

# 车载 LiDAR 点云混合索引新方法

张蕊<sup>1,2</sup> 李广云<sup>1</sup> 王力<sup>1</sup> 李明磊<sup>1</sup> 周阳林<sup>1</sup>

1 信息工程大学导航与空天目标工程学院,河南 郑州, 450001  
2 华北水利水电大学,河南 郑州, 450045

**摘 要:**以车载 LiDAR 点云数据为研究对象,为提高点云数据的组织与管理效率,提出了一种全局 KD 树与局部八叉树相结合的混合空间索引结构——KD-OcTree。全局 KD 树通过分辨率、分割平面的确定,重构点云之间的邻域关系,确保索引结构的整体平衡;在其叶子节点再构造二级索引结构——局部八叉树,避免了单一八叉树结构点云分布不均衡、树结构深度过大、出现大量无点空间等现象。以 3 个真实场景数据为测试数据进行试验和对比分析,结果表明,KD-OcTree 混合索引不仅能够提高索引构建、邻域搜索的速度,还对分类可靠性产生一定影响。

**关键词:**车载 LiDAR;点云;全局 KD 树;局部八叉树;混合空间索引

**中图分类号:**P232;P208      **文献标志码:**A

随着点云数据获取手段的提高,传统的数据组织与管理方法已成为海量点云后期数据处理的一大瓶颈,因此对海量点云数据进行高效的组织与管理变得日益迫切。针对车载点云数据的海量、空间离散等特性,为实现其高效管理,通常需要借助外存辅助数据结构加快空间查询的速度,如四叉树索引<sup>[1]</sup>、八叉树索引<sup>[2]</sup>、R 树索引及其变种<sup>[3-7]</sup>、双层四叉树索引<sup>[8]</sup>、八叉树与 KD 树混合索引<sup>[9]</sup>、八叉树 LOD 索引<sup>[10]</sup>、四叉树结合 R 树混合索引<sup>[11-12]</sup>、四叉树结合八叉树混合索引<sup>[13]</sup>、八叉树与 R<sup>+</sup> 树混合索引<sup>[14]</sup>、四叉树结合 KD 树混合索引<sup>[14-15]</sup>等。

现有的点云数据组织与管理方式对于海量车载 LiDAR 点云而言,存在一定的局限性,主要体现在以下几个方面:①单一索引(如四叉树、八叉树、R 树及其变种、KD 树等)对于少量、分布均匀的点云来说,其组织与管理效果较好,但不适用于车载 LiDAR 扫描系统获取的大尺寸场景点云(通常表现为海量、分布不均匀);②与应用领域相结合的混合索引鲁棒性差,当应用场景发生变化时,点云数据的检索效率较低;③点云数据的组织与管理方式和某些点云的后期处理算法不兼容,如八叉树与 KD 树混合索引适用于单点查询或  $k$

邻域搜索,对于以块为单位的点云搜索算法则无能为力,并且直接影响点云数据的后期处理效果。

针对以上问题,为了提高点云数据的检索效率,对叶子节点的点云进行二次重组变得尤为重要,重组的方式就是对原有索引结构的叶子节点再次构造索引,形成一种新的两级混合索引。构造混合索引的主要技术包括:①单一索引结构的选择要取长补短,充分发挥单一索引结构的优势,弥补劣势;②设计全局索引和局部索引的构建算法,包括数据存储结构的设计、分割维度的选择、分割面的确定等;③确定全局索引结构分割终止的条件等。

本文以车载 LiDAR 点云数据为研究对象,提出了一种全局 KD 树和局部八叉树相结合(KD-OcTree)的点云数据组织与管理方法。全局 KD 树保证了整体索引结构的平衡,使树结构的深度不会出现倾斜情况,因而树的层数也偏小;局部索引结构采用八叉树对点云进行管理,保证了后期数据处理时能以体素为单位对海量点云进行快速检索。新的 KD-OcTree 混合索引结构使得树结构均衡,能够对海量点云数据进行高效组织,并以块为单位对点云进行快速检索。

