TONG Xiaochong, BEN Jin, ZHANG Yongsheng, et al. The 3D Visualization Method of Hexagonal Discrete Global Grid Data[J]. Acta Geodaetica et Cartographica Sinica, 2013, 42(3): 374-382. (童晓冲, 贲进, 张永生, 等. 全球六边形离散格网数据的三维可视化方法[J]. 测绘学报, 2013, 42(3): 374-382.)

全球六边形离散格网数据的三维可视化方法

童晓冲[1,2], 贲 进[2], 张永生[2], 汪 滢[2]

1. 北京师范大学 地表过程与资源生态国家重点实验室, 北京 100875; 2. 信息工程大学 地理空间信息学院, 河南 郑州 450052

The 3D Visualization Method of Hexagonal Discrete Global Grid Data

TONG Xiaochong[1,2], BEN Jin[2], ZHANG Yongsheng[2], WANG Ying[2]

1. State Key Laboratory of Earth Surface Processes and Resource Ecology, Beijing Normal University, Beijing 100875, China;
2. Institute of Geospatial Information, Information Engineering University, Zhengzhou 450052, China

Abstract: Aiming at the 3D visualization method of hexagonal discrete global grid, a new spatial operation structure (hexagonal quaternary balanced structure, HQBS) for hexagonal discrete grid, was designed. It used 4 codes to encodes units, defined spatial vectors for grids and realized basic operations. Based on the operations, spatial index can be easily solved. And dynamic generation algorithms, spatial visualization methods and clipping of visible area of global grid were studied. Through visualization experiments, conclusions were obtained as follows. The average generation efficiency of global dynamic grid could reach 110 units per ms to 370 units per ms, and the loading time efficiency of grid data and spatial data per layer was about 300 ms. The average refreshing level of global discrete grid loaded with spatial data was about 20 frames per second, which ensured real-time displaying requirements.

Key words: discrete global grid system; hexagon; 3D visualization; clipping of visible area

摘 要: 针对全球六边形离散格网的三维显示化方法开展研究, 设计了一种六边形格网的空间层次结构(hexagonal quaternary balanced structure, HQBS), 采用四位码元对格网单元进行编码, 定义并实现了格网向量的基本运算, 利用这些运算可以方便地实现格网单元的空间索引。在此基础上还研究了全球离散格网的动态生成与显示算法、可视化区域裁剪等相关内容。试验表明: 全球格网动态生成的效率 110～370 单元/ms 之间, 加载空间数据后, 格网数据和空间数据逐层加载的时间在 300 ms 左右, 能够保证加载空间数据后的显示刷新率在 20 帧/s 左右。

关键词: 全球离散格网系统; 六边形; 三维可视化; 可视区裁剪

中图分类号: P228 **文献标识码**: A **文章编号**: 1001-1595(2013)03-0374-10

基金项目: 国家自然科学基金(41201392; 41271391); 国家自然科学基金重点项目(40930104); 国家 863 计划(2009AA12Z218)

1 引 言

基于多面体剖分的全球离散格网系统(discrete global grid system, DGGS)是一种用正多面体逼近球面, 将地球递归剖分为面积、形状近似或相等且具有多分辨率层次单元的新型全球空间数据结构[1-3]。它在许多科学应用领域具有良好的前景[4,5], 特别是在面向全球系统的模拟、分析与显示等方面具有优势[6-7]。作为一种空间模拟分析系统, 格网单元的选择关系到空间模拟的性能与效果。在 3 种能够进行规则化空间剖分的几何格网图形中, 六边形是最紧凑的一种, 它具有各向同性、邻域一致、角分辨率大等特性[1,3], 并被证明可以延续到球面格网系统上[3-4,7-9]。而可视化的目的就是为了更好地对地学空间模拟进行有效的展示。

在 DGGS 中, 随着剖分层次的增加, 格网的数据量也呈几何级数增长, 为了保证系统的效率, 必须研究合理高效的可视化方法。目前, 基于四边形格网、三角形格网的全球空间数据可视化技术已经相对成熟, 许多研究团队都给出了自己的解决方案[2,10-12], 基于平面六边形格网的空间数据可视化技术也有相关的研究报道[13]。而对于球面六边形离散格网的显示而言, 仍存在部分有别于前面的地方, 特别是在球面六边形格网的编码与索引、格网动态生成与可视区裁剪、格网数据

的实时显示等方面。目前，孔径为 3 的球面六边形格网编码与索引方法，文献[14—15]给出了相应的解决方案，而对于孔径为 4 的球面六边形格网，却鲜见相关的解决方案。本文以文献[1,16—17]设计的孔径为 4 的球面六边形格网 4HI 为对象，针对上面几个关键环节展开讨论。

2　全球六边形离散格网的编码运算与索引

2.1　全球六边形离散格网的编码运算

索引系统是全球离散格网的重要组成部分，格网显示需要的是一种简洁的编码方式和高效的索引技术作为支撑。针对孔径为 4 的六边形层次格网 4HI(图 1)可采用三角形四叉树进行标识，而对格网的标识实际上是对单元中心的标识，图 2 中“●”标识 4HI 格网的中心，“·”标识的是三角形四叉树结构中非 4HI 格网中心的节点。因此，4HI 的层次结构可以看做是三角四叉树的子集，可以采用类似的编码方法。图 2 是层次 $n=3$ 的剖分情况，可以对应第 3 层的四叉树结构得到相应的编码方式。

