基于八叉树的虚拟场景管理器的设计与实现①

沈永增，刘东岳，徐 均

(浙江工业大学，杭州 310023)

摘 要：以三维电子地图作为应用背景，重点介绍如何利用改进的八叉树来设计并实现一个三维场景管理器，用来对整个场景进行有效地组织和管理。在此基础上通过 Android 系统平台的原生 OpenGL ES 图形库将场景进行重现。针对场景划分过程，提出了基于模型特征的场景划分方式。经过实验证实了该方法可以在一定程度上避免不必要的内存开销，有利于应用在资源相对有限的嵌入式设备上。

关键词：三维电子地图；Android；OpenGL ES；八叉树；空间划分

Design and Implementation of an Octree-based Virtual Scene Manager

SHEN Yong-Zeng, LIU Dong-Yue, XU Jun

(Zhejiang University of Technology, Hangzhou 310023, China)

Abstract: In this paper, as a background of three-dimensional electronic map, we present a program to design and implement a three-dimensional scene manager, which can organize and manage the entire scene effectively, via optimized Octree. And then, on this basis, we reproduce the scene on the Android platform through native OpenGL ES graphics library. A new space division method, which is based on models' spatial features, will be presented later. After experiments, consequences confirm that this method can avoid unnecessary memory overhead to some extent, so it has the possibility of more conducive to application in embedded devices with the relatively limited resources.

Keywords: three-dimensional electronic map; Android; OpenGL ES; Octree; space division

近些年，嵌入式领域的硬件发展十分迅速，硬件处理三维网格模型并加以显示的能力也在逐年提高。软件技术、硬件性能的突飞猛进带动了 GIS 在移动手持设备市场的拓展。面向三维空间而不再拘泥于二维平面的电子地图，是导航领域发展的方向之一，目前也已经成为研究的热点，并取得了卓有成效的进展。在三维电子地图的实现当中，有许多功能要依靠大规模的室外场景的动态渲染，例如在虚拟场景中实现漫游功能。这方面存在着许多技术上的难点，其中一个就是大规模场景的管理，它关系到后期的场景调度问题。因此，程序中需要有一个模块能够对三维模型进行高效地存储与管理[1]。

到目前，国内、外研究学者已经提出了很多的数据结构和方法来管理三维的几何模型，例如较为常用的 BSP 树[2]、KD-tree[3]和八叉树[4]，还有多种构造方式不同的包围盒。其中八叉树是比较具有代表性的且相对简单易于实现的一种数据结构。本研究针对大规模网格模型这一具体应用环境，着重讨论基于标准八叉树的场景划分的改进方法。

1 八叉树与松散八叉树

对于一个场景而言，都有一个相对于自身的八叉树。一棵八叉树的每一个节点对应三维空间中的一部分，全部则对应根节点。它是将三维数据集划分到不同的子空间（8 个卦限）中的一种典型的数据结构。每一个节点都有自己的包围盒，这些包围盒或是沿坐

① 基金项目:浙江省科技厅面上项目(2007C30008)
收稿时间:2011-07-04;收到修改稿时间:2011-07-30

标轴的 AABB（axis-aligned bounding box），或是方向包围盒 OBB（oriented bounding box）等。本文讨论的是基于 AABB 的空间划分。空间划分的方式也较为简单：对于一个场景，首先建立一个最大的包围盒，并设置为根节点。可直接将此包围盒定义为能包含整个场景的最小正六面体。每一个节点都会被细分成八个子节点（对于标准八叉树来说，这八个子节点各自对应的包围盒都是相同的），逐层进行下去，可以采用广度优先的遍历顺序进行节点操作。节点划分有两个限制条件：第一，当子空间的包围盒体积小于预先定义的阈值时终止对该节点细分；第二，当子空间中的图元（点、面等）的个数小于预先定义的阈值时亦终止对该节点的细分[1]。

首先可以肯定的是，八叉树的使用可以极大地加速一些空间操作，例如可视范围的剪裁等。同时，八叉树还有一个优势，就是它的深度往往令人满意。树的深度会一定程度上影响程序运行效率。

当然标准八叉树由于自身划分的呆板性，使得它很容易遇到类似于小几何体高层聚集的问题[5]。而这会降低整个空间操作的效率。于是 Thatcher Ulrich 提出了松散八叉树来解决这种问题。但当对于一个场景中很少有小几何体，例如本应用中的三维地图，其场景中大多是一些大型的建筑物模型，因此松散八叉树并不能带来性能的明显提升。本文提出了基于空间模型的场景划分管理方式。

