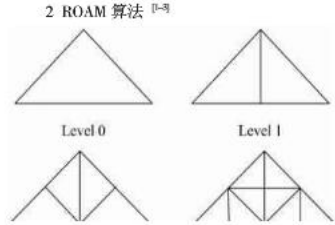


基于 DEM 三维海底地形的漫游系统在虚拟现实、地理信息系统以及军事相关的应用中扮演着重要的角色。然而在三维漫游过程中, 因受硬件资源的限制, 超量的数据成为地形三维场景实时绘制的最大瓶颈。对于地形建模而言, 数据如何简化成为解决问题的关键。新的地形可视化技术, 为地形可视化问题的解决, 取得了很大的进展。目前国内外对地形生成算法的研究主要集中于多分辨率静态地形生成, 其中比较主流的有 ROAM、GeoMipmapping、Chunk LOD、四叉树算法等。

1 多分辨率静态地形生成算法的选择
已有的多分辨率静态地形生成算法, 一般都采用一种根据相应的误差判定原则, 提供一系列连续的近似模型的层次树结构。这种结构主要用来标识近似模型间的层次关系。由于动态地形需要频繁地改变地形的几何属性, 而现有的很多多分辨率静态地形生成算法都无法满足这一要求。因此, 需要选择一种合适的多分辨率静态地形生成算法, 并对其作出扩充, 以使其适合动态地形的实时可视化。

在前期对地形可视化算法总体研究及动态地形可视化方案确定的基础上, 考虑基于不规则三角形网格 (Triangulated Irregular Networks, TINs) 表示的地形生成算法的复杂性, 而基于等腰直角三角形规则网格 (Right Triangulated Irregular Networks, RTINs) 表示的 ROAM (Real Time Optimally Adapting Meshes) 算法构建地形模型的近似序列的层次结构时不用考虑顶点的高程位, 并且三角形网格之间可以实现无裂缝和无 T-连接, 该算法的众多优良特性, 使其最适合于静态地形的动态扩充。



2.1 三角形二叉树结构
ROAM 算法要求对三角形二叉树结构进行操作, 这个结构可以看作是把这个地形块 (patch) 切分为两个同样大小的三角块。每个三角块就是一个二元三角树的根节点。每个 Patch 被切分为两个左右基三角形后, 控制各分块和相邻分块之间的数据位置, 进行数据操作。把每个等腰直角三角形进行分裂, 产生层次化的效果, 即生成三角形二叉树, 如图 1 所示。

虽然将整个地形分块进行处理, 但是地形作为一个整体, 所以有必要把地形的每个单元有机的联系在一起, 这样可以快速的遍历各个节点, 按照给定的条件选择合适的分辨率。ROAM 算法为我们提供了五种基本的邻接关系, 如图 2 所示。

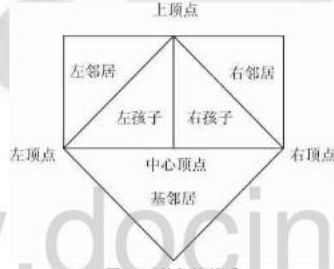


图2 五种邻接关系
Fig.2 Five neighboring relation
这五种邻接关系的数据结构表示如下:

```
struct TriTreeNode  
{  
    TriTreeNode *LeftChild;  
    TriTreeNode *RightChild;  
    TriTreeNode *BaseNeighbor;  
    TriTreeNode *LeftNeighbor;  
    TriTreeNode *RightNeighbor;  
};
```

2.2 地形分辨率的判定
ROAM 中, 决定地形分辨率的大小有两个重要的条件: 一个是视点距离地形的距离; 一个是地形本身的粗糙程度。根据这两个条

为地图的大小。计算出的 v 值与预设的一个可接受的值进行比较, 如果 v 小于预设值, 当前的分辨率满足我们的要求, 可以按照该分辨率进行显示, 否则, 要对地形块继续进行分割。

2.3 裂缝的处理
当分割在地形上相邻的二元三角树时, 如果分割线穿过 Patch 边界, 这个裂缝是由于不断的分割穿过 Patch 边界的树造成的, 如图 3 所示。

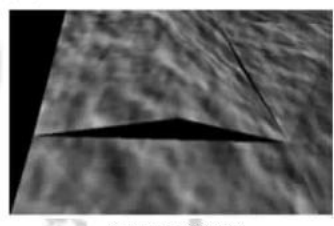


图3 网络中的裂缝
Fig.3 Crack of network

为解决这个问题, ROAM 使用了网格本身关于节点的一个规律: 一个节点与他的邻节点只存在两种关系, 共直角边关系 (如左、右节点) 和共斜边关系 (如下邻节点), 在建立网络模型时使用这一原理可以保持相邻的树与当前分解同步, 当分解一个节点时, 具体的实现过程为:

- ① 节点是当前规则格网的部分, 即当前的规则格网已经存在, 只需要分解节点和它的下邻节点。
- ② 节点是网络的边界, 即不可能组成新的规则格网, 只分割该节点。
- ③ 节点不是当前规则格网的部分, 即需要通过首先分割更大的格网生成该节点所需要的规则格网, 强制分解下邻节点逐级扩大分割范围, 直到当前需要分割的节点满足 ① 为止。

这种避免产生裂缝的方法在 ROAM 中被称为强制分割。图 4 为按照以上三个原则进行强制分割的示意图。

3 基于 ROAM 算法的海底地形可视化系统的设计与实现 [4-6]