## 1 两级混合索引结构的确定

若用四叉树与其他单一索引结构进行组合构建混合索引,首先需求出每个节点的最小外包矩形(minimum bounding rectangle, MBR),然后利用该 MBR 反算到三维空间中,提取对应格网中的点云数据,效率低下。若用高度平衡的 R 树及其变种对分布不均匀的车载点云进行组织与管理,则会有大量“无点空间”的存在,浪费大量存储空间,且检索的效率也会大大降低。KD 树是一棵平衡树,不要求所有的叶子节点都在同一层,同时八叉树为典型的三维空间数据结构,经过迭代划分,点云数据可均匀地分布在各个子节点中,特别适合“块”处理,因此选取 KD 树和八叉树来构建两级混合索引结构。

### 1.1 全局 KD 树索引

对点云数据集构建 KD 树表示对该三维数据集构成的三维空间的一次划分,每个节点对应一个三维超矩形区域。该过程涉及到两项技术:①分辨率的定义,即每层分割维度的确定;②分割超平面的确定,要确保在当前维度上。划分得到的两个子集合中,点的个数尽量相等,使 KD 树为一棵平衡二叉树或接近于平衡二叉树。

对于分割维度的选择,可选择最大方差法,也可选择  $k$  维轮流作为分割维度。最大方差法需先求出各个维度上的方差,再比较其大小,最后以最大方差所在的维度作为分割维度;该方法每次都需要对各个子空间在  $k$  个维度上分别求方差,计算复杂。相比而言, $k$  维轮流作为分割维度则较为简单,计算量少,分辨率定义为  $i \bmod k$  或  $(i+1) \bmod k$ ,其中  $i$  表示当前分割维度( $i=0, 1, 2 \cdots k-1$ )。

对于分割面的确定,为使点云均匀分布在各个子节点中,以分割维度的中值作为分割点坐标,建立垂直于当前坐标轴的平面。

### 1.2 局部八叉树索引

由车载 LiDAR 扫描系统获取的点云从空间分布上来说具有不均匀性,存在某些区域点云分布密集而某些区域点云稀疏甚至没有点的情况。此时八叉树单一索引结构存在的弊端就会凸现出来,即会造成叶子节点中的点云数据量不均衡,整棵八叉树深度过大,出现大量“无点空间”;在叶子节点,对点进行检索的效率也会大大降低。

为此,对经典八叉树进行改进,出现了自适应八叉树<sup>[16]</sup>,依据点的分布稀疏程度进行分割。该

方式避免了“无点空间”,节约了存储空间,但无法避免八叉树深度过大、树结构失衡的现象;此时在最深层叶子节点中进行点云搜索时,耗时仍然过多,会降低整个点云数据的检索效率。

由此可见,八叉树单一索引结构很难达到既节省存储空间又提高检索效率的效果。由于点云数据的海量特性与不均匀性,若仅采用 KD 树进行分割,存在以下问题:一是建立点云的邻接关系需要耗费大量的时间及计算资源;二是根据建好的 KD 树对点云进行逐点邻域搜索的速度较慢;三是鲁棒性低,与有些特征提取算法不兼容,如基于体素的向上生长算法<sup>[17]</sup>。为充分利用 KD 树可构造一棵平衡二叉树的优势,在数据全局用 KD 树格网组织数据,局部采用八叉树索引,便于“块”运算等算法的执行。因此,本文提出一种全局 KD 树与局部八叉树相结合的混合索引,命名为 KD-OcTree 索引。

## 2 KD-OcTree 混合索引的构建

### 2.1 KD-OcTree 混合索引的逻辑结构

构造全局 KD 树时,每层节点在当前层分辨率所决定的维度  $D_i$  上进行划分,对于分割面的确定,可以将当前层分割维度  $D_i$  上所有点的中位数作为分割平面。要求中位数,需要对点云数据集在当前维度  $D_i$  上的坐标分量重新排序。为提高 KD 树整体构建的效率,设置 KD 树叶子节点的点数阈值,当节点点数达到该阈值或当前分割层数达到最大分割层数时, KD 树分割终止,并标记当前节点为叶子节点。