2　空间场景的划分算法

场景中可能包含多个相互独立的几何体（彼此不连通，也不存在空间的交集），最简单的情况当然是只包含一个独立的几何体。现在考虑一般情况。假设对于一个场景，其中包含了一个拥有 n 个几何体 $g_i(0 \le i < n)$ 的集合 $G=\{g_0,g_1,...,g_i,...,g_{n-1}\}$，其中 $n \ge 1$，每一个几何体都会有其自身在三个轴方向上的最大坐标值、最小坐标值分量，记为 Max^i_{axis} 和 Min^i_{axis}（分别表示第 i 个物体在 $axis$ 轴方向的最大值和最小值，其它记法类似）。这些是属于局部范围内的，故可称之为第 i 个物体在 $axis$ 轴方向的极大值和极小值。通过读取模型的几何数据，可以得到整个场景中对于集合 G 的三个轴方向上的最大坐标值和最小指标值，记为（Max^i_{axis}、Min^i_{axis}）。通过比较得到三个轴向上切分平面的方程，可知这里运用的切分平面是轴对齐的。划分流程图如图 1 所示。在整个流程中，确定切分平面方程的部分位于划分节点包围盒子步骤，是算法的重点。

假定场景中有 n 个相对独立的几何体。场景节点的三个切分平面方程分别为 $x=a$、$y=b$、$z=c$，并且初始化时都经过包围盒中心原点。

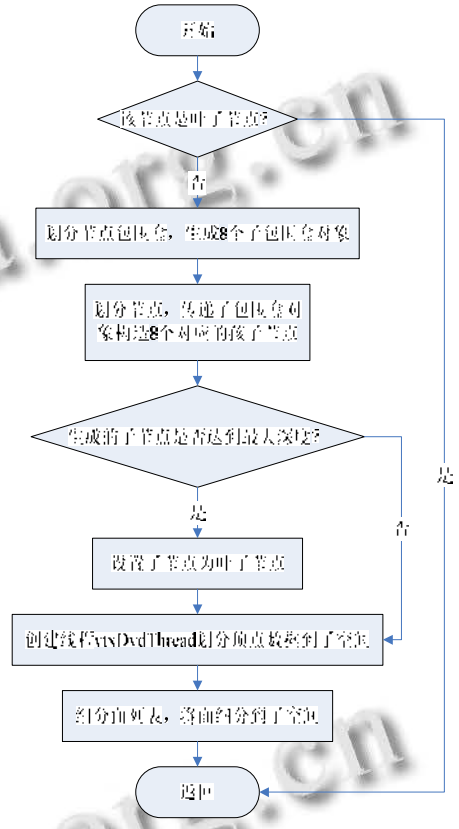

图 1　节点划分流程图

在下面将要描述的节点划分算法中，i 代替 (a,b,c)，j 为几何体的序号，且，代替 $j \in [0,n), j \in N$，$axis$、(x,y,z)、分别表示相对于当前空间节点中模型坐标数据在三个轴向上的最小值和最大值，k 为调节系数（缺省情况可以设置为 0.15），l 为调节的补充值。补充值在原则上以为了切分平面上尽量没有顶点为设置依据。算法描述如下：

① 首先判断节点包围盒自身的体积与最小体积阈值。如果小于最小体积 MINVOLUME，直接跳转到 6。否则继续；

② 初始化 a、b、c；

③ 用 a、b、c 分别和 $[Min_x,Max_x]$、$[Min_y,Max_y]$、$[Min_z,Max_z]$ 进行比较，如果发生以下情况：

(i).如果 $i \notin [Min_{axis}, Max_{axis}]$，跳转 5；

(ii). 如果 $i < Max_{axis}$ 并且 $i > Min_{axis}$，并且 $i < Min_{axis} + (Max_{axis} - Min_{axis}) \times k$，则 $i = Min_{axis} - l$；

(iii). 如果 $i < Max_{axis}$ 并且 $i > Min_{axis}$，并且 $i > Min_{axis} + (Max_{axis} - Min_{axis}) \times (1-k)$，则 $i = Max_{axis} + l$；

(iv). 如果 $i < Max_{axis}$ 并且 $i > Min_{axis}$，但 $i \in [Min + (Max - Min) \times k, Min + (Max - Min) \times (1-k)]$，再如果模型数量为 1，转步骤 5，否则转步骤 4。

