

DOI:10.16356/j.1005-2615.2015.01.008

临近空间平台下地形四叉树优化分割算法

关卓威 张 晔

(哈尔滨工业大学图像信息技术研究所, 哈尔滨, 150001)

摘要:提出一种利用四叉树算法生成临近空间平台下动态地形的新方法,并提出了一种新的四叉树递归分割算法的实时优化算法,利用可见性剔除的简化策略和数据简化的存储方式,解决地形绘制的裂缝问题。通过对该算法的实现和优化,在保证一定地形环境的视觉真实程度前提下,达到提高实时渲染速度的目的。实验结果表明:采用本文提出的四叉树算法可以快速对地形数据进行网格剖分,且可得到较好的剖分效果。

关键词:临近空间;四叉树;地形绘制;网格剖分

中图分类号: TP79 **文献标志码:** A **文章编号:** 1005-2615(2015)01-0059-05

Terrain Quadtree Segmentation Algorithm Optimization Based on Near Space Platform

Guan Zhuowei, Zhang Ye

(Institute of Image & Information Technology, Harbin Institute of Technology, Harbin, 150001, China)

Abstract: A new method for generating dynamic near space platform terrain based on the quadtree algorithm is put forward. And a new real-time optimization algorithm for the recursive quadtree segmentation algorithm is proposed. The algorithm uses the strategy of visibility culling and the storage way of data simplification to solve the problem of the cracks in the terrain rendering. Through the implementation and optimization of the algorithm, the real-time rendering speed is improved as long as the visual truth degree of terrain environment is guaranteed. The experimental results show that the proposed quadtree algorithm can rapidly generate the grid subdivision of the terrain data, and get the better mesh result.

Key words: near space platform; quadtree; topographic plotting; grid subdivision

临近空间,又称近空间,指 20~100 km 的高空,这个定义是以战略意义和实用角度为出发点,而不是直接由特性得出,所以研究临近空间,应该从实际要求出发,指明具体研究哪部分空间。现存的作战空间包括深海/浅海/水面/陆地/空中/地球轨道,临近空间的补入填补了空天联合作战的空白区域,与空天信息均能产生良好互动,具有重大的战略意义。而对平台所得到的系列图像进行三维重建,则可以结合目标检测等技术对战场指挥、情报侦察予以有效反馈。

本文讨论的三维重建属于表面重建,不需要获取目标的内部结构细节。三维重建中最常用的方法为网格剖分,网格剖分中最常用的算法为三角化算法。三角化算法虽然很多,但大多算法生成的是 Delaunay 三角网格,即以空外接圆准则为优化准则。经典的 Delaunay 网格剖分算法有 Bowyer 算法^[1], Watson 算法^[2], Bowyer-Watson 的联合划分算法等。为了克服有时不能准确表示地形的结构与细节特征的缺点,可通过附加地形特征数据,如地形特征点、山脊线、山谷线等,提高局部细节,从

收稿日期:2014-01-26;修订日期:2014-06-17

通信作者:张晔,男,教授,博士生导师,E-mail:zhye@hit.edu.cn。

而构成完整的地形地物数据模型^[3]。本文采用的网格剖分方法正是基于这种思想,利用 Delaunay 三角剖分和规则网格剖分相结合的方式建立地形地物模型,进而更好地表现目标及场景的特征。

四叉树网格兼具结构网格和非结构网格的特性,一方面具有结构网格的正交性特点,另一方面具有非结构网格的灵活性特点,便于网格的自适应化,进而方便处理复杂的边界问题。本文提出的新的四叉树算法得到的三维真实感地形不仅形状各异,而且实时动态变化。不必进行大量数据处理和附加计算,用少量的数据即可得到复杂的地形数据的逼真图像。

1 基于四叉树的动态地形生成算法

1.1 四叉树算法

树形结构是一种非线性结构,它在查找算法中有独特的优势。若将采样数据以某种形式划分,形成四叉树,缩小查找区域,减小查找数据量,将会提高查找效率。

建立采样数据四叉树结构的基本思想是:首先确定阈值,检查区域内采样数据个数是否超过阈值;若超过,则将该区域四等分,然后对4个子分区进行同样的处理,直到所有子分区内的采样数据个数均不超过阈值为止。具体操作为:设 n_q 为预先给定的阈值,若采样区域内采样数据个数 $n > n_q$, 则将该区域四等分,得到4个子区。若4个子区中存在大于 n_q 的子区,则再将其四等分,直到所有子区的采样数据个数小于 n_q 为止,由此得到的树即为采样数据的四叉树结构^[4]。在此四叉树中,只有叶子结点含有采样数据。