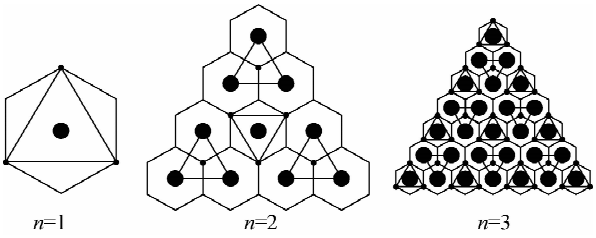

图 1　六边形层次格网 4HI 和三角四叉树

Fig. 1　Hexagonal hierarchical grid 4HI and triangle quaternary tree

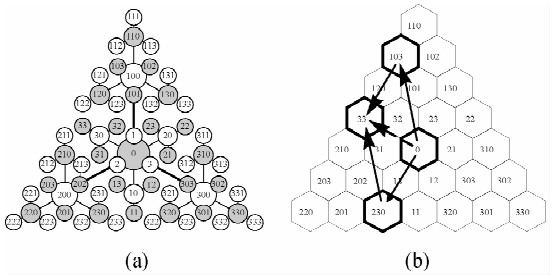

图 2　$n=3$ 层四叉树结构和 HQBS 格网编码

Fig. 2　The triangle quaternary tree of $n=3$ and HQBS grid code

笔者称这种用四叉树形式编码的六边形格网结构为 HQBS(hexagonal quaternary balanced structure)。设格网的剖分层次为 λ，则 HQBS 的任意格网单元可用 $G_\lambda=g_{\lambda-1}g_{\lambda-2}\cdots g_1g_0$，$g_i\in\{0,1,2,3\}$ 来表示。HQBS 编码是一种位置记录系统，它隐含了单元与原点(编码为 0 的单元中心)之间的位置关系。将 HQBS 单元编码看做一个向量，HQBS 单元的⊕运算可以定义为两个向量之间的加法，如图 2(b)所示。结合 HQBS 编码、向量运算和三角形四叉树的特点，任意层次的编码 G_λ 可展开成式(1)的形式。

$$\begin{aligned}G_\lambda=g_{\lambda-1}g_{\lambda-2}g_{\lambda-3}\cdots g_1g_0=&\\(-1)^0\cdot(f(g_{\lambda-1}))\cdot g_{\lambda-1}\cdot10^{\lambda-1}\oplus(-1)^1\cdot&\\(f(g_{\lambda-2})\cdot f(g_{\lambda-1}))\cdot g_{\lambda-2}\cdot10^{\lambda-2}\oplus\cdots\oplus&\\(-1)^{\lambda-2}\cdot(f(g_1)\cdots f(g_{\lambda-1}))\cdot g_1\cdot10^1\oplus&\\(-1)^{\lambda-1}\cdot(f(g_0)f(g_1)\cdots f(g_{\lambda-1}))\cdot g_0&\end{aligned}\tag{1}$$

式中，$f(g)=\begin{cases}-1 & g=0\\ 1 & g\neq0\end{cases}$。例如：编码 301 001＝300 000⊕−1000⊕1。下面首先给出两个定义。

(1) 编码序列：一系列码元的组合，用〈.,.,...,.〉表示。$\forall a_i\in\{-3,-2,-1,0,1,2,3\}$，则$\langle a_0,a_1,\cdots,a_{n-1}\rangle=a_0\underbrace{0\cdots0}_{n-1}\oplus a_1\underbrace{0\cdots0}_{n-2}\oplus\cdots\oplus a_{n-1}$称为编码序列。其中，格网编码只是编码序列的特例。例如：302＝300⊕−2＝〈3,0,−2〉。

(2) 规定化：对于任意编码序列$\langle a_0,a_1,\cdots,a_{n-1}\rangle$，如果能转化成满足式(1)的编码序列(称为合法 HQBS 编码)，则将这个转换过程称为编码规定化，用$\widehat{\langle\cdot,\cdot\rangle}$表示。

和十进制的加法运算类似，HQBS 的⊕运算也遵循加法的查找与进位操作，不过由于进率不同，两者的查找表不一致。根据式(1)，HQBS 的展开码元有正负之分，因此有{−3,−2,−1,0,1,2,3}7 个码元，结合格网的层次结构和编码规则可以构造表 1 中 HQBS 的⊕运算查找表。

表 1　⊕运算的查找表

Tab. 1　Look-up table for ⊕ operation

⊕	−3	−2	−1	0	1	2	3
−3	30	1	2	−3	32	31	0
−2	1	20	3	−2	23	0	21
−1	2	3	10	−1	0	13	12
0	−3	−2	−1	0	1	2	3
1	32	23	0	1	−10	−3	−2
2	31	0	13	2	−3	−20	−1
3	0	21	12	3	−2	−1	−30

利用⊕运算查找表，可以首先设计编码序列$A=\langle a_{n-1},a_{n-2},\cdots,a_0\rangle$的规定化运算，即$G=\widehat{A}$，编码序列规定化过程如下。

步骤1：初始化过程并① 编码序列$\langle g_n,g_{n-1},\cdots,g_1,g_0\rangle$，$g_0=g_1=\cdots=g_n=0$；循环变量$i=0$；② 定义新的数据结构$Q$，含两位整型的数据结构，分别为$\{q_1,q_0\}$；③ ⊕运算查找表$B[7][7]$，数据类型为$Q$，针对表1中的数据，根据式(1)的编码展开规则，例如$B[0][0]=\{3,0\}$，$B[0][1]=\{0,1\}$，$B[0][3]=\{0,-3\}$，$B[0][4]=\{3,-2\}$。

步骤2：if $i\leqslant n-1$：转到步骤3。

步骤3：if $a_i\neq 0$：初始化循环变量$L=-1$，转到步骤4；else：$g_i=0$，$i=i+1$，转到步骤2。

步骤4：if $L\leqslant 1$：$g_i=L\times a_i$，初始化进位变量$J=(L>0)?\ 0:g_i$，符号变量$F=(g_i>0)?\ -1:1$，定义查找表查找的两位码元分别为k_1、k_0(整型)，转到步骤5；else：编码序列A不存在规定化后的合法编码G，返回FALSE。

步骤5：if $i<n$：$k_1=B[a_{i+1}+3][J+3].q_1$，$k_0=B[a_{i+1}+3][J+3].q_0$，转到步骤6；else：转到步骤9。

