基于八叉树及 KD 树的混合型点云数据存储结构①

廖丽琼，白俊松，罗德安

(北京建筑工程学院 测绘工程系，北京 100044)

摘 要：通过对现有点云数据存储结构进行综合分析及比较，提出了一种基于八叉树及 KD 树的混合型点云数据存储结构模型，文中对该模型的基本原理、实现步骤及快速索引的建立等进行了全面的论述，最后以一组实测数据为例，比较了 KD 树、八叉树和本文提出的混合结构三种不同数据组织方式的检索效率，证明了所提出存储结构的有效性及实用性。

关键词：海量点云；数据结构；三维激光扫描；存储结构；索引

Integrated Point Cloud Storage Structure Based on Octree and KDTree

LIAO Li-Qiong, BAI Jun-Song, LUO De-An

(Dept. of Surveying and Mapping, Beijing Institute of Construction Engineering and Architecture, Beijing 100044, China)

Abstract: Based on analysis and comparison to existing point cloud storage structures, presents an integrated point cloud storage structure base on Octree and KDTree, describes its basic theory, work flow and index building method, at last a group of real data are employed to check and test the efficiency of Octree, KDTree and presented data structure, and the results shows that presented data structure has higher search efficiency and more suitable for point cloud data storage than the others.

Key words: huge quantity of point cloud; data structure; 3D laser scanning; storage structure; index

随着地面激光雷达技术的逐步成熟，以及相关仪器价格的逐步下降，越来越多的领域（如古建筑保护、施工质量检测、考古、文物数字化及建档等）开始引入该仪器，并开展了众多的应用实践，产生了良好的经济及社会效益。地面激光雷达能够以离散三维点全覆盖模式采集考察对象的高精度几何信息，和其他测量设备（如全站仪）比较而言，有其独特的技术优势，主要体现在速度快、整体精度高和全覆盖等方面。借助地面激光雷达，能够快速的获取考察对象相关的海量点云数据，根据这些数据，可以生成和制作各种三维模型、构建三维场景以及进行几何量测获取相关的几何信息等等，这些数据为数字城市、虚拟局部环境的快速建设提供了一种相对廉价而丰富的数据源。

尽管可以获得海量的廉价点云数据，但这并不是我们最终所需要的数据，点云数据必须经过一系列的数据处理（粗差剔除、数据分割、特征提取、建模、纹理映射及场景组建等）才能最终获取满足要求的信息[1]。面对如此海量（Gb 级甚至 Tb 级）的、没有拓扑关系的点云数据，如果没有一个高效的数据存取结构，几乎所有的分割及特征提取算法都将趋于无效，因为在邻域查找中将耗费大量的检索时间，甚至是根本无法实现。此外，对于有限的内存及运算速度，这样海量的数据有时是无法一次加载及显示的，必须选择和使用合理的数据调度方式（如分级、多分辨率、缓存及预取等）才能完成相应的数据处理及操作。

鉴于此，通过对现有点云数据存储结构及存取方法的综合分析及比较，本文提出了一种基于八叉树及 KD 树的混合型点云数据存储结构模型，文中对模型的基本原理、实现步骤及快速索引的建立等进行了全面的论述，最后以一组实测数据为例，比较了 KD 树、

① 基金项目:国家自然科学基金(40871196,40901192)
收稿时间:2011-06-18;收到修改稿时间:2011-07-24

八叉树和本文提出的混合结构三种不同数据组织方式的检索效率，证明了所提出数据存储结构的有效性及实用性。

1 文献回顾

点云数据处理中最为核心的问题就是建立离散点间的拓扑关系，实现基于邻域关系的快速查找。要实现海量空间数据的快速查找，就必须建立与其相适应的空间索引结构，通过一定的数据组织方式建立无序点云间的拓扑关系，同时存储相关的概要信息，辅助并加速邻域信息的查找。建立空间索引在 GIS（地理信息系统）中已被广泛应用，常见空间索引一般是自顶向下逐级划分空间的各种空间索引结构，比较有代表性的包括 BSP 树、KD 树、KDB 树、R 树、R+树、CELL 树、四叉树和八叉树等索引结构，而在这些结构中，KD 树和八叉树在 3D 点云数据组织中的应用较为广泛[2-6]。