在采样区域内随机分布着一系列采样数据点,当取 $n_q = 10$ 时,其采样数据的划分如图1所示。图2为图1中采样数据的四叉树表示,“○”为四叉树结点,“□”表示对应结点(即子区)所含的采样数据。在子区内,由于数据量已控制在阈值范围内,因此为简便起见,一般以线性链表存放。

地形绘制是四叉树算法的核心,用来完成三角形网格的所有绘制,算法流程如图3所示。

1.2 改进的四叉树算法

Ulrich 提出了一种层次细节算法^[5],该算法目的是在现代图形硬件上高效地渲染海量地形数据。本文算法借鉴了 Geomipmap 的一些思想^[6],但做了较大的改进。该算法基于四叉树结构。四叉树的每个节点覆盖某一块区域,形成一个连续继承的

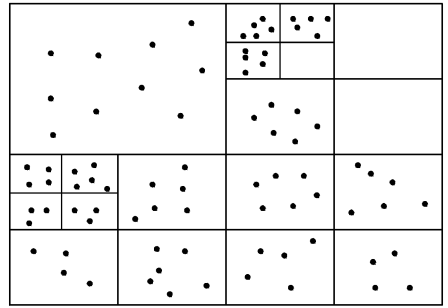


图1 采样区内采样数据的划分

Fig. 1 Division of sampling data in sampling area

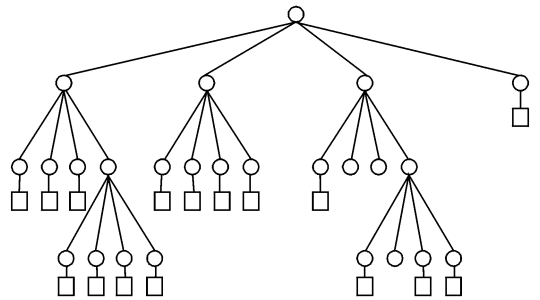


图2 采样数据的四叉树表示

Fig. 2 Quadtree representation of sampling data

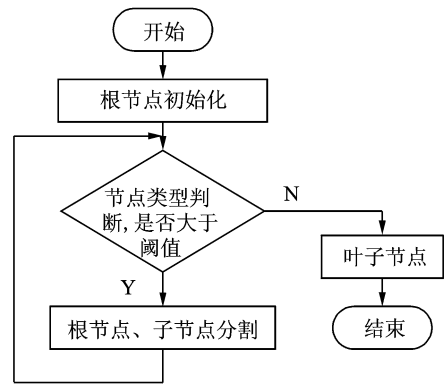


图3 地形的绘制流程

Fig. 3 Triangulating process of terrain

关系。根节点有一个最低分辨率的多边形网格覆盖整个地形,其4个子节点分别覆盖1/4地形,但分辨率更高。地形递归细分,直到得到原始分辨率的地形。

地形数据多边形网格存储在四叉树的每个子节点,称之为“地形块”。每个地形块与其他地形块的关系相互独立。图4是一个三层四叉树地形块的示意图,分别代表父节点、4个子节点、16个孙子节点。

每个节点即地形块有一个最大几何误差。与 Geomipmap 一样,几何误差就是当前数据与原始

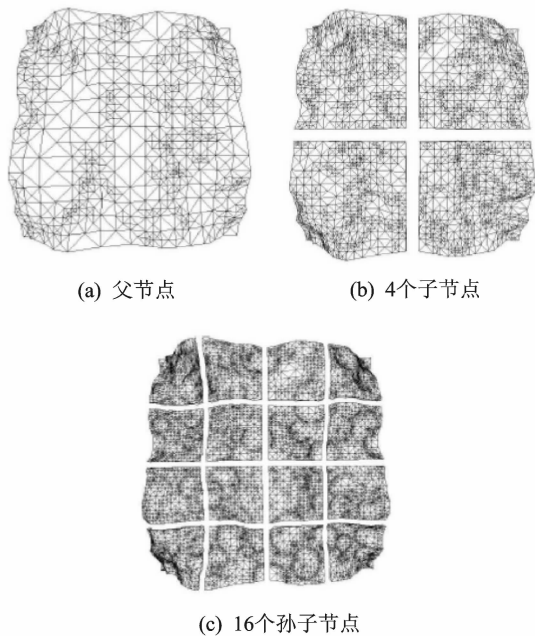


图 4 瓦片四叉树

Fig. 4 Tiles quadtree

数据在物方空间的最大背离。当地形块被构建时,它们的误差值随着每一层的细分而减半。例如一个节点有误差值 16,则其子节点误差值为 8,其他层次节点误差以此类推。

要选择合适的细节层次需要从四叉树的顶部开始递归。因为在 Geomipmap 算法中是假设视点沿着水平面移动,所以每个组块节点的物方空间误差被投影到屏幕空间得到误差。