为节省存储空间,全局 KD 树索引结构的根节点和中间节点不存储真实点云,真实点云直接存储在叶子节点。该存储方式不仅压缩了数据存储量,而且根据指针可以从根节点快速定位到每个叶子节点,实现快速检索。对 KD 树的每个叶子节点按照八叉树的数据组织方式重新进行组织。当所有的 KD 树叶子节点的局部八叉树都建立完成之后,整个混合索引构建结束。KD-OcTree 混合索引逻辑结构如图 1 所示。

### 2.2 KD-OcTree 混合索引的数据结构

KD-OcTree 混合索引的数据结构如图 2 所示,由 5 个数据对象构成,其中 KDTree 为 KD 树结构体类型, KDNode 为 KD 树节点结构体类型, Stack 为堆栈结构体类型, LinkStack 为链式栈结构体类型, OcTree 为八叉树类型。通常采用定长数组数据结构和递归方法来构建索引结构,而本

文采用了堆栈数据结构和循环方法来建立索引。堆栈数据结构为动态分配存储方法,使用多少分配多少,数据出栈后存储空间立刻释放;且在构建 KD-OcTree 混合索引时,在堆栈中存储的只是指向节点的指针,而非真实点云,从而大大减少了内存占有率;检索时从栈顶元素出发,根据指针指向获取相应节点中存储的点云信息,包括点云个数、点云编号等。

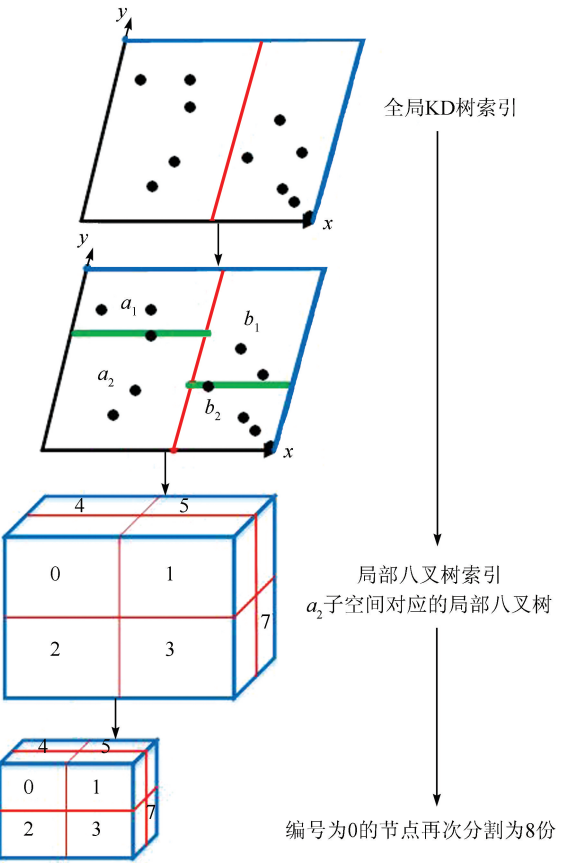


图 1 KD-OcTree 混合索引逻辑结构图  
Fig.1 Logic Structure of KD-OcTree

图 3、图 4 分别给出了构造全局 KD 树与局部八叉树的具体算法描述。

3 试验结果与性能分析

3.1 测试数据

要检测 KD-OcTree 混合索引的效率以及数据处理效果,需与数据后处理算法相结合。将本文所提出的 KD-OcTree 混合索引与基于体素向上生长的地面点滤除算法<sup>[16]</sup>结合起来以检测其有效性,主要基于以下考虑:①高效性,该算法采用分块局部地面点滤除策略,能够快速、高效地处理城区大场景三维点云数据,即使地面有高低起伏,也不影响滤波效果;②有效性,该方法能从最