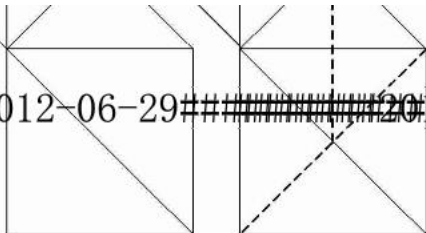


图4 强制分割示意图
Fig.4 Force split

态生成不同分辨率的三维地形场景。用户可以通过鼠标、键盘的操作,实现不同绘制方式、不同高度、不同角度的实时漫游。图5给出了该系统的系统构成。

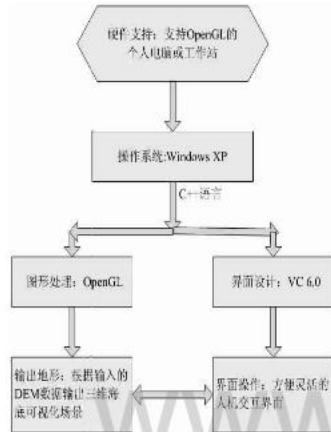


图5 系统构成
Fig.5 Compose of system

3.2 系统的功能模块

系统按功能划分为三大模块,如图6所示,下面对各个模块的实现分别加以说明。

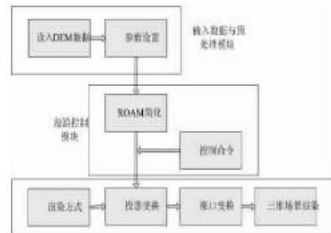


图6 系统的组成模块
Fig.6 Module of system

3.2.1 数据载入及预处理模块

主要进行 DEM 数据文件的读取,进行高程数据的预处理,包括 LOD 模型库的建立。

次装入大规模的地形数据显然不切实际,所以必须对原始地形数据进行分块处理。该系统将数据以“块”为单位,执行数据量简化及场景绘制操作。渲染时,依据视点的位置导入对应块的数据到内存,在漫游过程中,动态释放不可见的子块和调度新的子块。这样处理起来方便灵活,将场景由大化小,对场景的操作难度也相应的降低,从而获得简化与绘制效率的提高。

3.2.2 显示模式设置模块

该模块包含了纹理模式、光照模式、网络模型三种可选的显示模式,以满足不同的观察需要。其中,纹理模式对场景进行了纹理贴图,得到的是具有真实感的场景模拟;光照模式根据场景中节点的高程大小确定其颜色亮度程度;网络模型便于进行数据简化效果的观察和分析,这种方式是对 ROAM 算法的运用的完整体现。对于每一种不同的渲染方式,都需要通过 OpenGL 中为我们提供的 glEnable 0 函数开启相应的映射方式,同时也需要用 glDisable 0 函数关闭其他的两种。图7对分别对应了这三种地形渲染方式,其中的地形为我国东海部分海域的海底地形,纹理采用单一的绿色。



④ 纹理映射模式
④ Texture mapping mode



④ 光照模式
④ Illumination mode



④ 网络模式
④ Network mode

图7 三种显示模式
Fig.7 The three display modes

3.2.3 漫游实现模块

该模块为用户提供了与场景交互的接口。通过键盘鼠标等外部设备的输入来改变视点的位置,进而控制地形显示的变化,实现多方位多角度的观察。OpenGL 定义了一个名为 gluLookAt 0 的函数,利用该函数,可以实现一系列的旋转、移动的功能。从而实现漫游的效果。

4 结论

大规模地形可视化技术研究是近十几年发展起来的热点问题,利用 ROAM 算法实现了一个基于 DEM 海底地形数据的三维地形可视化仿真系统,并对我国东海部分海域的海底地形进行了实时显示。经过测试,在硬件条件为 PIV3.0G, 256 内存, NVidia GeForce4 MX400 显卡;软件条件为 Windows XP pro, Visual C++6.0, OpenGL2.0;数据大小为 140M (4198401 个高程点)的情况下,地形的浏览速度可以达到 45~65FPS,每一帧所绘制的三角形数量在 1000~5000 之间。通过鼠标与键盘的操作,很好地实现了地形的实时漫游。

参考文献

- [1]Lindstrom, P., Koller, D., Ribarsky, W., Hodges, L., Faust, N., Turner, G., " Real-Time Continuous Level of Detail Rendering of Height Fields"
- [2]Duchaineau, M. et al. ROAMing Terrain: Real-Time Optimally Adapting Meshes." IEEE Visualization'97.81~88. Nov.1997 [EB/OL].http://www.llnl.gov/graphics/ROAM.
- [3]Turner, Bryan. Real-Time Dynamic Level of Detail Terrain Rendering with ROAM. [EB/OL].http://www.gamasutra.com/features/2000/04/03/turner_01.htm.
- [4]美]Dave Shreiner. OpenGL 编程指南 [M].北京:机械工业出版社,2001.
- [5]苏虎,周美玉.一种大规模地形的实时绘制算法[J].武汉大学学报(工学版),2003,36(3):81~85.
- [6]郝燕玲,路辉.基于 OpenGL 海底地形显示的研究[J].计算机仿真,2003(20):10.

作者简介:王奎民(1971~),男,博士研究生。

责任编辑:杨帆

sdoiw ch jyeio jlspe;ksm clkj sl de jpw j jheisda j ksk kxm kksx p jW IU DH YU W EG FYU W G EPLUH SD K JN CK JSD N L K JH L IU SH D L IU H L KU H L KU huihu igd iu lth x jv ksdm fy kc jslanthy iheo d j kq j fthskah fllk h sd kch iksocyuvsayucv l ne sd j od f j safo v j