④ 比较每一对极值 Min_{axis}^{j} 和 Max_{axis}^{j}，找到可能存在的“空隙”，将 i 调节到这个空隙范围中；

⑤ 按照确定的 a、b、c 创建子空间的包围盒。

⑥ 结束返回。

经过上述划分过程，位于同一级的每个节点的包围盒大小会根据场景模型数据决定，体积并不一定完全相同。

3 数据获取

文献[6]提出使用 ASC 格式的文件作为场景重现的数据源。如果需要添加光照效果，需要法线数据。本应用以 3DS Max 直接导出的 ASE 文件作为数据源。该文件内容不但较为全面，且易于读取和修改。

关于 ASE 文件的内容，若以场景的重现为目的，程序更加关心 GEOMOBJECT 数据块中的 MESH 部分，其中包含了与场景模型相关的几何数据。程序定义了 CConstruction 类用来接收这些数据。场景中有几个独立模型，就有几个对应的数据文件。定义一个场景管理器对象，构造函数中传递一个八叉树对象。在构造这个八叉树对象时，传递这些数据到它的根节点中，然后运用在上一节描述的划分方式，将数据划分到不同的子空间中。

4 场景管理器的设计

整个程序包含了如下几个关键的类：场景管理器类 COctreeSceneManager、八叉树类 COctree、八叉树节点类 COctreeNode 等。

4.1 场景管理器类

该类作为场景中对应八叉树的管理者，包含了一个八叉树对象 mOctree。

类中提供了若干个获取数据成员的方法，如图 2 所示。preProcess()是在渲染场景之前，需要调用的对场景树进行必要操作的方法，具体操作包括场景树的生长，模型细节简化等。

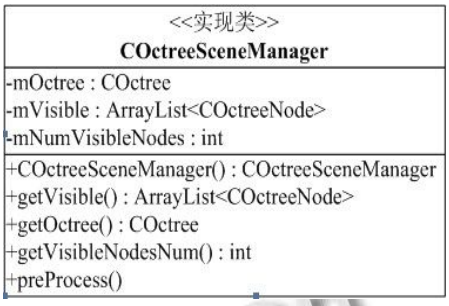

图 2 COctreeSceneManager 类

4.2 八叉树类

该类的功能是对场景中的数据进行有效的组织。结构如图 3：

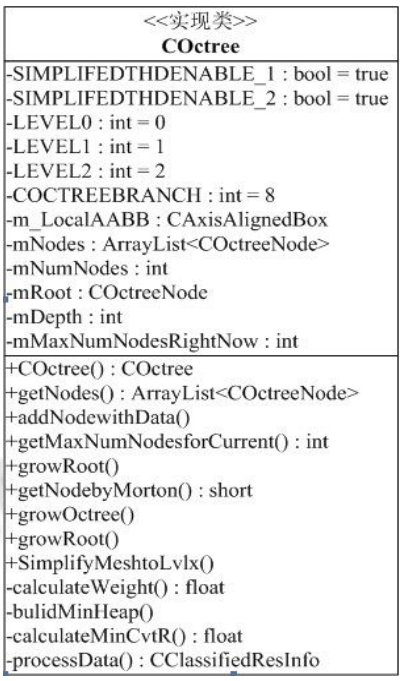

图 3 COctree 类

属性列表中需要一个表示自身的 mRoot 节点对象和它的包围盒对象 m_LocalAABB。它还必须包含一个整体叶子节点集合 mNodes。在设计这个类的时候，考虑到场景需要动态划分，所以提供一个划分方法 growOctree，具体划分操作在其中进行。