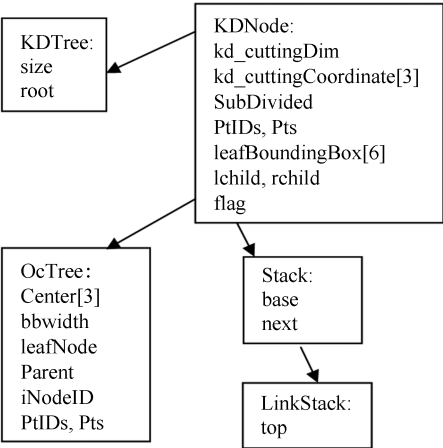


图 2 KD-OcTree 混合索引数据结构图  
Fig.2 Data Structure of KD-OcTree

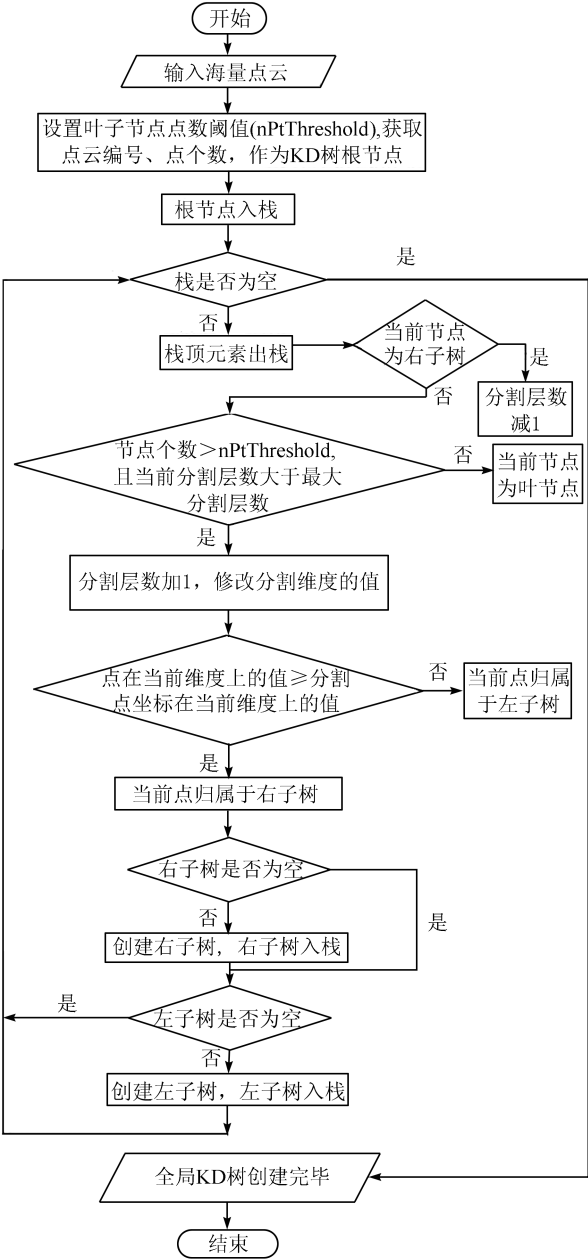


图 3 全局 KD 树构造流程  
Fig.3 Flowchart of Building Global KD-tree

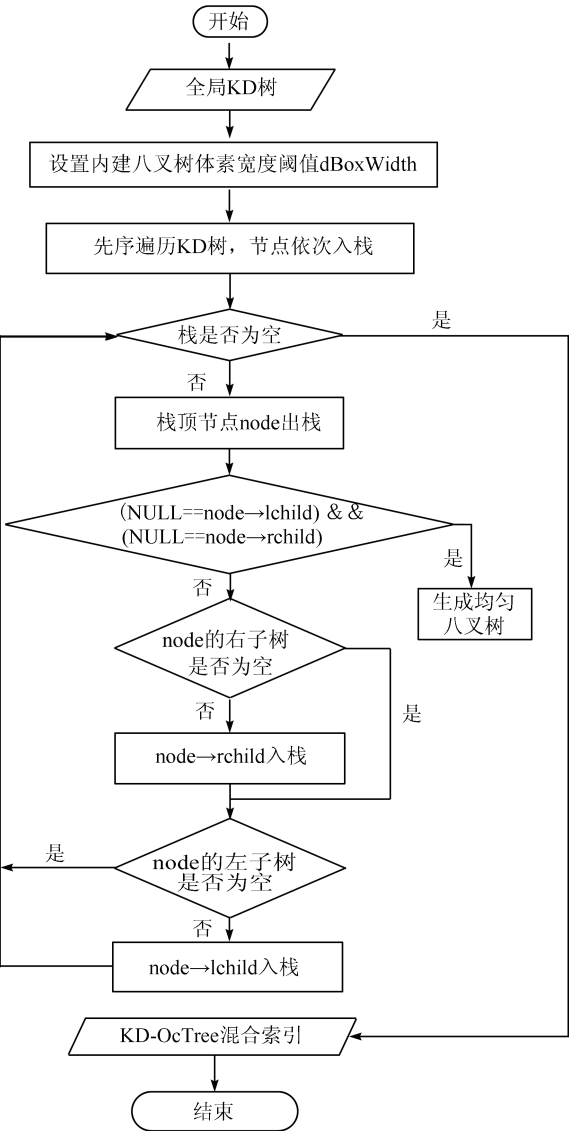


图4 局部八叉树构造流程

Fig.4 Flowchart of Building Local Octree

低点保留非地面目标的有效性。

本试验所获取的车载激光点云来源于西安市环城南路,主要包括西段和东段两部分,全长约4.8 km。从中截取3段作为测试数据:场景1全长49.37 m,点个数823 855;场景2全长80.24 m,点个数1 563 493;场景3全长143.05 m,点个数3 352 290。数据集中包含高密度地面点以及植被、路灯杆等。测试环境为Intel Core i5 2.6 GHz,RAM 8.0 GB,KINGSTON ATA 890 GB。

为验证本文提出的KD-OcTree两级混合索引的有效性,分别从4个方面将KD-OcTree与KD树、八叉树进行对比分析。

3.2 索引构造速度测试

分别以场景1、场景2和场景3为测试数据,将KD-OcTree混合索引、单一KD树、单一八叉树的建树时间进行对比,如表1所示。设全局

KD树叶子的节点的点数阈值为50 000个。可以看出,KD-OcTree混合索引的建树速度最快,八叉树次之,KD树最慢。由于要将索引结构与地面点滤除算法相结合,而测试数据中包含大量地面点,且点云分布均匀,在此情况下3种索引的建树时间已区别开来;若点云分布不均匀,八叉树出现倾斜,KD树深度过大,此时KD-OcTree混合索引的优势会更加突出。

表1 3种索引建树时间对比