离观察视点越远的地方细节越多,即节点变长越小^[7]。视点与节点中心的距离 d 与节点边长 e 成正比不等关系,设 C 为一个可调节常数:距离分辨率,则当 $d < e \times C$ 即 $d/e < C$ 时, e 需要继续分割。 C 越大则在 d 确定情况下节点边长 e 需要取得更细小,从而令节点处细节信息保留丰富;反之, C 越小, e 可取得大一些,无妨令细节粗略些。

如上所述,本文可以通过增加更多的采样值来为数据增加更多的细节。因为本文算法的点云数据存储在瓦片四叉树中,为了增加更多的采样值,四叉树必须用新的瓦片扩展。因为本文目的是实时扩展四叉树,所以输入的可能是低分辨率的点云数据,但是映射时可能会具有较高的细节层次^[8]。

1.3 裂缝处理

在对地形进行渲染时,如果相邻子块节点的分辨率不一致,则在构建三角形网格时,就会在相邻的边界上出现裂缝,这是地形绘制时必须处理的问题。针对块与块之间的裂缝问题,国内许多学者采

用限制相邻节点层次差不超过 1 的方法来消除裂缝,但是这种方法只适用于少量静态地形数据的场合。但在处理海量地形数据的过程中,要不断对节点的层次差进行判断,这个过程不仅计算量大,而且 CPU 处理时间比较长^[9]。为此本文提出了一种新的方法来消除裂缝。根据投影长度对规格网格进行重采样,使得边界采样点的个数与投影的长度成正比,网格内部采样点依据边界情况均匀过渡,使得均匀网格变成一种密度均匀变化的不规则网格,这样不仅可以消除裂缝,而且还大大减少了绘制图形的复杂程度,同时也不会影响观察的效果^[10]。结合节点分割和渲染规律,提出一种裂缝消除方法。算法基础是视点到节点中心距离较大时细节较少,因此裂缝消除通过缩减边的方式实现;而距离较小时需观察到的细节较多,通过剖分可使细节变得更丰富^[11]。算法流程如下:

(1)遍历过程中两个节点间层级相差一倍以上的情况;

(2)判断节点视点距离 d 与给定阈值 m 的大小;

(3)若 $d > m$ 则缩减高精度节点,重新设置分割标志;

(4)若 $d \leq m$ 则剖分低精度节点,重新设置分割标志;

(5)重复步骤(2~4)直至 1 所对应的节点全部剖分。

2 算法实现及实验结果

本文以某遥感数据网站提供下载的美国某地区地形数据为例^[12],如图 5 所示。

该地形数据包含 8 770 495 个点,对其进行了网格剖分实验。由于该地形数据包含的点云数据

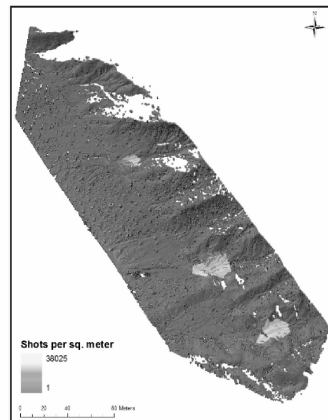


图 5 美国某地区地形数据

Fig. 5 Terrain data of given area in USA

量较大,直接使用 Delaunay 三角剖分,会使得地形的细节无法很好地显示出来,采用上述的四叉树原理,首先将水平方向和垂直方向二等分,得到 4 个子区,本文中以 1 300 000 个点为界限,如果该子区内的点小于等于 1 300 000 个,则该子区不需要进行继续划分,可以直接进行剖分;反之则将该区域继续四等分,直至所有的子区的点都小于 1 300 000 个为止。其采样后的子区划分结果如图 6 所示。每个子区内点的个数如表 1 所示。