4.2 八叉树节点类

八叉树中的节点对象是空间操作的基本对象，因此地位相当重要，数据成员也较多，结构如图 4 所示。

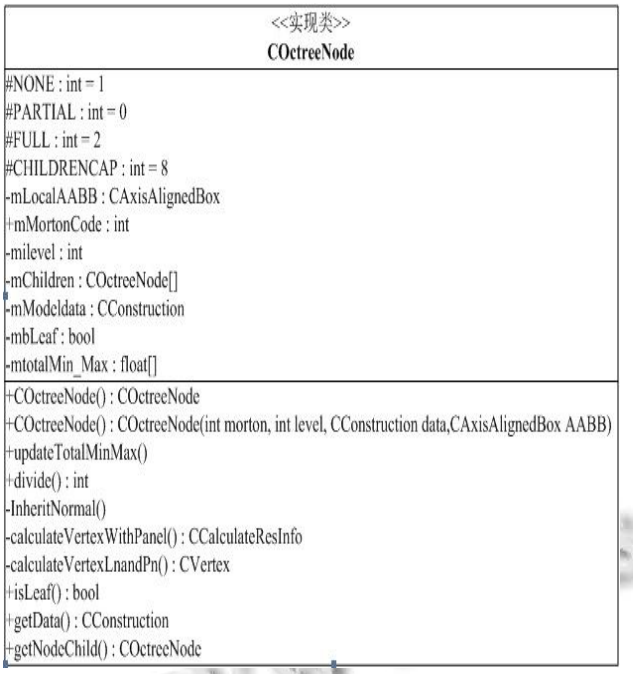

图 4　COctreeNode 类

类中包含如下主要数据成员：节点对应场景的包围盒 mLocalAABB、节点对应的 Morton 码[7]、节点所处的深度 milevel、所包含的子节点集合 mChildren 和场景中的数据 mModeldata 等。除了构造方法以外，最重要的还有划分方法 divide()，负责节点的划分工作：节点对应孩子节点的构造、包围盒的划分和模型数据的细分。

5　应用及结论

本应用以杭州市平面地图的路网数据作为依据，重点以上塘高架复杂地段为例，进行场景的创建。首先通过道路关键点创建样条线。它可以确定整条道路的走向。在得到代表上塘路和德胜路的两条线之后，确定相交处复杂地段的位置，然后根据 Google Map 卫星图像以及 E 都市的模型，创建场景如图 5。

场景采用 Android 系统原生的 OpenGL ES API 重现[8,9]，图 6 为程序初步运行结果。

为了凸显八叉树改进之后的效果，在创建模型时有意将整个模型的 Z 轴方向坐标调大，并使平面方程位于 Z 轴向的可调节范围内。将八叉树标准划分方式和本文提出的改进划分方式作比较，程序性能结果如表 1 所示。观察表中的 Data object，从 Count 数值可以看出，后者产生的数据对象个数有所减少。

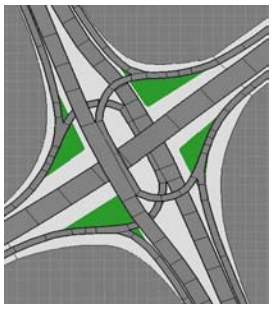

图 5　建模效果图

图 6　程序运行结果

表 1　数据对比

Type	Count	Total Size	Smallest	Largest	Average
Data object (standard)	53285	1.386MB	16B	384B	27B
Data object (optimized)	46808	1.236MB	16B	384B	27B

图 7 和图 8 是实验对应生成的柱状图。如果模型复杂度以及坐标范围扩大，这种优势会更明显。

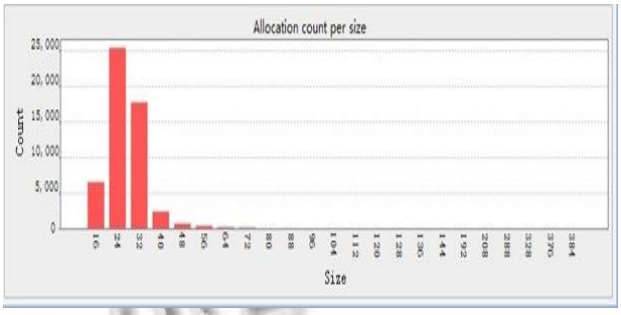

图 7　标准划分方式

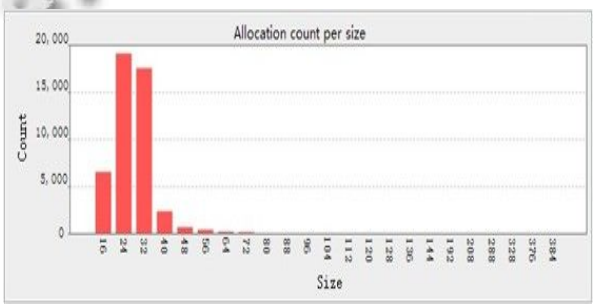

图 8　改进划分方式

参考文献

1 赵欣,桂岚,易平.八叉树在公路三维模型中的应用.长沙交通学院学报,2006,22(1):58–62.