Tab.1 Time of Building Three Indexes

场景	点个数	3种索引结构建树时间/s		
		KD树	八叉树	KD-OcTree
场景1	823 855	15.174	11.695	9.888
场景2	1 563 493	31.408	18.858	19.542
场景3	3 352 290	86.833	60.411	56.099

3.3 邻域搜索速度测试

将KD-OcTree混合索引和单一八叉树索引结构与基于体素的向上生长算法相结合,用于地面点滤除,测试两种索引结构的邻域搜索速度。由于KD树用于单点邻域搜索,与基于分块策略的体素生长算法无法配合使用,故不再测试KD树的邻域搜索速度,测试结果如表2所示。可以看出,基于KD-OcTree混合索引的地面点搜索速度比单一八叉树的搜索速度提高了一个数量级,且点云个数越多,利用KD-OcTree混合索引对数据进行组织与管理的效率越明显。

表2 地面点搜索时间对比

Tab.2 Time of Searching Ground Points

场景	点个数	2种索引结构搜索时间/s	
		八叉树	KD-OcTree
场景1	823 855	8.203	0.949
场景2	1 563 493	27.431	2.928
场景3	3 352 290	92.258	5.229

3.4 索引结构对三维目标感知效果的影响

经过大量试验发现,不同索引结构的选择还会对三维目标的感知效果产生明显的影响。表3列出了基于八叉树单一索引与KD-OcTree混合索引滤除地面点的个数对比情况。

表3 滤除地面点个数对比

Tab.3 Numbers of Extracting Ground Points

场景	点个数	2种索引结构滤除地面点个数	
		八叉树	KD-OcTree
场景1	823 855	608 627	635 217
场景2	1 563 493	1 085 441	1 100 836
场景3	3 352 290	1 763 461	2 110 014