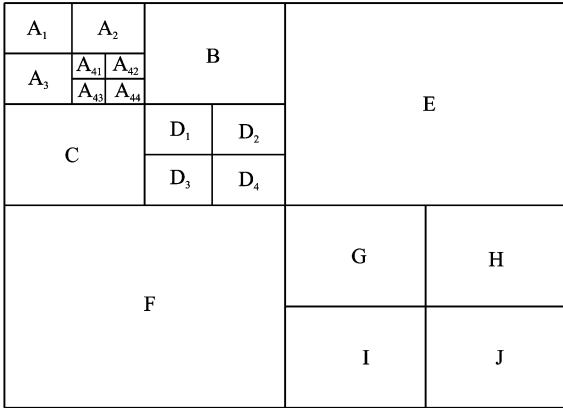


图 6 采样后子区划分结果

Fig. 6 Sub-area division results after sampling

表 1 每个子区内点的个数

Tab. 1 Number of points in each partition

子区	A ₁	A ₂	A ₃	A ₄₁	A ₄₂	A ₄₃	A ₄₄
点数	31 229	43 694	130 690	276 184	884 795	2761 20	538 941
子区	B	C	D ₁	D ₂	D ₃	D ₄	E
点数	104 097	667 070	1206 722	231 253	745 120	271 673	124 715
子区	F	G	H	I	J		
点数	1 266 803	1 093 160	74 866	531 166	272 197		

图 7 为采用经典 Delaunay, 剖分方法得到的结果。采用四叉树算法得到的图像如图 8 所示。

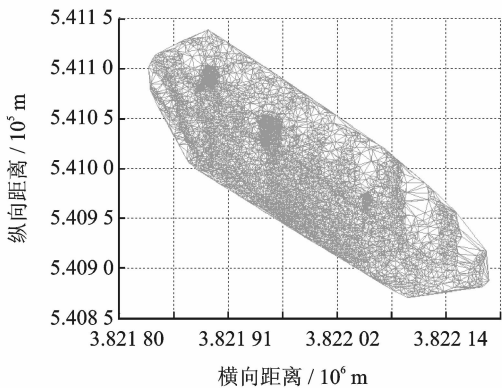


图 7 经典 Delaunay 方法剖分结果

Fig. 7 Meshing result using classical Delaunay method

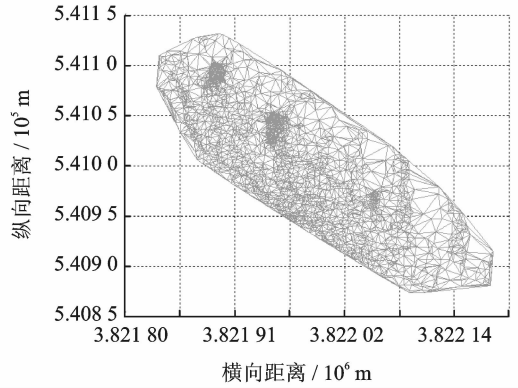


图 8 四叉树算法剖分结果

Fig. 8 Meshing result using quadtree Delaunay method

将图 7,8 比较可以看出, Delaunay, 剖分方法得到的结果采样密度较大, 层次感不够明显, 且采样时间长。而采用本文提出的算法剖分后得到的结果层次清晰, 可以更好地显示地形的层次结构。

分析实验结果可以看到: 由于在瓦片四叉树中每个瓦片数据块是完全独立于其他数据块甚至于树本身, 因而数据的调度简单, 调度效率高; 使用四叉树结构简化了算法的实现过程; 通过细化处理瓦片数据, 提高了数据的网格绘制效率, 在 INTEL i5 2.53 GHz 双核系统 Matlab 2010b 未优化环境下, 运行时间从整体数据时的 1 283.498 s 减到了四叉树分块数据时的 79.374 s, 缩短了 93.8%。

接下来针对临近空间平台半实物仿真地形数据进行同样四叉树优化分割算法的网格剖分实验。采样后子区内点的个数如表 2 所示。地形原图如图 9 所示。对图进行分区, 得到的分区结果如图 10 所示。

同理, 将本文方法与经典 Delaunay 剖分方法进行剖分时间的比较, 得到的结果如图 11 所示。

可以看出本文所提出的基于四叉树原理的地形分治三角剖分算法比起原有算法在速度上有大幅提升, 且这一优势在处理更多数量点云数据时愈加明显。

表 2 地形图采样后每个子区内点的个数

Tab. 2 Number of points in each partition of topographic map after sampling

子区	A	B	C	D	E	F	G
点数	21 928	13 893	18 392	23 495	34 215	15 298	14 928
子区	H	I	J	K	L	M	
点数	20 368	19 289	16 378	14 089	12 302	11 483	