2 陶志良,成迟薏,潘志庚,石教英.多分辨率 BSP 树的生成及

（下转第 45 页）

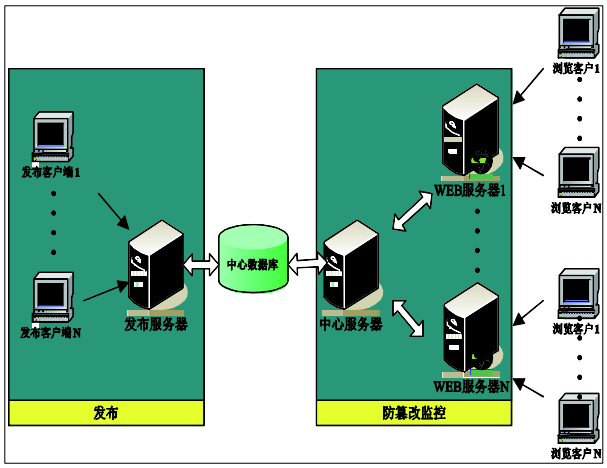

图 2 系统部署图

防篡改监控：这部分属于实时阻断，部署在中心服务器，一旦部署网页文件的文件目录或文件有非法进程进行更改，系统立即进行阻止。

发布: 这部分属于事后恢复，部署在发布服务器，当用户请求访问网页文件，系统将网页文件的水印与中心数据库文件的水印进行比较，如发现不能匹配，即中止响应，并恢复被篡改的文件。

通过对系统的层次、模块和部署三个维度的分析和设计形成了本系统的完整实现思路，本系统作为一种在线部署的产品，其目标不仅仅在于能够精确识别各种网页篡改方式，而且在不影响正常业务流量的前提下，使得公众无法看到被篡改页面，实施实时阻断，其运行性能和检测实时性都达到较高的水准，为用户提供较全面的安全检测与防御。

4 结语

本系统是在响应国家“信息安全”产品产业化专项基金的要求下，旨在研发出一种能自动采取行动阻止非法篡改和入侵。通过部署该防护系统，同时与防火墙、入侵检测等方式，形成深度防御体系。与传统的网页防篡改产品相比，本系统可以对网页的非法篡改做出更全面、更敏捷快速的反应，同时对系统资源占用更少的特点，最大限度地保护企业和组织的网络安全。目前本系统已经通过国家发改委验收，并成功在 PowerSEC 高性能监控一体化网络安全平台上，实践证明该系统具有极大的研究价值、开发价值和市场前景。然而，网站防护现在还处于不断完善的过程，“三分技术、七分管理”，在管理上还需要加强对技术设备的日常管理以及制定出网站安全的相关安全管理制度，并在日常工作中加以培训、落实，从思想真正提高重视，这样才能够有效地做好网站安全的防护工作。

参考文献

1 CNCERT/CC.中国互联网网络安全报告(2011 上半年)北京:国家计算机网络应急技术处理协调中心,2011.
2 张岸,张毅东.政府网站安全防护解决方案研究.情报探索,2010,(11):84–86.
3 范建华,宋云波.基于文件过滤驱动和事件触发的网页防篡改机制.重庆工学院学报,2009,23(12):65–70.
4 张迎春.基于三层过滤的分布式网页防篡改系统.济南:山东大学,2007.
5 符广全,王海峰.基于文件过滤驱动的内核病毒防火墙技术.计算机应用与软件,2006,(7):21–23.

（上接第 150 页）

应用.软件学报,2001,12(1):117–125.
3 石波,卢秀山,陈允芳.基于 kd-tree 的建筑物散乱点云平面分割.测绘科学,2008,33(1):135–136.
4 王世东,陈杨,张本福,孙光灵,黄晓梅.八叉树在三维建模中的应用.安徽建筑工业学院学报(自然科学版),2006,14(6):96–98.
5 胡斌,江南,邹志强,邵华,王鹏.大规模室外动态场景调度机制研究.地球信息科学,2007,9(5):8–13.
6 郭景,雷鸣.3DS MAX 模型在 OPENGL 中的读取与重现.计算机应用,2002,(5):46–49.
7 Lewiner T, Mello V, Peixoto A, Pesco S, Lopes H. Fast Generation of Pointerless Octree Duals. Computer Graphics Forum, 2010,29(5):1661–1669.
8 刘琪,迟贤书.OpenGL 与 OpenGL ES 在开发过程中的异同.辽宁工程技术大学学报,2008,27(2):261–262.
9 Astle D, Durnil D. OpenGL ES Game Development. Course Technology PTR, 2004:35–45.