3个场景、两种索引结构滤除地面点的效果分别如图5~7所示。可以看出,KD-OcTree混



合索引滤除地面点的效果明显优于八叉树单一索引。当场景中包含低矮植被时,八叉树无法将低矮植被与地面点分离,会出现点云连成一片的现象,即图 5(a)、图 6(a)、图 7(a)中红色矩形框标示的部分;而 KD-OcTree 混合索引能够更加准确地滤除地面点,如图 5(b)、图 6(b)、图 7(b)所示。

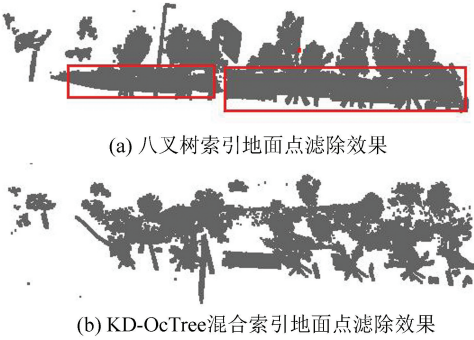


图 5 场景 1 地面点滤除效果图

Fig.5 Effect of Ground Filtering in Scene 1

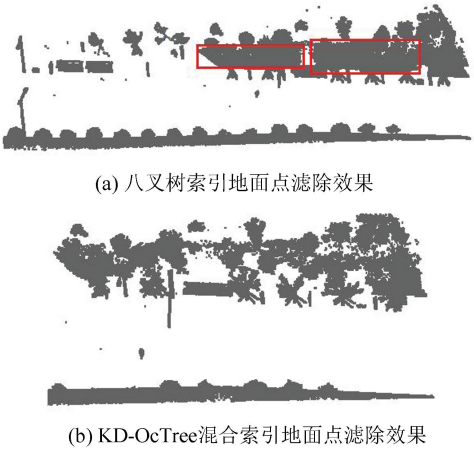


图 6 场景 2 地面点滤除效果图

Fig.6 Effect of Ground Filtering in Scene 2

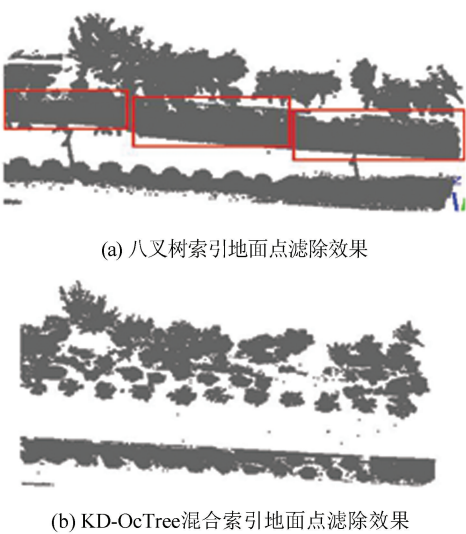


图 7 场景 3 地面点滤除效果图

Fig.7 Effect of Ground Filtering in Scene 3

3.5 阈值敏感度测试

使用 KD-OcTree 混合索引对海量点云进行组织与管理时,涉及到一个关键阈值,即全局 KD 树叶子节点的点数阈值。该阈值越大,全局 KD 树叶子节点包含的点数越多,全局 KD 树层数越小,而局部八叉树层数相对越大。本试验对点数阈值与建树时间的关系进行测试,结果如图 8 所示。从测试结果可以看出,随着点数阈值的增大,构建 KD-OcTree 混合索引所消耗的时间逐渐增加。当阈值增到最大值,即达到点个数甚至超过点个数时,建树的时间主要为局部八叉树所耗费的时间,逐渐趋于稳定。