图 9 半实物仿真地形图

Fig. 9 Topographic map of semi-physical simulation system

A	B	E	
C	D	E	
F	G	J	K
H	I	L	M

图 10 地形图采样后子区划分结果

Fig. 10 Sub-area division results of topographic map after sampling

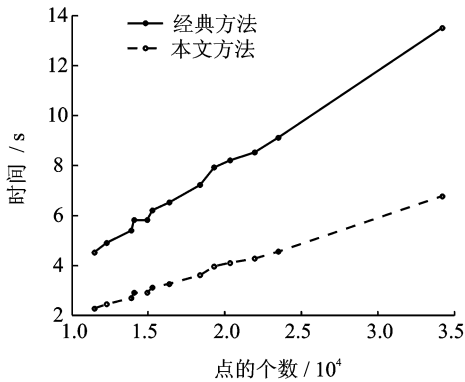


图 11 两种方法的剖分时间比较

Fig. 11 Comparison of subdividing time cost

3 结束语

本文提出了一种分形地形的新的四叉树生成算法,仿真图与仿真数据较好地体现了其优越性,能够随机性模拟地形和四叉树的层次细节,实时绘制地形。该算法在地形绘制速率和真实感效果上也有提升,实现了地形的快速剖分,且剖分效果良好。

参考文献:

- [1] Bowyer A. Computing dirichlet tessellations[J]. The Computer Journal, 1981, 24(2): 164.
- [2] Watson D F. Computing the n -dimensional Delaunay

tessellation with application to Voronoi polytopes[J]. The Computer Journal, 1981, 24(2): 170.

- [3] Mekni M. Using GIS data to build informed virtual geographic environments (IVGE)[J]. Journal of Geographic Information System, 2013, 5(6): 550-551.
- [4] McMorris H, Kallinderis Y. Octree advancing front method for generation of unstructured surface and volume meshes[J]. AIAA Journal, 1997, 35(6): 978-980.
- [5] Ulrich T. Continuous LOD terrain meshing using adaptive quadtrees[R/OL]. http://www.gamasutra.com/features/20000228/ulrich_pfv.htm, (2000-02-28)[2012-06-06].
- [6] De Boer W H. Fast terrain rendering using geometrical mipmapping [R/OL]. <http://www.flipcode.com/articles/geomipmaps.pdf>, (2000-10-01) [2012-06-06].
- [7] 罗景馨,唐璠. 基于改进四叉树分割和结点存储的 LOD 算法[J]. 计算机工程, 2009, 35(20): 202-204. Luo Jingxin, Tang Jin. LOD algorithm based on improved quadtree-split and node storage[J]. Computer Engineering, 2009, 35(20): 202-204.
- [8] 于卓,梁晓辉,马上,等. 一种支持大规模多种精度地形的实时绘制算法[J]. 计算机研究与发展, 2010, 47(6): 994-995. Yu Zhuo, Liang Xiaohui, Ma Shang, et al. A real-time rendering method for large terrain including details in different resolutions[J]. Journal of Computer Research and Development, 2010, 47(6): 994-995.
- [9] 王臻,胡敏,李响. 基于四叉树的动态多分辨率 LOD 地形快速简化[J]. 计算机应用, 2007, 7: 1643-1644. Wang Zhen, Hu Min, Li Xiang. Quick simplification for LOD terrain with dynamic multi-resolution based on quad tree[J]. Computer Applications, 2007, 7: 1643-1644.
- [10] 闵卫东,唐泽圣. 二维任意域内点集的 Delaunay 三角划分生成算法[J]. 计算机学报, 1995, 18(5): 368-369. Min Weidong, Tang Zesheng. Algorithms for generating the delaunay triangulation of a point set within an arbitrary 2D domain[J]. Chinese Journal of Computers, 1995, 18(5): 368-369.
- [11] 张真. 海量数据 Delaunay 三角网的并行构建算法[J]. 计算机工程与科学, 2013, 35(3): 5-6. Zhang Zhen. Parallel generation algorithm for Delaunay TIN with massive data[J]. Computer Engineering & Science, 2013, 35(3): 5-6.
- [12] NSF OpenTopography, LiDAR point cloud data[DB/OL]. http://opentopo.sdsc.edu/gridsphere/gridsphere?cid=geonlidarframeportlet&gs_action=lidarDataset&opentopoID=OTLAS.082013.26917.1, (2012-09-01)[2013-11-24].