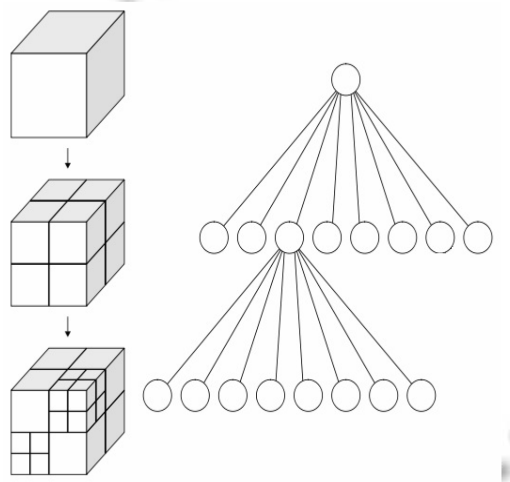

图 1 八叉树划分及结构示意图(Wikipedia)

1.1 八叉树结构

八叉树结构的概念是由 Hunter 博士于 1978 年首次提出的一种数据模型[4]。八叉树结构通过对三维空间的几何实体进行体元剖分，每个体元具有相同的时间和空间的复杂度，通过循环递归的划分方法对大小为 $2^n \times 2^n \times 2^n$ 的三维空间的几何对象进行剖分，从而构成一个具有根节点的方向图。在八叉树结构中如果被划分的体元具有相同的属性，则该体元构成一个叶节点；否则继续对该体元剖分成 8 个子立方体，依次递归剖分，对于 $2^n \times 2^n \times 2^n$ 大小的空间对象，最多剖分 n 次，如图 1 所示。八叉树的生成及使用在很多文献都有详细描述，这里将不再赘述。

1.2 KD 树结构

KD 树是一种面向 k 维空间点数据组织的二叉树结构[7]，其每一个非叶节点可以被一超平面分割成两个子空间，且相应的每一子空间又可以以相同的方式递归地进行分割，所有子空间被分为左右两部分或者被分为上下两部分。KD 树的分割是沿坐标轴进行的，所有的超平面都垂直于相应的坐标轴。例如，沿 x 轴分割，只需要给定某 x 值即可确定该超平面，因为超平面的法矢方向就是 x 轴向，该超平面将原节点空间分割成两个新的子空间，所有左边子空间中的点，其 x 值都小于右边子空间中的任意点的 x 值，如图 2 所示。KD 树是一种较为有效的 k 维空间点数据组织结构，在涉及高维空间查找领域（如 k 邻域查找）具有自身独特优势。

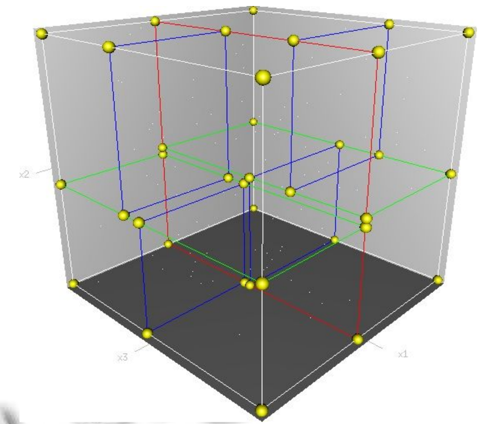

图 2 三维空间 KD 树分割(Wikipedia)

通过对八叉树及 KD 树进行分析和实际的对比后发现，八叉树算法的算法实现简单、自动化程度高，但是针对海量的点云数据，制约其效率的主要因素在于其最小分割粒度（叶节点）的确定，粒度过大，后续的定位及查询效率降低，反之，粒度过小，八叉树的深度增加，同样也会导致效率下降，此外，对于建筑物（或古钟文物等）等薄壳类扫描对象的点云数据，还可能存在有相当数量的叶节点将不存储任何点云数据，而形成所谓的“空白节点”[8]；KD 树在邻域查找效率方面具有较大的优势，但在海量数据情形下，其邻域关系的重建将耗费大量的计算资源，整体效率也不是很高；上述两算法单独应用都存在一定的缺陷，而