步骤6：if $k_0=0$：$g_{i+1}=0$，$J=k_1$，$i=i+1$，转到步骤5；else：转到步骤7。

步骤7：if $k_0\times F>0$：$g_{i+1}=k_0$，$J=k_1$，$F=-F$，$i=i+1$ 转到步骤5；else：转到步骤8。

步骤8：$g_{i+1}=-k_0$，$J=B[g_{i+1}+3][k_1+3].q_0$，$F=-F$，$i=i+1$，转到步骤5。

步骤9：if $J=0$：初始化标识变量$Z=n$，转到步骤10；else：转到步骤12。

步骤10：while $g_{Z-1}=0$：$Z=Z-1$；转到步骤11。

步骤11：if $g_{Z-1}<0$：$L=L+2$，转到步骤4；else：转到步骤13。

步骤12：if $J<0\ \|\ F\times J<0$：$L=L+2$，转到步骤4；else：$g_n=J$，转到步骤13。

步骤13：输出规定化后的合法HQBS编码$G=|g_n||g_{n-1}|\cdots|g_1||g_0|$，返回TRUE。

由于任意HQBS编码都可以利用式(1)展开成编码序列的形式，完成各种运算后再规定化成HQBS编码，因此规定化运算是所有格网编码运算的基础。根据⊕查找表和编码序列规定化方法，可以设计格网编码的⊕运算，设两个格网编码按照式(1)展开成编码序列为$G_\lambda=\langle g_{\lambda-1},g_{\lambda-2},\cdots,g_0\rangle$，$H_\mu=\langle h_{\mu-1},h_{\mu-2},\cdots,h_0\rangle$，其中$\lambda\geqslant\mu$，计算编码$L=G_\lambda\oplus H_\mu$，HQBS格网编码⊕运算步骤如下。

步骤1：初始化过程：① 设置⊕运算查找表$B[7][7]$，数据类型、初始化方法与表2中一致；② 进位变量$J=0$；循环变量$i=0$；③ 定义查找表查找的两位码元分别为k_1、k_0(整型)；④ 定义用于记录查找表查找的两位码元的中间变量分别为m_1、m_0(整型)；⑤ 编码L展开的编码序列为$\langle l_\lambda,l_{\lambda-1},l_{\lambda-2},l_{\lambda-3},\cdots,l_1,l_0\rangle$。

步骤2：if $i<\mu$：$k_1=B[g_i+3][h_i+3].q_1$，$k_0=B[g_i+3][h_i+3].q_0$，$m_1=B[k_0+3][J+3].q_1$，$m_0=B[k_0+3][J+3].q_0$，$l_i=m_0$，$J=B[k_1+3][m_1+3].q_0$，$i=i+1$，转到步骤2；else：转到步骤3。

步骤3：if $i<\lambda$：转到步骤4；else：转到步骤5。

步骤4：if $J=0$：$l_i=g_i$，$i=i+1$，转到步骤3；else：$k_1=B[g_i+3][J+3].q_1$，$k_0=B[g_i+3][J+3].q_0$，$l_i=k_0$，$J=k_1$，$i=i+1$，转到步骤3。

步骤5：$l_i=J$，调用表2中HQBS编码序列规定化过程，返回合法格网编码$L=\langle l_\lambda,l_{\lambda-1},l_{\lambda-2},l_{\lambda-3},\cdots,l_1,l_0\rangle$。

下面用例子说明⊕运算和规定化的过程，例：103⊕230=33(图3(a))；23⊕32=101(图3(b))。

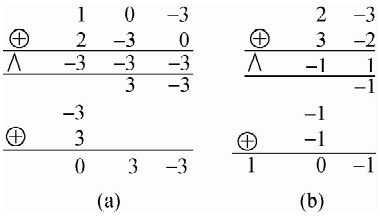

图3 HQBS格网编码⊕运算和规定化例

Fig. 3 Examples of ⊕ operation and normalization for HQBS grid codes

2.2 格网的层次索引

利用HQBS编码的⊕运算，格网的索引查找算法可以快速地实现，包括单元的邻近查找和层次搜索。首先是邻近单元的查找。由于六边形单元只存在6个直接相邻的单元，分别归属6个不同的方向，并不像三角形格网和四边形格网那样需要考虑边邻近和角邻近的情况，因此单元G_λ的邻近单元为

$G_\lambda\oplus 12$、$G_\lambda\oplus 13$、$G_\lambda\oplus 31$、$G_\lambda\oplus 32$、$G_\lambda\oplus 23$、$G_\lambda\oplus 21$

在邻近单元搜索的基础上，很方便地可以得到层次搜索中的子单元的查找方法。对于剖分层次是λ的任意一个4HI格网单元，其在$\lambda+1$层必

有 7 个子单元，其中 1 个与它本身是对准的[1,3,17]，其余 6 个是那个单元的邻近单元。设 λ 层的单元 $G_\lambda=g_\lambda g_{\lambda-1}\cdots g_0$，查找其在 $\lambda+1$ 层的子单元。

(1) G_λ 子单元中与它对准的中心子单元编码为：$G_{\text{son},0}=G_{\lambda+1}=g_\lambda g_{\lambda-1}\cdots g_0 0$。

(2) 其他周围的 6 个子单元 $G_{\text{son},i}(i=1,2,3,4,5,6)$ 分别中心子单元的 6 个邻近单元。

$G_{\lambda+1}\oplus 12$、$G_{\lambda+1}\oplus 13$、$G_{\lambda+1}\oplus 31$、$G_{\lambda+1}\oplus 32$、$G_{\lambda+1}\oplus 23$、$G_{\lambda+1}\oplus 21$

另一类层次关系搜索就是查找父单元。4HI 层次格网可以分成两类：一类是与其父单元对准的单元，称为中心继承单元[1]，该类单元具有 1 个父单元；另一类是与其父单元不对准，称之为偏心继承单元[1]，该类单元具有两个父单元。根据 HQBS 编码的特点，针对 λ 层的单元 $G_\lambda=g_\lambda g_{\lambda-1}\cdots g_0$，分情况讨论。

(1) 如果码元满足 $g_0=0$ 条件，则该单元即为中心继承单元，其父单元为

$$G_{\text{dad}}=G_{\lambda-1}=g_\lambda g_{\lambda-1}\cdots g_1 g_1$$

(2) 如果不满足(1)的条件，则该单元为偏心继承单元。由于 $g_0\neq 0$，则 $g_0\in M=\{1,2,3\}$，计算集合 $N=M-\{g_0\}=\{n_1,n_2\}$，则 G_λ 的两个父单元分别为

$G_{\text{dad},1}=g_\lambda g_{\lambda-1}\cdots g_1\oplus -n_1$、$G_{\text{dad},2}=g_\lambda g_{\lambda-1}\cdots g_1\oplus -n_2$

