicates that the adjacent search algorithm is applicable to the polar grids.

4. Performance analysis and comparison

Relative to the latitude–longitude grid systems, the adjusted latitude–longitude grid systems produce more analogous cell areas. However, many adjusted latitude–longitude grid systems have non-hierarchical structures, complex cell adjacencies and irregular grid shapes. In contrast, the geometrical distortion is limited and the encoding, decoding, and adjacent search performances are efficient in the DQG (Sun et al. 2009). Therefore, the DQG was chosen as an example of adjusted latitude–longitude grid systems and was compared with the QQM.

The level definitions for the QQM and DQG are different. The QQM starts from level 0 with 28 grids and the DQG starts from level 1 with 24 grids. The grid numbers in these two DGGSs are nearly quadrupled with each level increase. Therefore, the DQG grid numbers at level \(N+1\) are approximately equal to the QQM grid numbers at level \(N\). By subtracting 1 from the DQG level numbers (changing \(N+1\) to \(N\)), the DQG level number was comparable with the QQM level number.

The experiments in Sections 5 and 6 were performed on a PC with an Intel(R) Core™3.1 GHz 2 CPU processor with 3.2GB RAM and NVIDIA Quadro 600 graphics.

4.1. Geometrical distortion of the grids

The lengths of longitude edges were calculated from the following equation:

\[ S = R \times \Delta \phi \]  

(6)

In addition, the lengths of the latitude edges were calculated by using the following equation:

\[ S = R \cos \phi \times \Delta \lambda \]  

(7)

where \(R\) is the radius of the earth, \(\Delta \phi\) is the latitudinal interval (in radians), \(\Delta \lambda\) is the longitudinal interval (in radians) and \(\phi\) is the latitude of the circle (in radians).

The cell areas \(A\) were calculated by using the following equation:

\[ A = \int_0^{\Delta \lambda} d\theta \int_0^{R \cos \phi_{\text{max}}} R \frac{R}{\sqrt{R^2 - r^2}} dr = (\sin \phi_{\text{max}} - \sin \phi_{\text{min}}) \Delta \lambda R^2 \]  

(8)

where \(\phi_{\text{max}}\) and \(\phi_{\text{min}}\) are the maximum and minimum latitudes of the cell, respectively.

By assuming that the average radius of the earth is \(R = 6,371,004\), Eqs. (6)–(8) were used to calculate all grid areas and edge lengths for levels 0–25. The lengths of the longitude edges \(S_{\text{lon}}\), the equator edges \(S_r\), the minimum latitude edges \(S_{\text{min}}\) and

![Fig. 8](image-url) Two examples of adjacent searches at level 1. The latitude range is \([-90^{\circ}, -60^{\circ}]\). (a) grid \((1, 0)\) and its neighbours, (b) grid \((0, 1)\) and its neighbours.
which, and 2.10, respectively. Results from a similar DQG calculation suggested the level number before converging at approximately 1.047, 2.10.

Fig. 9. Comparison between the QQM and DQG geometrical distortions. (a) $S_{\text{max}}/S_R$ (b) $S_{\text{max}}/S_{\text{min}}$ (c) $A_{\text{max}}/A_{\text{min}}$.

The maximum latitude edges $S_{\text{max}}$ (in meters), the minimum cell areas $A_{\text{min}}$, and the maximum cell areas $A_{\text{max}}$ (in square meters) and the ratios of each level are listed in Table 4.

From Table 4, the QQM with (1) $S_{\text{min}} = S_R/2$ and $S_{\text{max}} = S_{\text{min}}$ (from which $S_{\text{max}}/S_R = S_{\text{max}}/S_{\text{min}}$ was deduced) and (2) the values of $S_{\text{max}}/S_R$, $S_{\text{max}}/S_{\text{min}}$, and $A_{\text{max}}/A_{\text{min}}$ gradually increased with increasing level number before converging at approximately 1.047, 2.10 and 2.10, respectively. Results from a similar DQG calculation indicated the following: (1) $S_{\text{min}} = S_R$ (from which $S_{\text{max}}/S_{\text{min}} = S_{\text{max}}/S_R$ was deduced) and (2) the values obtained from $S_{\text{max}}/S_R$, $S_{\text{max}}/S_{\text{min}}$, and $A_{\text{max}}/A_{\text{min}}$ converged at approximately 1.571, 2.23 and 2.23, respectively. The convergent values for $S_{\text{max}}/S_R$, $S_{\text{max}}/S_{\text{min}}$, and $A_{\text{max}}/A_{\text{min}}$ in the QQM were smaller than in the DQG (Fig. 9). The converging ratios suggested that the cell areas and shapes of QQM were analogous in the recursive subdivisions. The geometrical distortions of the QQM and DQG stabilized at level 22 (as shown in Table 4 and Fig. 9). The relative numbers of cells at level 22 in the QQM and DQG.