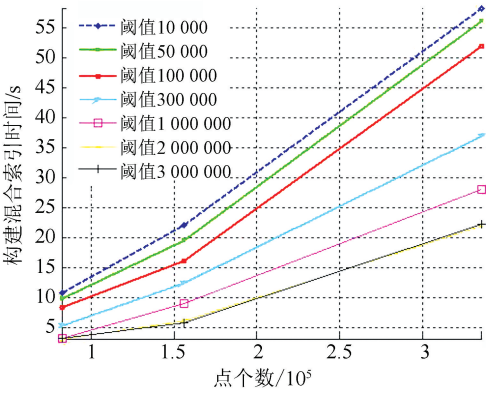


图 8 不同阈值情况下构建 KD-OcTree 的时间对比

Fig.8 Time Comparison of Building KD-OcTree with Different Thresholds

4 结 语

本文提出了一种对车载 LiDAR 点云进行快速、高效的组织与管理方法,即两级混合空间索引结构——KD-OcTree。首先,使用全局 KD 树对海量点云进行组织,保证索引结构的整体平衡,避免树结构出现倾斜、深度过大等问题;在此基础上,使用局部八叉树对海量点云进行管理,以块为单位对点云实现快速检索。使用 3 个场景数据测试 KD-OcTree 混合索引的效率,从 4 个方面与 KD 树和八叉树索引进行对比分析,验证了 KD-OcTree 混合索引不仅能够提高索引构造和邻域搜索的速度,而且能够改善三维目标感知的效果。但由于测试数据为真实场景点云,无法预知测试数据中确切的地面点数量,所以试验结果仅从目视滤除效果判断 KD-OcTree 混合索引对三维目标感知的影响。