3　六边形格网的动态生成与可视区裁剪

3.1　六边形格网的动态生成算法

在全球六边形格网可视化过程中，还必须根据格网的层次确定每 1 个六边形单元 7 个节点(1 个中心节点＋6 个边界节点)的坐标值，这样才可以进行格网显示，形成一个连续的格网表面，因此，六边形单元的生成是球面建模的重要环节。文献[18]提出了一种静态生成算法，它采用单元变换的方式一次性生成所有单元的节点坐标，并将其保存下来。这种方式在实时显示中将耗费大量磁盘空间用于存储坐标数据，读取和检索将变得非常困难。本文设计了一种任意层次六边形离散格网的动态生成方法。为了简单起见，先考虑 HQBS 六边形格网在平面上的情况，基本步骤如下：

(1) 设定初始的格网层次 n_0，该层次上有且只有一个单元 G_{n_0,G_0}，其中下标 G_0 为其父单元(n_0-1 层)的编码，G_0 的下标 0 说明 G_{n_0,G_0} 单元是 G_0 的中心继承单元，根据 2.2 节的描述有 $G_{n_0,G_0}=G_0 0$。该单元由 7 个节点构成，中心的节点坐标用 $P(G_{n_0,G_0})_0$ 来表示，周围 6 个节点坐标分别用 $P(G_{n_0,G_0})_j$ 来表示，其中 $j=1\sim 6$，排列顺序如图 4。

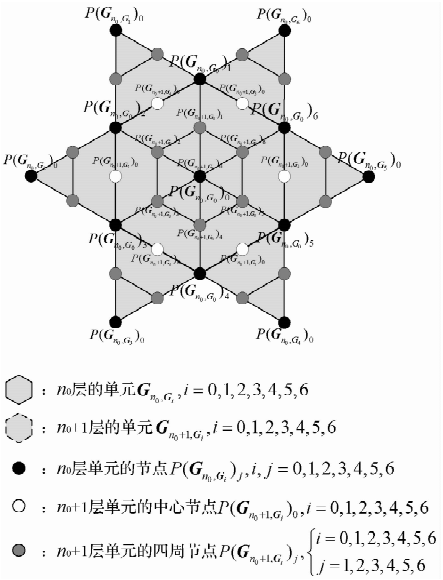

图 4　格网动态生成过程中单元与节点之间的关系

Fig. 4　The relation between the cells and nodes in the process of generating grids

(2) 在 n_0 层的离散格网上，G_{n_0,G_0} 的 6 个邻近单元的编码分别为 G_{n_0,G_i}，$i,j=1,2,\cdots,6$。

(3) n_0 层的单元 G_{n_0,G_0} 拥有 7 个子单元。含中心继承单元 1 个：G_{n_0+1,G_0}，令其对应的节点分别为 $p(G_{n_0+1,G_0})_j$，$j=0,1,\cdots,6$；偏心继承单元 7 个：G_{n_0+1,G_i}，令其对应的节点分别为 $P(G_{n_0+1,G_i})_j$，$j=0,1,\cdots,6$。

格网的动态层次生成方法，由低层次的单元生成高层次的单元。根据父单元的编码 G_{n_0,G_0}，检索 7 个子单元的编码 G_{n_0+1,G_i}，$i=0,1,\cdots,6$。子单元 G_{n_0+1,G_i} 的所有节点坐标由 G_{n_0,G_0} 及其 6 个邻近单元 G_{n_0,G_i} 的节点通过简单的平均运算得到，其中 $i=1,2,\cdots,6$。在初始 n_0 层的格网上，存储着单元 G_{n_0,G_i}，$i=0,1,\cdots,6$ 的全部节点坐标，则 G_{n_0,G_0} 的 n_0+1 层子单元的所有节点数据 $P(G_{n_0+1,G_i})_j$，$i,j=0,1,\cdots,6$ 可以由式(2)得到。其中式 2(a)是计算中心继承单元节点的过程，式 2(b～f)是计算偏心继承单元节点的过程。

整个计算过程在纵向上(从 $n\rightarrow n+1$)是一个层次递进的关系，从上一层单元的节点坐标必然

能够得到与其相关的所有后代单元的节点坐标；在横向上(同层次)是一个由内到外的过程，按照式2(a～f)的顺序计算节点坐标，即先计算中心继承单元，再计算邻近单元，这种方式所有子单元的节点仅需要计算1次，可以大大减少运算量。下面将格网生成从 $n\rightarrow n+1$ 层推广到 $n\rightarrow n'$ 层。根据HQBS编码的特点，利用单元子单元查找与邻近搜索算法，设计了单元填充(生长)算法，该算法在层与层间遵循式(2)的过程，同层之间遵循邻域生长规则。

$$
\begin{cases}
P(G_{n_0+1,G_0})_0=P(G_{n_0,G_0})_0,P(G_{n_0+1,G_0})_j=\dfrac{1}{2}(P(G_{n_0,G_0})_0+P(G_{n_0,G_0})_j) & \text{(a)}\\
P(G_{n_0+1,G_i})_0=\dfrac{1}{2}(P(G_{n_0,G_0})_{f(i,3)}+P(G_{n_0,G_0})_{f(i,4)}) & \text{(b)}\\
P(G_{n_0+1,G_i})_i=P(G_{n_0+1,G_0})_{f(i,4)'},P(G_{n_0+1,G_i})_{f(i,1)}=P(G_{n_0+1,G_0})_{f(i,3)} & \text{(c)}\\
P(G_{n_0+1,G_i})_{f(i,2)}=P(G_{n_0,G_0})_{f(i,3)'},P(G_{n_0+1,G_i})_{f(i,5)}=P(G_{n_0,G_0})_{f(i,4)} & \text{(d)}\\
P(G_{n_0+1,G_i})_{f(i,4)}=\dfrac{1}{2}(P(G_{n_0,G_{f(i,3)}})_0+P(G_{n_0,G_0})_{f(i,4)}) & \text{(e)}\\
P(G_{n_0+1,G_i})_{f(i,3)}=\dfrac{1}{2}(P(G_{n_0,G_{f(i,3)}})_0+P(G_{n_0,G_0})_{f(i,3)}) & \text{(f)}
\end{cases}
\tag{2}
$$

式中，$f(a,b)=\begin{cases}\{(a+b)/6\}, & [(a+b)/6]\neq 0\\ 6, & [(a+b)/6]=0\end{cases}$，[]和{}为取整和求余符号，$i=1,2,\cdots,6$。