混合型数据存储结构也许能够充分利用多种数据存储方式的优点，通过合理的调度，理论上能够使数据组织的整体结构趋于合理，数据存取效率达到最优。

2 混合存储结构原理

通过对目前几种适合于点云数据存储的数据结构进行综合分析，提出了一种融合八叉树和 KD 树的、适合于大规模点云数据存取和快速检索的多级分层嵌套式点云数据存储结构。

2.1 算法原理

算法的基本思想及步骤如下：

① 首先获取存储对象几何边界（最小外包围盒）并设定八叉树的分割深度，据此进一步确定八叉树的分割粒度大小（即叶节点的大小，对应确定叶节点几何尺寸）；

② 根据上面获得的分割参数建立相应的八叉树，存储相应的节点信息；

③ 对于八叉树的叶节点按照 KD 树的数据组织方式逐一进行数据组织，分别存储索引信息和节点坐标信息，与此同时还需将每个 KD 树的首记录（指针地址或是记录 ID）存放到对应的八叉树叶节点中，以保证这种嵌套式结构的唯一性。

④ 数据检索时，首先根据三维坐标判断其所属八叉树的叶子节点，然后找到该叶节点对应的 KD 树数据集，对其进行二次检索，根据检索条件，找到满足要求的数据（或数据集）。值得注意的是，在进行邻域搜索时，如果当前坐标点所在的八叉树叶节点无法找到满足要求的数据集，则必须扩展搜索范围，将与该节点关联的邻近节点并入搜索范围，再次进行搜索，直到找到满足要求的结果或是达到算法终止条件为止。

提出的这种嵌套式复合数据存储结构，克服了单一算法所带来的各种弊端，平衡了数据的分布，提高了数据存取及检索的效率，尤其是针对海量数据的存取与检索，性能提高较为显著。

2.2 算法实例

本文算法主要是服务于海量的数据点云的组织及存储，数据量一般都在百万级以上，限于篇幅，同时为了说明算法的操作流程，这里仅以有限的几十个点来作为示范数据。假定某组 3 维数据点集 pSet={(2,3,1), (4,7,7), (5,4,6) (7,2,4) (9,6,3) (8,1,6), (-2,3,1), (-4,7,7), (-5,4,6) (-7,2,4) (-9,6,3) (-8,1,6), (2,-3,1), (4,-7,7), (5,-4,6) (7,-2,4) (9,-6,3) (8,-1,6), (-2,-3,1), (-4,-7,7), (-5,-4,6) (-7,-2,4) (-9,-6,3) (-8,-1,6), (2,3,-1), (4,7,-7), (5,4,-6) (7,2,-4) (9,6,-3) (8,1,-6), (-2,3,-1), (-4,7,-7), (-5,4,-6) (-7,2,-4) (-9,6,-3) (-8,1,-6), (2,-3,-1), (4,-7,-7), (5,-4,-6) (7,-2,-4) (9,-6,-3) (8,-1,-6), (-2,-3,-1), (-4,-7,-7), (-5,-4,-6) (-7,-2,-4) (-9,-6,-3) (-8,-1,-6)} ，共计 48 个点。这里将该组数据进行 1 次八叉树分割（即深度值设为 1），很明显，分割后的 8 个子节点每个将包含 6 个对应的 3 维点，在此基础上进行 KD 数据分割。如果要逐一陈述每个八叉树节点数据分割后的情况，将会占用大量篇幅，这里仅以 top-left-back 节点的数据集为例进行 KD 分割，该数据集为 ptlbSet={(2,3,1), (4,7,7), (5,4,6) (7,2,4) (9,6,3) (8,1,6)}，按 x 轴、y 轴和 z 轴顺序剖分，剖分结果如图 3 所示。

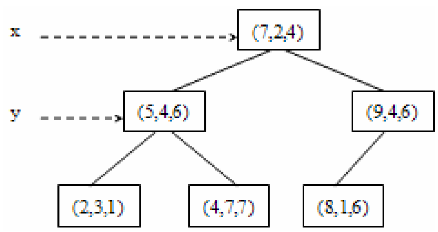

图 3 某八叉数据节点的 KD 划分

3 算法对比分析

为验证本算法的有效性，这里选取了一组共 3 个测试点云数据：测试数据 1 为某建筑物局部点云，数据量达 936770 个点，用莱卡 Scanstation 2 扫描获得；测试数据 2 为某完整的古钟（腔径在 50cm 左右）点云数据，数据总量达到 3176096 个，该数据为通过海克斯康关节臂扫描获取的超精细点云数据，分辨率在 1mm 左右，其点云模型和真实纹理模型如图 4 所示；测试数据 3 为某完整的建筑物点云数据，数据量达到 6423463，也是用莱卡 Scanstation 2 扫描获取的点云。