### Table 5

<table>
<thead>
<tr>
<th>Class center (± 2.5)</th>
<th>QQM</th>
<th>DQG</th>
</tr>
</thead>
<tbody>
<tr>
<td>52.5</td>
<td>0.0568679</td>
<td>0</td>
</tr>
<tr>
<td>57.5</td>
<td>0.064183</td>
<td>0</td>
</tr>
<tr>
<td>62.5</td>
<td>0.0666585</td>
<td>0</td>
</tr>
<tr>
<td>67.5</td>
<td>0.0698254</td>
<td>0</td>
</tr>
<tr>
<td>72.5</td>
<td>0.0739117</td>
<td>0.0598395</td>
</tr>
<tr>
<td>77.5</td>
<td>0.0794154</td>
<td>0.0861544</td>
</tr>
<tr>
<td>82.5</td>
<td>0.0872962</td>
<td>0.101846</td>
</tr>
<tr>
<td>87.5</td>
<td>0.0998054</td>
<td>0.116439</td>
</tr>
<tr>
<td>92.5</td>
<td>0.124271</td>
<td>0.144982</td>
</tr>
<tr>
<td>97.5</td>
<td>0.275141</td>
<td>0.320998</td>
</tr>
<tr>
<td>102.5</td>
<td>0.00263509</td>
<td>0.0779506</td>
</tr>
<tr>
<td>107.5</td>
<td>0</td>
<td>0.0185509</td>
</tr>
<tr>
<td>112.5</td>
<td>0</td>
<td>0.0188776</td>
</tr>
<tr>
<td>117.5</td>
<td>0</td>
<td>0.0192413</td>
</tr>
<tr>
<td>122.5</td>
<td>0</td>
<td>0.0196486</td>
</tr>
<tr>
<td>127.5</td>
<td>0</td>
<td>0.0201065</td>
</tr>
<tr>
<td>132.5</td>
<td>0</td>
<td>0.0201065</td>
</tr>
<tr>
<td>137.5</td>
<td>0</td>
<td>0.0201065</td>
</tr>
<tr>
<td>142.5</td>
<td>0</td>
<td>0.00897635</td>
</tr>
<tr>
<td>147.5</td>
<td>0</td>
<td>0.004219</td>
</tr>
<tr>
<td>152.5</td>
<td>0</td>
<td>0.00299195</td>
</tr>
<tr>
<td>157.5</td>
<td>0</td>
<td>0.000256851</td>
</tr>
</tbody>
</table>

**Fig. 9.** Comparison between the QQM and DQG geometrical distortions. (a) $S_{\text{max}}/S_R$ (b) $S_{\text{max}}/S_{\text{min}}$ (c) $A_{\text{max}}/A_{\text{min}}$.
frequency histogram (Fig. 10). The equatorial cell is defined to have an area of 100%. Furthermore, each class is 5% wide. The value assigned to each class represents the relative number of cells that have an area of within \(\pm 2.5\%\) of the class center value. Thus, the 97.5\% class contains the percent of cells that have an area of \([95\%, 100\%]\). The area distribution of the QQM is more compact (Fig. 10). The area range of the QQM is \([50\%, 105\%]\). In contrast, the area range of the DQG is \([70\%, 160\%]\). The smaller area range of the QQM indicates that the cell areas are more consistent.

4.2. Encoding and decoding efficiency

The encoding and decoding efficiency is defined as the encoding and decoding time in 1 ms. In Table 6, the encoding and decoding efficiency of the QQM and DQG are compared by using 5 million random QQM address codes, 5 million random DQG address codes, and 5 million random latitude–longitude locations. Table 6 indicates that the encoding and decoding efficiency of the QQM was 1.9 times greater than the encoding and decoding efficiency of the DQG.

![Figure 10. Relative frequency histogram of the cell areas in the QQM and DQG.](image)

**Table 6**

<table>
<thead>
<tr>
<th>Efficiency (times/ms)</th>
<th>Efficiency ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>QQM</td>
<td>DQG</td>
</tr>
<tr>
<td>3729.69</td>
<td>1956.56</td>
</tr>
<tr>
<td>2822</td>
<td>1498.27</td>
</tr>
</tbody>
</table>

4.3. Adjacent search efficiency

Here, the adjacent search efficiency is defined as the adjacent search time in 1 ms. The adjacent search efficiency of the QQM and DQG at the same level are compared here. For the \(N < 5\) level, the QQM cell numbers are so small that the time consumed in an adjacent search is nearly zero. For the \(N > 12\) level, the time consumed is too great for testing. Thus, levels 6 through 12 were selected in this experiment because their results are stable and readily reflect the relationships between the QQM and DQG (Table 7). As mentioned in the introduction, different cases occur inside some of the seven cell categories in the DQG. Unfortunately, no explanation exists regarding how to choose a case. In this experiment, we calculated one of the DQG cases without determination. Thus, the neighbor cells that were obtained in the DQG may not be accurate. If the detailed judgments are employed, the adjacent search efficiency of the DQG may decrease. In the QQM, all of the neighbor cells were correct (as shown in Section 4.2). The adjacent search efficiency of the QQM is 8 times greater than the adjacent search efficiency of the DQG (Table 7). This difference partially results from the encoding and decoding efficiency variations in the QQM and DQG.