由于偏心继承单元的存在，进行邻域生长的过程中需要将已经计算过的单元剔出，避免出现重复计算的问题。考虑从 n_0 层的单个单元 $G_0\rightarrow n$ 层的单元生长情况，HQBS单元无限填充(生长)算法如下。

步骤1：初始化数据。

(1) HQBS编码数组 $A[6]=\{12,21,23,32,32,31,13\}$，循环变量 $i=n_0$。

(2) 第 k 层单元数据集合用 $G_k=\{G_0\}$ 表示，包括单元的编码与单元的节点数据。

(3) 初始化 $G_{n_0}=\{G_0\}$，$C_{n_0}=\{0\}$；$G_{n_0+1\sim G_n}=$NULL；$C_{n_0+1}\sim C_n=$NULL。

步骤2：if $i<n$：循环变量 $j=0$，$k=0$，转到步骤3；else：返回第 n 层的单元集合 G_n。

步骤3：if $i\leqslant$card(G_i)：$G_i[j]$的中心子单元 $G=G_i[j]0$，利用式2(a)计算 G 的节点信息，加入到 G_{i+1} 中，$k=k+1$，初始化变量 $m=0$，转到步骤4；else：清空 G_i，令 $G_i=$NULL，$i=i+1$，返回步骤3。

步骤4：if $m<6$：$G=G\oplus A[m]$，转到步骤5；else：$j=j+1$，返回步骤3。

步骤5：if $G\notin G_{i+1}$：利用式2(b～f)计算偏心继承单元 G 的节点信息，加入到 G_{i+1} 中，$k=k+1$，返回步骤4；else：返回步骤4。

根据HQBS单元无限填充(生长)算法可以得到任意层次的单元集合，图5给出了依照该算法得到的从第 n_0 层单元生长出第 n_0+3 层单元集合过程中，计算顺序用0，1，2，…表示。

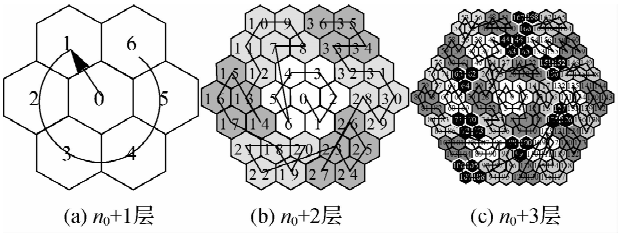

图5 格网动态生成过程中每一层单元的计算顺序

Fig. 5 The calculation order of dynamic generating grids each hierarchy

前面的讨论都是针对平面六边形格网的生成问题，当研究对象转变成球面六边形离散格网后，将有哪些变化？实际上，根据文献[17]中的讨论，利用球面离散格网系统逼近地球表面，当剖分层次大于10时，球面效果实际上就已经不是很明显了，因此，笔者以第10层作为转折点分段进行处理。可以事先用地图投影转换的方法将第10层的格网数据计算好，以文件的方式进行存储。针对不同层次的格网显示采用3种解决方案：

(1)当球面格网层次=10时，加载第10层的格网文件，利用裁剪算法进行单元裁剪即可，无需进行球面格网计算与生成；

(2)当球面格网层次<10时，加载第10层的格网文件，利用亚采样获得前面各层单元节点，建立单元间的对应关系，无需进行格网的计算与生成；

(3)当球面格网层次>10时，加载第10层的格网文件，利用式(2)计算深层次单元节点的坐标，与平面格网的生成方法一致。

3.2 可视区域裁剪

在全球格网系统可视化过程中，还需要确定

三维视场下的数据分辨率，即剖分深度的控制。本文采用场景凝视点（视线与球体的交点）处的格网分辨率来控制剖分的深度，满足当前层次凝视点处格网的大小在屏幕上的投影与像素大小最接近即可[19]。结合文献[1,11]的分析，三维可视化的视景体与球面相交成一个不规则的曲面八边形的可见区，为了计算简便，将其中的八边形简化成球面凸四边形，这样的处理虽然放大了裁剪区域，但减少了判断在每个单元可视区内的计算量，提高了计算效率。

对于视景体在球面上形成的可见区域，裁剪的目的就是判断哪些单元需要出现在显示序列中，这些单元如何显示。图 6 列出了可视区域边缘与六边形单元的几种位置关系。判断六边形单元是否属于该区域的关键就是判断单元中心与区域的隶属关系，这是因为具有多分辨率的 4HI 格网系统本身是一种上下层对准的格网，即低分辨率单元的中心必然是更高一层单元的中心，虽然在本层格网中选定区域内部的单元面积要小于选定区域外部的单元面积，但是更高分辨率的中心继承单元就不一定满足这样的情况。

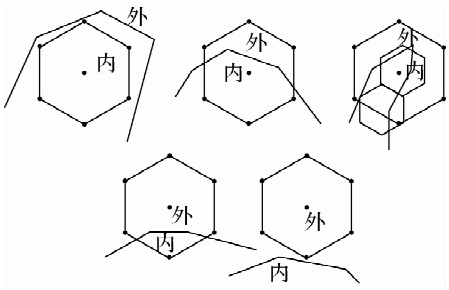

图 6　区域边缘与六边形单元的位置判断

Fig. 6　The ubiety between the edge and the hexagonal grid cell

判断单元中心是否在可视区域中的方法比较简单，通常采用过该点的射线与可视区边界相交的奇偶次数来判断，若交点次数为奇，则该点位于多边形内；交点数为偶，则该点位于可视区内。对于可视区四边形交点的特殊情况，遵循文献[20]中的“上闭下开”原则进行处理。

结合 HQBC 单元无限填充（生长）算法中格网的动态生成和剖分深度的控制，可以得到全球离散格网的可视区域裁剪方法，主要分为 3 步：① 根据视景体方位，控制视点凝视方向上格网的剖分深度 n；② 利用视景裁剪技术，计算裁剪区域的范围（球面四边形）；③ 在 HQBC 单元格网动态层次生长算法基础上增加一个初始条件和一个判断条件，就可以完成可视区域裁剪。初始条件包括最初层次中有多少个单元可以完整覆盖可视区，如果从第 0 层开始，只要判断第 0 层的单元需要哪几个就可以完整覆盖可视区。对于球面格网而言，初始层次的格网可以由 3.1 节的讨论得到。将判断单元中心节点是否在可视区域的条件加入到 HQBS 单元无限增长算法中，就可以得到区域约束下的单元生长（裁剪）方法（如图 7）。