图 4 古钟模型（左为点云，右为真实纹理模型）

对上述的任意一测试数据，分别采用了 3 种不同的数据存储结构来组织数据，并针对指定的同一个点进行最近 K 邻域点（K=2）查找。对于本文的算法，八叉树的深度统一都设置成 5。算法运行的环境为 Dell Vostro 1710 笔记本，CPU 主频为 2.1GHz，内存为 2G，运行结果如表 1 所示

表 1　结构建立及邻域查找耗时

点数	KD 树	八叉树	本文算法
936770	0.78s	6.207s	0.241s
3176096	2.321s	25.452s	0.798s
6423463	5.522s	68.357s	1.452s

从表 1 可以看出，KD 树的效率明显高于八叉树，这是由于随着数据量的增加，八叉树的深度也迅速增加，从而导致其检索速度变慢；KD 树在数据量适中或较小的情形下具有极快的搜索速度，但随着数据量增加，同样会增加其检索的复杂度，导致速度很快下降；本文算法明显优于前两种算法，搜索速度大幅度提高，主要因为本文算法充分利用了两者的优势：先利用八叉树实现概略检索，再利用 KD 树实现检索精确定位，这样有效平衡了各环节的数据处理工作量，提高了整体效率。应当指出，这里将 3 组数据的八叉树深度都设置成 5 显得过于武断，实际运用应该做适当的测试，针对不同的数据类型及其数据量大小设置更为合适的八叉树深度值。

4　结语

为满足海量点云数据高效管理及组织的需要，本文提出了一种基于八叉树及 KD 树的混合型点云数据存储结构，该数据结构很好地利用了八叉树和 KD 树的技术优势，平衡了各环节的工作量，效率明显优于单纯的八叉树及 KD 树索引结构，实验也证明了该数据结构能够实现海量数据环境下的快速邻域查找。

存在的问题及进一步工作：如前所述，针对不同的数据量及数据类型，八叉树的深度最优值应该是不同的，但目前只能通过预测试来确定，如何能够实现自动化及智能化设置八叉树的深度值还需进一步加以研究；此外，点云数据的处理效率除了直接和使用的算法有关外，还和数据的读取方式、计算机的物理限制（如内存限制等）相关联，当数据量达到计算机的物理极限无法一次性将数据读入内存时，则必须考虑使用缓存、预取、分级或多分辨率（LOD）等渐进传输的方式来实现数据处理。

参考文献

1 廖丽琼,罗德安.地面激光雷达的数据处理及其精度分析.四川测绘,2004,27(4):153–155.

2 董道国,薛向阳,罗航哉.多维数据索引结构回顾.计算机科学,2002,29(3):1–6.

3 宋扬,潘懋,朱雷.三维 GIS 中 R 树索引研究.计算机工程与应用,2004,(14):9–10.

4 惠文华,郭新成.三维 GIS 中的八叉树空间索引研究.测绘通报,2003,1:25–27.

5 郑坤,朱良 S 峰,吴信才,等.3DGIS 空间索引技术研究.地理与地理信息科学,2006,22(4):35–39.

6 吴涵,杨克俭.基于 kd 树的多维索引在数据库中的运用.计算机应用,2007,26(9):37–39.

7 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.

8 路明月,何永健.三维海量点云数据的组织与索引方法.地球信息科学,2008,10(2):190–194.

（上接第 219 页）

幕特点，采用 Qt 图形用户界面开发工具，设计电纸书图形用户界面，同时以文本文档显示为例，实现不同编码方式的文本文档的解码及分页显示。系统在满足用户基本阅读需求的前提下，最大程度地利用了电子纸显示屏的显示界面，节省了按键资源。

参考文献

1 张勇.电纸书-引领数字阅读潮流.科技信息,2010,(28):379.

2 曾祥健,姚倩茹.浅谈国内电子书市场现状.财经视点，2010,(7).

3 郭晓琳.电子阅读器驱动程序设计与实现.成都:电子科技大学,2006.

4 吴蔚.任重道远电子书.中国计算机报,2007,12.

5 王丽洁.嵌入式 Linux 的图形界面技术研究与实现.长沙:国防科技大学,2006.