5. Application

The QQM was used to determine the levels and blocks that were used to organize the global Blue Marble data at a resolution of 1000 m. Image blocks with \(128 \times 64\) pixels were used to construct the image pyramid. The polar data was stored only once in a polar image block. Each pole had two polar image blocks. Therefore, each polar data point was stored twice at any level in the QQM. The data redundancy in the polar regions significantly decreased. The organized global image data was visualized by using C++ and OpenGL (Fig. 11). The QQM uses a simple structure with smooth visualization, especially in the polar regions.

6. Conclusions and future work

A new pole-oriented discrete global grid, QQM, was proposed in this study. The QQM uses semi-hexagon subdivisions in the polar regions, and rectangular subdivisions elsewhere. In addition, the QQM significantly mitigates reoccurring problems that are found in existing adjusted latitude–longitude grid, including polar singularities, irregular cell shapes and complex adjacencies. This improvement in the polar regions allows the QQM to index and visualize global spatial data as follows:

1. **Regular hierarchy:** Starting from level 1, each QQM cell has four descendants at the next level. In the DQG, the polar grids have three descendants and other grids have four descendants.

**Table 7**

Adjacent search efficiencies of the QQM and DQG.

<table>
<thead>
<tr>
<th>(N)</th>
<th>QQM Search times</th>
<th>Time QQM takes</th>
<th>Efficiency (times/ms)</th>
<th>Efficiency ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>109,161</td>
<td>32</td>
<td>QQM 3584</td>
<td>452.794</td>
</tr>
<tr>
<td>7</td>
<td>458,752</td>
<td>124</td>
<td>QQM 3699.61</td>
<td>464.809</td>
</tr>
<tr>
<td>8</td>
<td>1,835,008</td>
<td>468</td>
<td>QQM 3920.96</td>
<td>485.969</td>
</tr>
<tr>
<td>9</td>
<td>7,340,032</td>
<td>1844</td>
<td>QQM 3980.49</td>
<td>483.605</td>
</tr>
<tr>
<td>10</td>
<td>29,360,128</td>
<td>7419</td>
<td>QQM 3957.42</td>
<td>483.135</td>
</tr>
<tr>
<td>11</td>
<td>117,440,512</td>
<td>295,00</td>
<td>QQM 3981.03</td>
<td>483.354</td>
</tr>
<tr>
<td>12</td>
<td>469,762,048</td>
<td>118,404</td>
<td>QQM 3967.45</td>
<td>483.135</td>
</tr>
</tbody>
</table>
(2) Quadrangular cells: All QQM cells are quadrangles, including semi-hexagons and rectangles. In contrast, the DQG cells are triangular in the polar regions.

(3) Convergent geometrical distortion: The $S_{\text{max}}/S_{\text{min}}$ and $A_{\text{max}}/A_{\text{min}}$ of the QQM converge at approximately 2.10, which is less than the convergence point in the DQG (2.23). Thus, the cell areas and shapes were more consistent in the QQM.

(4) An efficient encoding-decoding scheme: In this experiment, the encoding and decoding efficiencies of the QQM were approximately 2822 times/ms and 3729 times/ms, respectively (approximately 1.9 times greater than the corresponding DQG values).

(5) An efficient adjacent search algorithm: In this experiment, the adjacent search efficiency of the QQM was approximately 3950 times/ms, which was approximately 8 times that of the DQG.

The QQM results in similar cell areas and cell shapes, which makes this DGGS suitable for indexing and visualizing global data, especially polar data. In addition, this DGGS significantly reduces data redundancy in the polar regions. The QQM has a regular hierarchical structure and a simple relationship with the latitude and longitude. Thus, the QQM is easily used with existing geographical data. In addition, the QQM has a simple adjacency structure that was designed to facilitate spatial analysis operations. Even the neighbor cells of the polar cells can be found by using the uniform adjacent search algorithm. The QQM smoothly visualizes data and overcomes the oscillations that are frequently found in the polar regions with other DGGSs that are based on the geographic coordinate system. In addition to addressing the organization of multi-scale global spatial data based on QQM, monitoring the impacts of climate change on polar glaciers and on sea levels around the world is the next step in this research.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (Nos. 41171314 and 41023001). In addition, comments from the anonymous reviewers and editor are appreciated.

References


Fig. 11. Renderings of organized global image data based on the QQM. (a) and (b) are obtained from a polar viewpoint and (c) and (d) are obtained from an equatorial viewpoint.