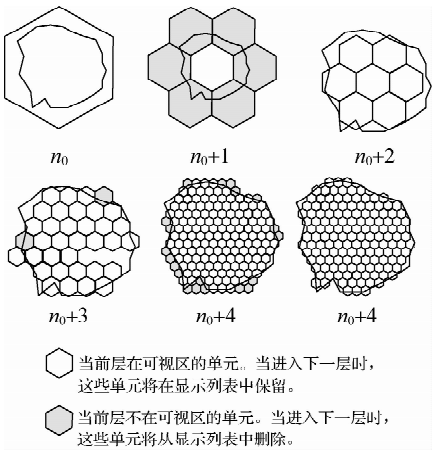

图 7　区域约束下的单元生长（裁剪）方法

Fig. 7　The method of growth for cells which were restricted within limits

4　六边形格网的实时显示

在全球格网实时显示的过程中，讨论 3 个基本的操作：放大、缩小与漫游。其中，放大和缩放操作可以归结为观察点到凝视点距离的变化，漫游操作归结为观察点位置和凝视点位置的变化。

（1）放大操作，是从低层次的格网到高层次格网的过程，这符合前面单元填充的特点，只不过是在场景放大中，可视区的范围 $(S_1S_2S_3S_4)_n$ 是随着视点的拉近而缩小的，如图 8（a）所示。因此，随着视场范围的变化，格网从 $n\rightarrow n+1$ 层的过程中，首先对于第 n 层的单元集合 G_n，重新判断各单元中心节点相对于可视区范围 $(S_1S_2S_3S_4)_{n+1}$ 的情况，将不属于该范围的单元从 G_n 中清除；再按照表 3 中 Step3～ Step5 的过程进行第 $n+1$ 层单元集合 G_{n+1} 的生成，其他单元填充过程与前面的方法类似。

（2）缩小操作，是放大操作的逆过程，可视区的范围 $S_1S_2S_3S_4$ 是随着视点的远离而放大，如图 8（b）所示。缩小操作同样是单元亚采样的过程，

与前面不同的是，可视区的范围在不断扩大，从第 $n \rightarrow n'$，$(n \geqslant n')$ 层格网的过程中，单元 $G_n \in$ 可视区 $(S_1S_2S_3S_4)_n$ 时的可以通过逐层亚采样得到 $G_{n'}$。对于单元 $(S_1S_2S_3S_4)_{n'} - (S_1S_2S_3S_4)_n$ 的情况，由于 $G_n \notin (S_1S_2S_3S_4)_{n'} - (S_1S_2S_3S_4)_n$，则采用亚采样的方式将无法进行，这就需要采用区域增长的办法，从初始层开始逐层填充。在 $(S_1S_2S_3S_4)_n - (S_1S_2S_3S_4)_{n+1}$ 中的区域进行单元生长，生长至最终的层次 n' 停止。还存在一些单元虽然 $\in (S_1S_2S_3S_4)_n$，但由于亚采样的过程中，找不到边界节点，因此，只能获得中心节点的数据，这将导致缩小到 n' 层后，部分处于 $(S_1S_2S_3S_4)_n$ 边界处的单元不完整。由于在生成过程中采用图 6 的单元归属判别方式，两个部分的单元在同一层必定能够无缝地拼接到一起，那些 $\in (S_1S_2S_3S_4)_n$ 而无法获取边界节点的单元，可通过编码运算，在 $(S_1S_2S_3S_4)_{n'} - (S_1S_2S_3S_4)_n$ 区域内查找邻近单元，获得对应方向的节点数据。

(3) 漫游，也是三维显示中一个重要的操作，在漫游的过程中，格网数据逐渐过渡，内存中的单元数据不断更新，这就需要每次计算可视区内的单元，如图 8(c)。重新填充区域可以解决这个问题，但同样会带来由于数据重复计算而导致的效率问题。这就需要设计一种在横向上逐渐过渡的单元数据更新方法，只计算更新部分的单元，而仍在显示区内的单元保持不变，这样就能最大限度地减少新增单元的计算量，保证显示效率。整个漫游过程中存在 3 种类型的单元：① 一直处于显示区的单元；② 新增的单元；③ 需要裁剪的单元。漫游过程中，单元的剖分层次保持不变，不存在亚采样或生成更高层次单元的需求。新增的单元，可以采用与缩小操作相同的逐层增长方式完成；对于删除的单元，也可以采用逐层判断的方式将出了显示区的单元裁剪掉。