下一步研究工作主要有:①选取尺寸更大、场景更复杂的数据测试 KD-OcTree 混合索引的效率;②使用机器学习方法自动设置全局 KD 树阈

值;③在考虑点云数据空间几何关系的基础上,考虑更多维信息,如曲率、回波强度等,构建多维空间索引结构;④研究大数据架构下并行索引构建技术等。

## 参 考 文 献

- [1] Samet H. The Quadtree and Related Hierarchical Data Structures [J]. *ACM Computing Surveys*, 1984, 16(2): 187-260
- [2] Meagher D. Geometric Modeling Using Octree Encoding[J]. *Computer Graphics and Image Processing*, 1982, 19(2): 129-147
- [3] Karlsson J S. HQT\*: A Scalable Distributed Data Structure for High-Performance Spatial Accesses [M]. Norwell: Kluwer Academic Publishers, 2000
- [4] Gong Jun, Xie Xiao. Three-Dimension Visualization Query Method Based on R-Tree[J]. *Geomatics and Information Science of Wuhan University*, 2011, 36(10): 1 140-1 143(龚俊,谢潇.基于R树索引的三维可视化查询方法[J].武汉大学学报·信息科学版,2011,36(10):1 140-1 143)
- [5] Bechmann N, Kriegel H P, Schneider R, et al. The R\*-Tree: An Efficient and Robust Access Method for Points and Rectangles[C]. The ACM SIGMOD International Conference on Management of Data, Atlantic City, New Jersey, USA, 1990
- [6] Zhu Q, Yao X, Huang D, et al. An Efficient Data Management Approach for Large Cybercity GIS[C]. Symposium on Geospatial Theory, Processing and Applications, Ottawa, Canada, 2002
- [7] Gong Jun, Zhu Qing, Zhang Yeting, et al. An Efficient 3D R-tree Extension Method Concerned with Levels of Details [J]. *Acta Geodaetica et Cartographica Sinica*, 2011, 40(2): 249-255(龚俊,朱庆,张叶廷,等.顾及多细节层次的三维R树索引扩展方法[J].测绘学报,2011,40(2):249-255)
- [8] Chen Chi, Wang Ke, Xu Wenxue. Real-Time Visualizing of Massive Vehicle-Borne Laser Scanning Point Clouds[J]. *Geomatics and Information Science of Wuhan University*, 2015, 40(9): 1 163-1 168(陈驰,王珂,徐文学.海量车载激光扫描点云数据的快速可视化方法[J].武汉大学学报·信息科学版,2015,40(9):1 163-1 168)
- [9] Guo Ming. Research on Huge Fine Spatial Data Management [D]. Wuhan: Wuhan University, 2011(郭明.海量精细空间数据管理技术研究[D].武汉:武汉大学,2011)
- [10] Xie Hong. Model Reconstruction Algorithms for Massive Point Cloud from Terrestrial Laser Scanner [D]. Wuhan: Wuhan University, 2013(谢洪.基于地面三维激光扫描技术海量定义模型重建关键算法研究[D].武汉:武汉大学,2013)
- [11] Wang Yanmin, Guo Ming. A Combined 2D and 3D Spatial Indexing of Very Large Point-Cloud Data Sets[J]. *Acta Geodaetica et Cartographica Sinica*, 2012, 41(4): 605-612(王晏民,郭明.大规模点云数据的二维与三维混合索引方法[J].测绘学报,2012,41(4):605-612)
- [12] Yang Jiansi, Huang Xianfeng. A Hybrid Spatial Index for Massive Point Cloud Data Management and Visualization [J]. *Transaction in GIS*, 2014, 18(S1): 97-108
- [13] Richter R, Behrens M, Döllner J. Object Class Segmentation of Massive 3D Point Clouds of Urban Areas Using Point Cloud Topology[J]. *International Journal of Remote Sensing*, 2013, 34(23): 8 408-8 424
- [14] Qi Xiaolong. Research on Mixed Indexing of Octree and R<sup>+</sup> Tree for Points Cloud [D]. Beijing: Beijing University of Civil Engineering and Architecture, 2013(齐晓隆.基于八叉树与R<sup>+</sup>树的点云混合索引研究[D].北京:北京建筑大学,2013)
- [15] Yang Jiansi. A Method of Combing the Model of the Global Quadtree Index with Local KD-Tree for Massive Airborne LiDAR Point Cloud Data Organization[J]. *Geomatics and Information Science of Wuhan University*, 2014, 39(8): 918-922(杨建思.一种四叉树与KD树结合的海量机载LiDAR数据组织管理方法[J].武汉大学学报·信息科学版,2014,39(8):918-922)
- [16] Li Minglei. Technology of Preprocessing on 3D Laser Scanned Point Clouds [D]. Zhengzhou: Information Engineering University, 2014(李明磊.地面激光扫描点云预处理技术研究[D].郑州:信息工程大学,2014)
- [17] Yu Yongtao. Research on Three-Dimensional Object Detection from Large-Scale Mobile Laser Scanning Point Clouds [D]. Xiamen: Xiamen University, 2015(于永涛.大场景车载激光点云三维目标检测算法研究[D].厦门:厦门大学,2015)

A New Method of Hybrid Index for Mobile LiDAR Point Cloud Data

ZHANG Rui<sup>1,2</sup> LI Guangyun<sup>1</sup> WANG Li<sup>1</sup> LI Minglei<sup>1</sup> ZHOU Yanglin<sup>1</sup>

1 School of Navigation and Aerospace Engineering, Information Engineering University, Zhengzhou 450001, China  
2 North China University of Water Resources and Electric Power, Zhengzhou 450045, China

**Abstract:** For mobile LiDAR point cloud data, a new hybrid index structure combining global KD-tree and local Octree is proposed to improve the efficiency of data organization and management, which is named as KD-OcTree index. Firstly, global KD-tree reconstructs the spatial neighborhood relations by defining the segmenting dimension and segmenting planes, for the purpose of ensuring the balance of the whole index. Then, local Octree is constructed in the leafs of KD-tree, which can avoid some shortcomings such as the unbalance of point cloud distribution, deeper Octree, large amount of non-point space, and so on. Lastly, we take three real scenes’ point clouds as test data to process. The experimental results and comparative analysis show that the KD-OcTree index can not only improve the speed of constructing index and neighborhood searching, but also improve the effect of data-processing and influence the reliability of classification.

**Key words:** mobile LiDAR; point clouds; global KD-tree; local Octree; hybrid spatial index

**First author:** ZHANG Rui, associate professor, specializes in management of massive point clouds and 3D targets recognition. E-mail: zhangrui@ncwu.edu.cn  
**Foundation support:** The National Natural Science Foundation of China, No. 41501491; the Key Science Research Program of Higher Education of Henan Province, No. 16A520062.