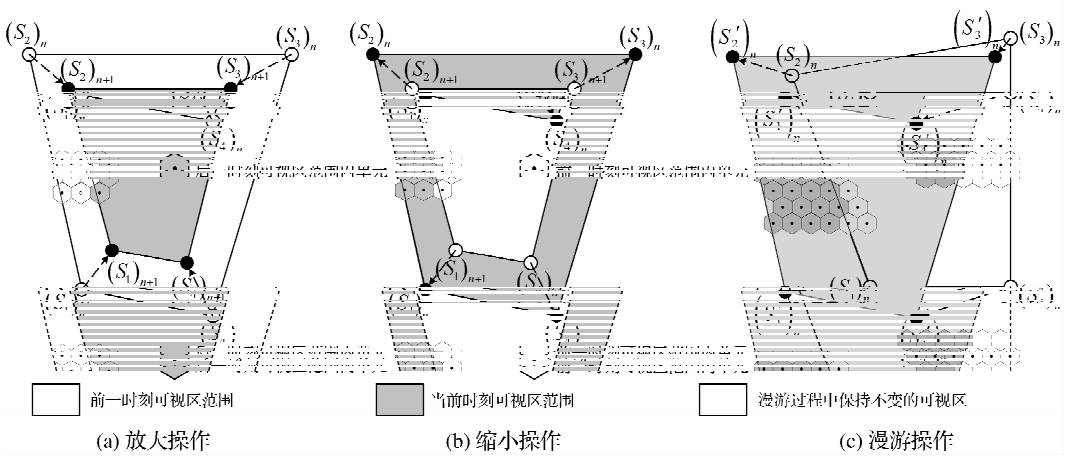

图 8 离散格网显示过程中可视区的变化

Fig. 8 Discrete grid shows the changes in the process of viewing area

5 试验与分析

下面给出六边形球面离散格网的三维可视化试验。格网的初始层数为 $N_0 = 10$，将该层次格网坐标数据预先算好[1]，作为二进制文件存储(保存全球格网对应二十面体一个面上的格网坐标即可)，导入后显示。对于格网层次 $n < N_0$ 的情况，利用亚采样方式获得前面各层单元的节点；当 $n > N_0$ 时，利用式(2)逐层递进内插。图 9 是 3 种情况下，全球六边形离散格网系统的显示效果。试验测试了全球格网动态生成的效率，以第 10 层格网为基础，分别生成第 7～13 六层格网的坐标数据。由于动态生成算法是层次算法，因此测试的顺序是分两个方向 10→9→8→7 和 10→11→12→13 进行的，试验结果如表 2。

表 2 不同层次全球格网动态生成的效率

Tab. 2 The efficiency of generating global grids dynamically in different levels

格网层次	单元数目	耗时/ms	效率/(单元/ms)
9	1 474 562	13 579	108.591 4
8	368 642	3 248	113.498 2
7	92 162	787	117.105 5
11	23 592 962	62 683	376.385 3
12	94 371 842	255 358	369.566 8
13	377 487 362	1 029 419	366.699 4

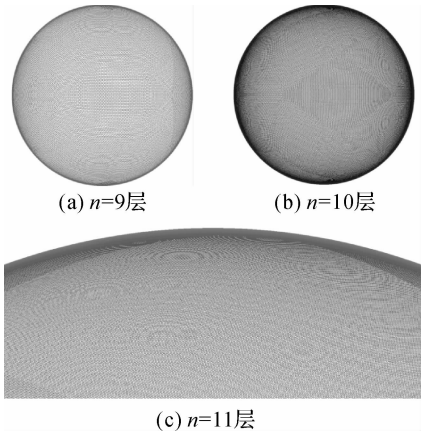

图 9　不同层次全球六边形离散格网系统的显示

Fig. 9　The discrete global grid system shown in different levels

试验环境：ThinkPad T61，CPU Intel(R) Core(TM)2 Duo，0.98 GB 内存，7200 转硬盘，WinXP 操作系统，Visual Studio C^{++} 2008 编译环境。

分析试验结果发现，格网动态生成的效率在第 10 层之后反而比之前的层次要高。进行亚采样的离散格网实际上是不需要进行坐标计算的，因此运算时间的消耗只是子单元查找的过程，对于第 n 的格网生成，实际上只是执行了 $45 \cdot 2^{2n-3}+2$ 次子单元查找运算；而对于 $n>N_0$ 层的格网生成，除了需要执行 $45 \cdot 2^{2(n-1)-3}+2$ 次邻近单元查找外，考虑单元间节点的共用性，按照式(2)需要计算 6/2+(3×6)/3=9 个点的坐标(取中点)，共生成了 $45 \cdot 2^{2n-3}+2$ 个单元，这样理论上效率应该不到亚采样过程的 4 倍，试验的结果也证明了这一点。

为了测试全球离散格网加载空间数据后的显示性能，选取了下面数据集进行测试：① 全球 GTOPO30 高程晕渲数据，采样点数 43 200×21 600，采样间隔 0.008 333 33°，6.95 GB；② 黄河小浪底库区的多光谱融合图像数据和 DEM 数据，采样点数皆为 10 764×8 812，采样间隔 25 m，数据量 271 MB(图像)+361 MB(DEM)；③ 郑州市 QuickBird 图像，全色波段，地面分辨率 0.61 m，采样点数 33 837×32 272，8.14 GB；④ 全球大陆的矢量边界数据，9.90 MB；⑤ 全国县级行政区划矢量数据，17.3 MB。图 10 中是不同类型空间数据在球面离散格网上的显示情况。表 3 统计的是不同层次全球离散格网上遥感图像数据+矢量数据显示时部分指标的比较，对应图 11 的显示。

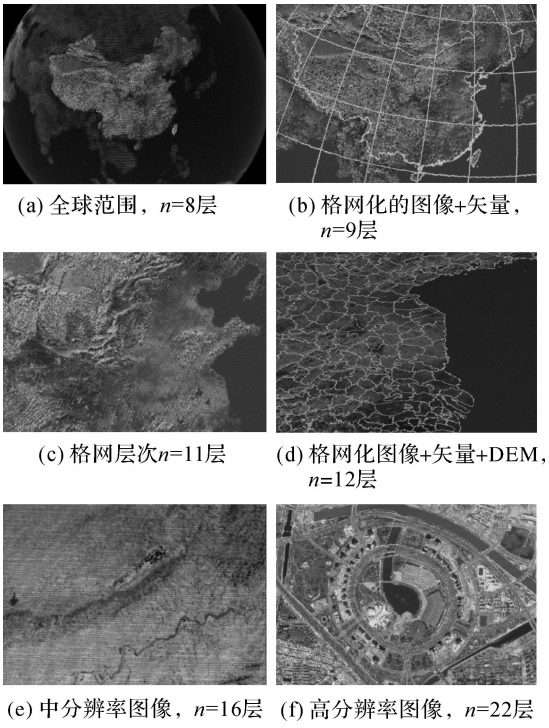

图 10　不同类型空间数据在球面离散格网上的显示

Fig. 10　Different types of spatial data shown in discrete global grid system

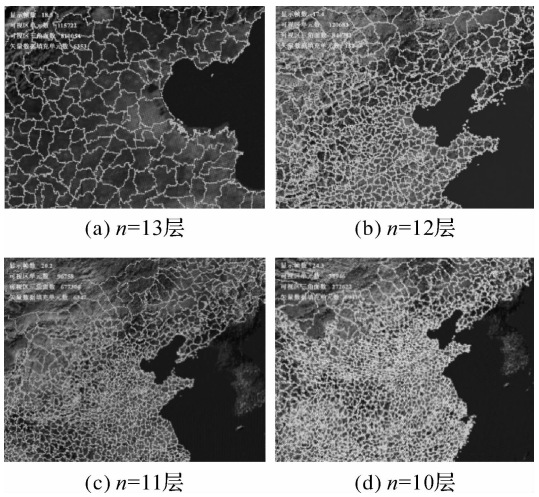

图 11　不同层次全球离散格网上遥感图像数据+矢量数据显示时指标的比较

Fig. 11　The display indicators of the RS image and vector data shown in the discrete global grid of different levels

通过本节的三维显示试验，可以得到以下结论：① 本文设计的全球六边形格网空间数据显示算法的平均刷新率在 20 帧/s 左右，能够保证实

时显示的需要;② 可视区内的单元数目基本稳定在 100 000 左右,可视区内的三角面数目稳定在 700 000 个左右;③ 初始层次 $n=10$ 的加载时间在440 ms左右。由于可视区内的单元数目相对稳定,因此,格网数据和空间数据的逐层加载(从 n 层到 $n+1$ 层)时间也相对稳定,大约在 300 ms 左右。数据加载时间包括格网数据生成、矢量数据填充和栅格数据加载的耗时总合,初始层次需要进行矢量数据的填充和栅格数据的采样,其余层次数据加载的耗时主要是格网从 n 层到 $n+1$ 层的动态生成耗时以及矢量数据综合的耗时,栅格数据的简化与格网动态生成过程同步,基本不耗时。

表 3 不同层次全球离散格网上图像+矢量数据的显示比较

Tab. 3 The display of the image and vector data shown in the discrete global grid of different levels

格网层次	10	11	12	13
视点高度/km	2 127.03	1 536.35	1 036.59	514.05
可视区单元数	38 946	96 758	120 683	115 722
可视区三角面数	272 622	677 306	844 781	810 054
矢量填充单元数	5 941	6 347	15 240	6 353
格网+空间数据的加载时间/ms	442(直接加载)	261(10～11 层)	335(11～12 层)	319(12～13 层)
显示帧数	24.3	20.2	17.4	18.5

参考文献:

[1] ZHANG Yongsheng, BEN Jin, TONG Xiaochong. Discrete Global Grids for Geospatial Information: Principles, Methods and Its Applications[M]. Beijing: Science Press, 2007. (张永生,贲进,童晓冲. 地球空间信息球面离散网格——理论、算法及应用[M]. 北京:科学出版社,2007.)

[2] ZHAO Xuesheng, BAI Jianjun, WANG Zhipeng. An Adaptive Visualized Model of the Global Terrain Based on QTM[J]. Acta Geodaetica et Cartographica Sinica, 2007, 36(3): 316-320. (赵学胜,白建军,王志鹏. 基于 QTM 的全球地形自适应可视化模型[J]. 测绘学报,2007,36(3):316-320.)

[3] KIDDR A. NWP Discrete Global Grid Systems[M]. Australia: [s. n.], 2005.

[4] VINCE A, ZHENG X. Arithmetic and Fourier Transform for the PYXIS Multi-resolution Digital Earth Model[J]. International Journal of Digital Earth, 2009, 2(1): 59-79.

[5] GOODCHILD M, YANG S. A Hierarchical Spatial Data Structure for Global Geographic Information Systems[J]. Graphical Models and Image Processing, 1992, 54(1): 31-44.

[6] KIDD R A, TROMMLER M, WAGNER W. The Development of A Processing Environment for Time-series Analysis of Sea Windsscatterometer Data[C]// International Geoscience and Remote Sensing Symposium. Vienna: [s. n.], 2003: 4110-4112.

[7] DAVID A, TODD D, ROSS P. Climate Modeling with Spherical Geodesic Grids[J]. Computing In Science & Engineering, 2002,4(5): 32-41.

[8] SAFF E B, KUIJLAARS A. Distributing Many Points on a Sphere[J]. Mathematical Intelligencer, 1997, 19(1): 5-11.

[9] KIMERLING A J, SAHR K, WHITE D, et al. Comparing Geometrical Properties of Global Grids[J]. Cartography and Geographic Information Science, 1999, 26(4): 271-87.

[10] GONG Jianya. Data Processing and Analysis of Earth Observation[M]. Wuhan: Wuhan University Press, 2007. (龚健雅. 对地观测数据处理与分析[M]. 武汉:武汉大学出版社,2007.)

[11] DU Ying. A Research on Key Technologies for Global Multi-resolution Virtual Terrain Environment [D]. Zhengzhou: Information Engineering University, 2005. (杜莹. 全球多分辨率虚拟地形环境关键技术的研究[D]. 郑州:信息工程大学,2005.)

[12] BERNARDIN T, COWGILL E, KREYLOS O, et al. Crusta: A New Virtual Globefor Real-time Visualization of Sub-meter Digital Topography at Planetary Scales[J]. Computers and Geosciences, 2010,37(1):75-85.

[13] SUBNER G, DACHSBACHER C, GREINER G. Hexagonal LOD for Interactive Terrain Rendering[C]// Proceeding of 10th International Fall Workshop Vision, Modeling and Visualization. Erlangen:[s. n.], 2005: 16-18.

[14] SAHR K. Icosahedral Modified Generalized Balanced Ternary and Aperture 3 Hexagon Tree: United States, 8229237[P]. 2012-07-24.

[15] PETERSON P. Closed-packed Uniformly Adjacent, Multi-resolutional Overlapping Spatial Data Ordering: United States, 8018458[P]. 2011-09-13.

[16] TONG Xiaochong, BEI Jin, WANG Ying, et al. Efficient Encoding and Spatial Operation Scheme for Aperture 4 Hexagonal Discrete Global Grid System[J]. International Journal of Geographical Information Science, 2012(3): 1-24.

[17] TONG Xiaochong. The Principles and Methods of Discrete Global Grid Systems for Geospatial Information Subdivision Organization [D]. Zhengzhou: Information Engineering University, 2010. (童晓冲. 空间信息剖分组织的全球离散格网理论与方法[D]. 郑州:信息工程大学,2010.)

[18] TONG Xiaochong, BEN Jin, QING Zhiyuan, et al. The Subdivision of Partial Grid Based on Discrete Global Grid Systems[J]. Acta Geodaetica et Cartographica Sinica, 2009, 38(6): 506-514. (童晓冲,贲进,秦志远,等. 基于全球离散网格框架的局部网格划分[J]. 测绘学报,2009,38(6):506-514.)

(下转第 403 页)